

UNIVERSITÀ DEGLI STUDI DI BERGAMO

FACOLTÀ DI INGEGNERIA

Corso di dottorato di ricerca in **Meccatronica e Tecnologie Innovative**

*XXIII* Ciclo



**DISPOSITIVI MECCATRONICI PER  
L'ABBATTIMENTO DELLE OSCILLAZIONI DI  
PRESSIONE CON PROPAGAZIONE  
MONODIMENSIONALE**

**Coordinatore:**

Chiar.mo Prof. Riccardo RIVA

**Tutor:**

Chiar.mo Prof. Paolo RIGHETTINI

**Tesi di Dottorato di**

Ing. Michele MATTANA

Matr. 1004491

Ing. Paolo MOLTRASIO

Matr. 1004490

Anno Accademico 2010/2011



*Ai miei genitori*

*A Simona*



## **RINGRAZIAMENTI:**

*I nostri più sentiti ringraziamenti vanno al Prof. Paolo Righettini per la sua dedizione e impegno profuso in questo lavoro, e per tutti gli insegnamenti ricevuti.*

*Un ringraziamento va sicuramente al direttore del Dottorato in Meccatronica e Tecnologie Innovative, Prof. Riccardo Riva, anima e memoria storica del Dipartimento di Progettazione e Tecnologie, per la sua disponibilità e per le sue pillole di saggezza.*

*Un ringraziamento sentito va al prof. Roberto Strada per il supporto fornitoci in tutta l'attività svolta.*

*Ringrazio i miei genitori e mia sorella per avermi aiutato lungo tutto il percorso di studi fino al raggiungimento di questo importante traguardo. Ringrazio inoltre Laura per essermi stata sempre accanto nelle situazioni di difficoltà e per darmi ogni giorno gli spunti necessari per raggiungere nuovi obiettivi.*

*Ringrazio i miei genitori per avermi trasmesso la curiosità per la conoscenza e per il sostegno su cui ho sempre potuto contare, e la mia famiglia allargata composta da sorelle e cognati per la disponibilità e pazienza che hanno sempre dimostrato. Un ringraziamento particolare a Simona, per aver cambiato il senso della mia vita.*



# INDICE

<b>INTRODUZIONE</b>	<b>25</b>
<b>1 La propagazione e il controllo delle onde di pressione</b>	<b>31</b>
1.1 Suono e rumore . . . . .	31
1.1.1 Elementi di psicoacustica . . . . .	36
1.2 Propagazione delle onde piane . . . . .	42
1.3 Il processo di abbattimento delle onde piane . . . . .	50
1.3.1 Meccanismo di cancellazione delle onde piane . . . . .	54
1.3.2 Meccanismo di assorbimento delle onde piane . . . . .	62
1.4 Soluzioni disponibili per l'abbattimento delle onde di pressione . . . . .	66
1.4.1 Metodi passivi . . . . .	67
1.4.2 Metodi attivi . . . . .	71
<b>2 Progettazione di un sistema ANC</b>	<b>77</b>
2.1 Progettazione di un sistema per la generazione di onde piane . . . . .	77
2.1.1 Modellazione numerica del condotto . . . . .	86
2.2 Sistema di controllo per l'abbattimento del rumore . . . . .	89
2.2.1 Schema di controllo . . . . .	91
2.2.2 Sensori e attuatori per il controllo . . . . .	95
2.2.3 Hardware di controllo . . . . .	98
2.3 Algoritmi di controllo . . . . .	101
2.3.1 Tecnica di controllo ARMS . . . . .	102
2.3.2 Tecnica di controllo AS . . . . .	107
2.3.3 Tecnica di controllo MFS . . . . .	111
2.3.4 Tecnica di controllo MFP . . . . .	113

<b>3</b>	<b>Analisi sperimentale degli algoritmi ANC</b>	<b>115</b>
3.1	Apparato sperimentale . . . . .	115
3.2	Sensori, attuatori e hardware per il controllo attivo . . . . .	119
3.2.1	Scelta di sensori e attuatori . . . . .	119
3.2.2	Hardware e software utilizzato . . . . .	120
3.2.3	Progetto di un amplificatore . . . . .	125
3.2.4	Caratterizzazione sperimentale degli attuatori . . . . .	132
3.3	Caratterizzazione sperimentale dell'impianto di test . . . . .	147
3.4	Risultati sperimentali . . . . .	156
3.4.1	Tecnica di controllo ARMS . . . . .	156
3.4.2	Tecnica di controllo AS . . . . .	168
3.4.3	Tecnica di controllo MFS . . . . .	176
3.4.4	Tecnica di controllo MFP . . . . .	182
3.5	Conclusioni . . . . .	189
<b>4</b>	<b>Sistemi per l'abbattimento con attuazione indiretta</b>	<b>193</b>
4.1	Abbattimento con attuazione indiretta . . . . .	193
4.2	Teoria e modellazione di un condotto . . . . .	196
4.2.1	Applicazione del modello ad un risonatore di Helmholtz . . . . .	204
4.2.2	Applicazione del modello ad un risonatore side-branch . . . . .	210
4.3	Sistemi indiretti per l'abbattimento . . . . .	212
4.3.1	Risonatore di Helmholtz . . . . .	212
4.3.2	Risonatore side-branch . . . . .	215
4.4	Sensibilità alla variazione dei parametri dei risonatori . . . . .	217
4.4.1	Analisi di sensibilità di un risonatore di Helmholtz . . . . .	218
4.4.2	Analisi di sensibilità di un risonatore side-branch . . . . .	228
4.4.3	Modellazione numerica dei risonatori . . . . .	232
<b>5</b>	<b>Impianto di test per i sistemi di abbattimento</b>	<b>239</b>
5.1	Introduzione al problema . . . . .	239
5.1.1	Campagna di misura su compressore industriale . . . . .	240
5.1.2	Definizione della catena di misura . . . . .	242
5.1.3	Definizione delle prove sperimentali . . . . .	246
5.1.4	Analisi dei risultati sperimentali . . . . .	251
5.2	Progettazione e caratterizzazione di un impianto di test . . . . .	259



5.2.1	Progettazione dell'impianto . . . . .	261
5.2.2	Caratterizzazione sperimentale dell'impianto . . . . .	268
<b>6</b>	<b>Analisi sperimentale dei sistemi ad attuazione indiretta</b>	<b>275</b>
6.1	Progettazione e realizzazione dei risonatori fissi . . . . .	275
6.1.1	Progettazione dei risonatori side-branch rettilinei . . . . .	277
6.1.2	Progettazione dei risonatori side-branch a spirale . . . . .	281
6.1.3	Progettazione dei risonatori di Helmholtz . . . . .	288
6.2	Risonatore di Helmholtz a geometria variabile . . . . .	298
6.2.1	Progettazione meccanica del dispositivo . . . . .	300
6.2.2	Scelta degli attuatori . . . . .	309
6.2.3	Progettazione del circuito elettrico . . . . .	311
6.3	Sistema di controllo . . . . .	314
6.3.1	Schema di controllo . . . . .	316
6.3.2	Algoritmo di controllo . . . . .	319
6.4	Risultati sperimentali . . . . .	326
	<b>CONCLUSIONI</b>	<b>331</b>
<b>A</b>	<b>Complementi analitici</b>	<b>335</b>
A.1	Punti di minimo di una funzione . . . . .	335
A.2	Analisi spettrale di un segnale . . . . .	337
A.3	Differenti espressioni per l'equazione delle onde . . . . .	339
A.4	Stima della funzione di trasferimento di un sistema . . . . .	340
<b>B</b>	<b>Sensori e apparecchiature</b>	<b>345</b>
B.1	Sistemi di acquisizione . . . . .	345
B.2	Amplificatori di segnale . . . . .	347
B.3	Microfoni . . . . .	350
B.4	Attuatori acustici . . . . .	350
B.5	Trasduttori di pressione . . . . .	351
<b>C</b>	<b>Script di MATLAB utilizzati per il sistema ANC</b>	<b>355</b>
C.1	Script MatLab per la mappatura dell'RMS della pressione . . . . .	355
C.2	Script MatLab per la mappatura dello spettro della pressione . . . . .	356

---

<b>D Codice di controllo per il sistema ANC</b>	<b>359</b>
D.1 Codice sorgente per l'algoritmo ARMS . . . . .	359
D.1.1 Struttura del codice . . . . .	359
D.1.2 Analisi dettagliata delle funzioni del codice . . . . .	386
D.2 Codice sorgente per l'algoritmo AS . . . . .	396
D.2.1 Struttura del codice . . . . .	396
D.2.2 Analisi dettagliata delle funzioni del codice . . . . .	410
D.3 Codice sorgente per l'algoritmo MFS . . . . .	412
D.3.1 Struttura del codice . . . . .	413
D.3.2 Analisi dettagliata delle funzioni del codice . . . . .	427
D.4 Codice sorgente per l'algoritmo MFP . . . . .	427
D.4.1 Struttura del codice . . . . .	427
D.4.2 Analisi dettagliata delle funzioni del codice . . . . .	443
<b>E Script di MATLAB utilizzati per i sistemi risonanti</b>	<b>447</b>
E.1 Analisi della misura su compressore industriale . . . . .	447
E.2 Script MatLab modello condotto e risonatore di Helmholtz . . . . .	464
E.3 Script MatLab modello condotto e risonatore side-branch . . . . .	471
<b>F Progetto del risonatore di Helmholtz</b>	<b>477</b>
F.1 Progettazione del risonatore di Helmholtz a geometria variabile . . . . .	477
<b>G Codice di controllo per risonatore a geometria variabile</b>	<b>481</b>
G.1 Codice sorgente per il risonatore di Helmholtz a geometria variabile . . . . .	481
G.1.1 Struttura del codice . . . . .	481
<b>BIBLIOGRAFIA</b>	<b>555</b>

# ELENCO DELLE FIGURE

1.1	<i>Onde di pressione prodotte da una superficie piana vibrante. . . . .</i>	32
1.2	<i>Andamento della pressione sonora rilevata in un punto vicino alla superficie vibrante. . . . .</i>	32
1.3	<i>Variazione della pressione atmosferica assoluta in relazione alle onde sonore. . .</i>	33
1.4	<i>Rumore generato da vibrazioni. . . . .</i>	36
1.5	<i>Rumore generato da effetti aerei. . . . .</i>	36
1.6	<i>Curve isofoniche. . . . .</i>	38
1.7	<i>Gamma di frequenze e livelli di pressione sonora dell'orecchio umano. . . . .</i>	39
1.8	<i>Generazione di onde piane in un condotto. . . . .</i>	51
1.9	<i>Andamento della pressione acustica e della velocità di particella dovuta alla presenza di una sorgente monopolo con generazione di onde piane. . . . .</i>	53
1.10	<i>Cancellazione di due onde piane in un condotto. . . . .</i>	55
1.11	<i>Processo di cancellazione perfetta dell'onda a valle della sorgente di controllo. . .</i>	56
1.12	<i>Andamento delle pressioni acustiche all'interno del condotto nel caso in cui la distanza delle sorgenti non rispetta la relazione 1.77. . . . .</i>	58
1.13	<i>Andamento delle pressioni acustiche all'interno del condotto nel caso in cui la distanza delle sorgenti rispetta la relazione 1.77. . . . .</i>	59
1.14	<i>Rappresentazione grafica della relazione 1.88: andamento del rapporto delle potenze in <math>x = 0</math> con e in assenza della sorgente di controllo. . . . .</i>	61
1.15	<i>Rappresentazione grafica della relazione 1.92: andamento della potenza acustica al variare del flusso di volume <math>q_c</math> introdotto dalla sorgente di controllo. . . . .</i>	63
1.16	<i>Processo di cancellazione dovuto al massimo assorbimento da parte della sorgente di controllo. . . . .</i>	65
1.17	<i>Risonatori di Helmholtz. . . . .</i>	68
1.18	<i>Risonatore di helmholtz semplice. . . . .</i>	68

1.19	<i>Risonatore di helmholtz semplice. Principio di funzionamento.</i>	68
1.20	<i>Accumulatore per regolarizzare il flusso.</i>	69
1.21	<i>Filtro acustico passa-basso.</i>	70
1.22	<i>Filtro acustico passa-alto.</i>	70
1.23	<i>Filtro "volume-choke-volume".</i>	71
1.24	<i>Cuffie con controllo attivo del rumore.</i>	72
1.25	<i>Sistema di controllo attivo del rumore all'interno dell'abitacolo di un automobile.</i>	73
1.26	<i>Performance di abbattimento del rumore del sistema di controllo attivo.</i>	73
2.1	<i>Andamento del rapporto <math>R/X</math> in funzione della distanza, per varie frequenze.</i>	81
2.2	<i>Schema del sistema fisico oggetto di studio.</i>	84
2.3	<i>Modi di vibrare dell'ambiente sonoro.</i>	86
2.4	<i>Definizione del modello dell'ambiente sonoro mediante metodo ad elementi finiti.</i>	88
2.5	<i>Analisi armonica dell'ambiente sonoro: 1° modo di vibrare.</i>	88
2.6	<i>Analisi armonica dell'ambiente sonoro: 2° modo di vibrare.</i>	88
2.7	<i>Analisi armonica dell'ambiente sonoro: 3° modo di vibrare.</i>	89
2.8	<i>Analisi armonica dell'ambiente sonoro: risposta in frequenza da 200 a 450 Hz.</i>	89
2.9	<i>Principio di sovrapposizione degli effetti.</i>	89
2.10	<i>Elementi di un sistema di controllo attivo del rumore.</i>	91
2.11	<i>Sistema di controllo attivo del rumore in feedforward.</i>	91
2.12	<i>Sistema di controllo attivo del rumore in feedback.</i>	92
2.13	<i>Limiti della risposta in frequenza di quattro microfoni di differenti taglie: 1/8, 1/4, 1/2 e 1".</i>	95
2.14	<i>Range dinamico di quattro microfoni di differenti taglie: 1/8, 1/4, 1/2 e 1".</i>	95
2.15	<i>Risposta in frequenza di un classico microfono per misurazioni acustiche (Fonte: Bruel &amp; Kjaer).</i>	96
2.16	<i>Mappatura analitica dell'RMS - 3D.</i>	104
2.17	<i>Mappatura analitica dell'RMS - isolinee.</i>	104
2.18	<i>Funzione <math>y = (x + 1)^2</math>.</i>	105
2.19	<i>Segnale complessivo acquisito dal sistema.</i>	107
2.20	<i>Segnali costituenti il segnale complessivo.</i>	108
2.21	<i>Spettro del segnale complessivo.</i>	109
2.22	<i>Mappatura analitica dell'ampiezza spettrale - 3D.</i>	110
2.23	<i>Mappatura analitica dell'ampiezza spettrale - isolinee.</i>	110

3.1	<i>Controllo attivo del rumore: schema del sistema ANC.</i>	116
3.2	<i>Controllo attivo del rumore: modello 3D del condotto.</i>	116
3.3	<i>Controllo attivo del rumore: sezione del condotto.</i>	117
3.4	<i>Controllo attivo del rumore: modello 3D del supporto per le casse acustiche preposte alla generazione di rumore.</i>	117
3.5	<i>Controllo attivo del rumore: modello 3D dell'apparato di test.</i>	118
3.6	<i>Controllo attivo del rumore: apparato sperimentale di test.</i>	118
3.7	<i>Controllo attivo del rumore: supporto per la cassa di generazione del rumore.</i>	118
3.8	<i>Controllo attivo del rumore: vista posteriore del sistema reale.</i>	118
3.9	<i>Interfaccia utente, GUI - host.</i>	122
3.10	<i>Interfaccia utente, schema a blocchi - host.</i>	123
3.11	<i>Controllore NI PXI.</i>	123
3.12	<i>Schema a blocchi del sistema ANC - target.</i>	124
3.13	<i>Schema elettrico di un amplificatore non invertente.</i>	127
3.14	<i>Circuito equivalente di un amplificatore non invertente.</i>	128
3.15	<i>Simulazione dell'amplificatore mediante NI Circuit Design 10.</i>	129
3.16	<i>Schema elettrico dell'amplificatore audio.</i>	130
3.17	<i>Funzione di trasferimento teorica e sperimentale tra ingresso e uscita dell'amplificatore.</i>	131
3.18	<i>Prototipo dell'amplificatore audio.</i>	132
3.19	<i>Schema della catena di misura per la caratterizzazione delle casse acustiche.</i>	133
3.20	<i>Funzione di trasferimento della cassa di controllo eseguita con prove a frequenza costante.</i>	139
3.21	<i>Spettro del segnale in ingresso, costituito da uno sweep in frequenza.</i>	140
3.22	<i>Funzione di trasferimento dell'altoparlante di controllo, ottenuta mediante prova con sweep in frequenza.</i>	140
3.23	<i>Confronto tra i risultati ottenuti dalle due metodologie di misura.</i>	142
3.24	<i>Funzione di trasferimento dell'altoparlante di controllo a <math>f = 300</math> Hz in funzione della distanza del microfono.</i>	143
3.25	<i>Funzione di trasferimento dell'altoparlante di controllo ottenuta con sweep in frequenza e correzione di fase.</i>	143
3.26	<i>Funzione di trasferimento dell'altoparlante di controllo ottenuta con l'inserimento dell'amplificatore.</i>	144

3.27	<i>Funzione di trasferimento dell'altoparlante di controllo ottenuta con l'inserimento dell'amplificatore e la correzione di fase. . . . .</i>	144
3.28	<i>Funzione di trasferimento dell'altoparlante per la generazione del rumore ottenuta con l'inserimento dell'amplificatore. . . . .</i>	145
3.29	<i>Funzione di trasferimento dell'altoparlante per la generazione del rumore ottenuta con l'inserimento dell'amplificatore e la correzione di fase. . . . .</i>	145
3.30	<i>Spettro del segnale in ingresso alla cassa acustica, rilevato a valle dell'amplificatore. . . . .</i>	146
3.31	<i>Confronto tra le funzioni di trasferimento della cassa di controllo, rilevate a monte e a valle dell'amplificatore. . . . .</i>	146
3.32	<i>Confronto tra le funzioni di trasferimento della cassa di per la generazione del rumore, rilevate a monte e a valle dell'amplificatore. . . . .</i>	146
3.33	<i>Schema della catena di misura della funzione di trasferimento tra la cassa acustica di rumore ed il microfono di misura del sistema di controllo sistema. .</i>	147
3.34	<i>Schema della catena di misura della funzione di trasferimento tra la cassa acustica di controllo ed il microfono di misura del sistema di controllo sistema.</i>	147
3.35	<i>Funzione di trasferimento tra la cassa preposta alla generazione del rumore ed il microfono posto all'interno del condotto con distanza della cassa acustica pari a 10 mm. . . . .</i>	148
3.36	<i>Funzione di trasferimento tra la cassa preposta alla generazione del rumore ed il microfono posto all'interno del condotto con distanza della cassa acustica pari a 100 mm. . . . .</i>	149
3.37	<i>Funzione di trasferimento tra la cassa acustica di controllo ed il microfono posto all'interno del condotto con distanza della cassa acustica di rumore pari a 15mm.</i>	150
3.38	<i>Funzione di trasferimento tra la cassa acustica di controllo ed il microfono posto all'interno del condotto aperto ad entrambe le estremità. . . . .</i>	150
3.39	<i>Andamento delle frequenze proprie dell'ambiente sonoro in configurazione di condotto chiuso ad una estremità. . . . .</i>	151
3.40	<i>Andamento delle frequenze proprie dell'ambiente sonoro in configurazione di condotto aperto. . . . .</i>	151
3.41	<i>Andamento delle frequenze proprie dell'ambiente sonoro in configurazione di condotto aperto. Confronto fra i risultati ottenuti. . . . .</i>	152
3.42	<i>Andamento delle frequenze proprie dell'ambiente sonoro in configurazione di condotto aperto. Sintesi dei risultati ottenuti . . . . .</i>	152

3.43	<i>Mappatura sperimentale dell'RMS - 3D.</i>	154
3.44	<i>Mappatura sperimentale dell'RMS - isolinee.</i>	154
3.45	<i>Confronto tra la mappatura analitica e sperimentale dell'andamento dell'RMS.</i>	154
3.46	<i>Mappatura sperimentale dell'ampiezza spettrale - 3D.</i>	155
3.47	<i>Mappatura dell'ampiezza spettrale - isolinee.</i>	155
3.48	<i>Confronto tra la mappatura analitica e sperimentale dell'andamento dell'ampiezza spettrale della pressione sonora.</i>	156
3.49	<i>Schema a blocchi dell'algoritmo di ricerca del minimo mediante la valutazione del valore quadratico medio della pressione sonora.</i>	157
3.50	<i>Soglia di rumore impostata nell'algoritmo di controllo attivo del rumore.</i>	162
3.51	<i>Andamento del valore quadratico medio <math>RMS_1</math>.</i>	164
3.52	<i>Andamento del gradiente <math>\nabla RMS_A</math> dell'ampiezza del segnale di controllo.</i>	164
3.53	<i>Andamento dell'ampiezza <math>A_{cont}</math> del segnale di controllo.</i>	164
3.54	<i>Andamento del gradiente <math>\nabla RMS_\phi</math> della fase del segnale di controllo.</i>	165
3.55	<i>Andamento della fase <math>\phi_{cont}</math> del segnale di controllo.</i>	165
3.56	<i>Andamento dei valori quadratici medi <math>RMS_1</math>, <math>RMS_2</math> e <math>RMS_3</math>.</i>	165
3.57	<i>Andamento del guadagno <math>G_2^k</math>.</i>	166
3.58	<i>Andamento del numero di iterazioni con l'algoritmo basato sul RMS.</i>	167
3.59	<i>Schema a blocchi dell'algoritmo di ricerca del minimo mediante l'analisi spettrale su una frequenza di interesse.</i>	169
3.60	<i>Andamento dell'ampiezza del picco spettrale con algoritmo basato sull'analisi spettrale.</i>	171
3.61	<i>Andamento del gradiente <math>\nabla Pf_A</math> dell'ampiezza del segnale di controllo con algoritmo basato sull'analisi spettrale.</i>	172
3.62	<i>Andamento dell'ampiezza <math>A_{cont}</math> del segnale di controllo con algoritmo basato sull'analisi spettrale.</i>	172
3.63	<i>Andamento del gradiente <math>\nabla Pf_\phi</math> della fase del segnale di controllo con algoritmo basato sull'analisi spettrale.</i>	173
3.64	<i>Andamento della fase <math>\phi_{cont}</math> del segnale di controllo con algoritmo basato sull'analisi spettrale.</i>	173
3.65	<i>Andamento di <math>Pf_1</math>, <math>Pf_2</math> e <math>Pf_3</math> con algoritmo basato sull'analisi spettrale.</i>	173
3.66	<i>Andamento del guadagno <math>G_2^k</math> con algoritmo basato sull'analisi spettrale.</i>	174
3.67	<i>Andamento del numero di iterazioni con algoritmo basato sull'analisi spettrale.</i>	174

3.68	<i>Schema a blocchi dell'algoritmo di ricerca del minimo nel caso di rumore con più armoniche. . . . .</i>	176
3.69	<i>Andamento delle ampiezze dei picchi spettrali con algoritmo ricorsivo basato sull'analisi spettrale. . . . .</i>	178
3.70	<i>Andamento del gradiente <math>\nabla Pf_A</math> delle ampiezze del segnale di controllo con algoritmo ricorsivo basato sull'analisi spettrale. . . . .</i>	179
3.71	<i>Andamento delle ampiezze <math>A_{cont}</math> del segnale di controllo con algoritmo ricorsivo basato sull'analisi spettrale. . . . .</i>	179
3.72	<i>Andamento del gradiente <math>\nabla Pf_\phi</math> delle fasi del segnale di controllo con algoritmo ricorsivo basato sull'analisi spettrale. . . . .</i>	179
3.73	<i>Andamento delle fasi <math>\phi_{cont}</math> del segnale di controllo con algoritmo ricorsivo basato sull'analisi spettrale. . . . .</i>	179
3.74	<i>Andamento delle ampiezze dei picchi spettrali <math>Pf_1</math>, <math>Pf_2</math> e <math>Pf_3</math> con algoritmo ricorsivo basato sull'analisi spettrale. . . . .</i>	180
3.75	<i>Andamento dei guadagni <math>G_2^k</math> con algoritmo ricorsivo basato sull'analisi spettrale.</i>	180
3.76	<i>Schema a blocchi dell'algoritmo di ricerca del minimo basato sull'analisi spettrale in parallelo. . . . .</i>	182
3.77	<i>Andamento delle ampiezze dei picchi spettrali con algoritmo simultaneo basato sull'analisi spettrale. . . . .</i>	185
3.78	<i>Andamento del gradiente <math>\nabla Pf_A</math> delle ampiezze del segnale di controllo con algoritmo simultaneo basato sull'analisi spettrale. . . . .</i>	186
3.79	<i>Andamento delle ampiezze <math>A_{cont}</math> del segnale di controllo con algoritmo simultaneo basato sull'analisi spettrale. . . . .</i>	186
3.80	<i>Andamento del gradiente <math>\nabla Pf_\phi</math> delle fasi del segnale di controllo con algoritmo simultaneo basato sull'analisi spettrale. . . . .</i>	186
3.81	<i>Andamento delle fasi <math>\phi_{cont}</math> del segnale di controllo con algoritmo simultaneo basato sull'analisi spettrale. . . . .</i>	186
3.82	<i>Andamento delle ampiezze dei picchi spettrali <math>Pf_1</math>, <math>Pf_2</math> e <math>Pf_3</math> con algoritmo simultaneo basato sull'analisi spettrale. . . . .</i>	187
3.83	<i>Andamento dei guadagni <math>G_2^k</math> con algoritmo simultaneo basato sull'analisi spettrale.</i>	188
3.84	<i>Riepilogo risultati con algoritmo basato sulla minimizzazione del picco spettrale con minimizzazione parallela. . . . .</i>	188
4.1	<i>Esempio di applicazione di un TMD in una struttura civile. . . . .</i>	194
4.2	<i>Modello di un assorbitore dinamico. . . . .</i>	194



4.3	<i>Modello del condotto con elemento infinitesimo di gas.</i>	197
4.4	<i>Forze agenti sull'elemento infinitesimo di gas.</i>	199
4.5	<i>Condotto elementare per la determinazione delle costanti <math>A_1</math> e <math>B_1</math>.</i>	202
4.6	<i>Modello condotto con risonatore di Helmholtz.</i>	204
4.7	<i>Modello condotto con impedenza equivalente.</i>	205
4.8	<i>Modello condotto con impedenza equivalente, con suddivisione fittizia in <math>\hat{x}_1</math> e <math>\hat{x}_2</math>.</i>	208
4.9	<i>Modello del condotto principale in assenza di dispositivi risonanti.</i>	209
4.10	<i>Modello condotto con risonatore side-branch.</i>	211
4.11	<i>Modello risonatore di Helmholtz.</i>	213
4.12	<i>Modello risonatore side-branch.</i>	217
4.13	<i>Risonatore di Helmholtz cilindrico.</i>	218
4.14	<i>Helmholtz: <math>L_R</math> in funzione di <math>f_R</math> e <math>L_C</math>.</i>	219
4.15	<i>Helmholtz: <math>f_R</math> in funzione di <math>L_C</math>.</i>	220
4.16	<i>Helmholtz: <math>f_R</math> in funzione di <math>L_R</math>.</i>	220
4.17	<i>Helmholtz: <math>f_R</math> in funzione di <math>D_C</math>.</i>	220
4.18	<i>Helmholtz: <math>f_R</math> in funzione di <math>D_R</math>.</i>	220
4.19	<i>Risonatore di Helmholtz cilindrico a geometria variabile.</i>	222
4.20	<i>Andamento pressione dinamica in assenza di dispositivi risonanti.</i>	224
4.21	<i>Andamento pressione dinamica in presenza del risonatore di Helmholtz con frequenza <math>f_R = 20</math> Hz.</i>	224
4.22	<i>Andamento portata dinamica in assenza di dispositivi risonanti.</i>	225
4.23	<i>Andamento portata dinamica in presenza del risonatore di Helmholtz con fre- quenza <math>f_R = 20</math> Hz.</i>	225
4.24	<i>Andamento pressione dinamica con risonatore di Helmholtz con diametro del collo ridotto.</i>	226
4.25	<i>Andamento portata dinamica con risonatore di Helmholtz con diametro del collo ridotto.</i>	226
4.26	<i>Andamento pressione dinamica con risonatore di Helmholtz tarato a 100 Hz.</i>	227
4.27	<i>Andamento portata dinamica con risonatore di Helmholtz tarato a 100 Hz.</i>	227
4.28	<i>Andamento del rapporto delle pressioni con e senza Helmholtz in <math>\bar{x}_2 = 5</math> m.</i>	228
4.29	<i>Risonatore side-branch cilindrico.</i>	228
4.30	<i>Side-branch: <math>f_R</math> in funzione di <math>L_R</math>.</i>	229
4.31	<i>Andamento pressione dinamica con risonatore side-branch con frequenza <math>f_R =</math> 20 Hz.</i>	231

4.32	<i>Andamento portata dinamica con risonatore side-branch con frequenza <math>f_R = 20</math> Hz.</i>	231
4.33	<i>Andamento del rapporto delle pressioni con e senza side-branch in <math>\bar{x}_2 = 6.9</math> m.</i>	231
4.34	<i>Modello numerico del sistema di analisi del risonatore di Helmholtz.</i>	233
4.35	<i>Rapporto delle pressioni con e senza risonatore di Helmholtz nel punto <math>P_1</math>: confronto fra i risultati del modello analitico e del modello numerico.</i>	234
4.36	<i>Modello numerico del sistema di analisi del risonatore di Helmholtz.</i>	235
4.37	<i>Rapporto delle pressioni con e senza risonatore side-branch nel punto <math>P_1</math>: confronto fra i risultati del modello analitico e del modello numerico.</i>	236
4.38	<i>Confronto tra il comportamento dinamico del risonatore di Helmholtz e il risonatore side-branch. Risultati dell'analisi numerica</i>	236
5.1	<i>Compressore industriale TEMPO 3.</i>	241
5.2	<i>Strumentazione utilizzata per la campagna di misura su compressore industriale. Da sinistra a destra: NI PXI, Bruel &amp; Kjaer NEXUS 2690, Bruel &amp; Kjaer PULSE 3560B, Alimentatore per encoder, PC per registrazione dati.</i>	245
5.3	<i>Schema dei punti di misura su compressore industriale TEMPO 3.</i>	246
5.4	<i>Posizione dei punti di misura su compressore industriale: mandata 1° stadio.</i>	247
5.5	<i>Posizione dei punti di misura su compressore industriale: aspirazione 2° stadio.</i>	248
5.6	<i>Posizione dei punti di misura su compressore industriale: mandata 2° stadio.</i>	248
5.7	<i>Posizione dei punti di misura su compressore industriale: aspirazione 3° stadio.</i>	248
5.8	<i>Posizione dei punti di misura su compressore industriale: mandata 3° stadio.</i>	248
5.9	<i>Campagna di misura su compressore industriale: andamento temporale delle pressioni nel punto P13.</i>	251
5.10	<i>Campagna di misura su compressore industriale: andamento della pressione sull'angolo giro (punto P13).</i>	251
5.11	<i>Campagna di misura su compressore industriale: analisi spettrale su un'acquisizione nel punto P13.</i>	253
5.12	<i>Campagna di misura su compressore industriale: analisi spettrale e rappresentazione delle frequenze in multipli della frequenza di rotazione dell'albero del compressore.</i>	254
5.13	<i>Campagna di misura sperimentale su compressore industriale: spettrogramma del punto P13.</i>	255

5.14	<i>Campagna di misura sperimentale su compressore industriale: spettrogramma del punto P13 per tutti i regimi di rotazione. In rosso vengono indicate le frequenze di pulsazione teoriche. . . . .</i>	255
5.15	<i>Campagna di misura sperimentale su compressore industriale: velocità istantanea di rotazione dell'albero del compressore. . . . .</i>	256
5.16	<i>Campagna di misura sperimentale su compressore industriale: spettro delle pulsazioni di pressione nel punto P12 per l'acquisizione effettuata nel gruppo G2. . . . .</i>	257
5.17	<i>Campagna di misura sperimentale su compressore industriale: spettro delle pulsazioni di pressione nel punto P13 per l'acquisizione effettuata nel gruppo G2. . . . .</i>	257
5.18	<i>Campagna di misura sperimentale su compressore industriale: spettro delle pulsazioni di pressione nel punto P12 per l'acquisizione effettuata nel gruppo G4. . . . .</i>	257
5.19	<i>Campagna di misura sperimentale su compressore industriale: spettro delle pulsazioni di pressione nel punto P13 per l'acquisizione effettuata nel gruppo G4. . . . .</i>	257
5.20	<i>Campagna di misura sperimentale su compressore industriale: rapporto tra i picchi spettrali dei segnali rilevati nei punti P12 e P13, per i vari gruppi di acquisizione. . . . .</i>	258
5.21	<i>Schema costruttivo del dimostratore: 1. Compressore alternativo monostadio; 2. Condotto di mandata; 3. Telaio di sostegno del condotto; 4. Sensori; 5. Sistema di attenuazione; 6. Tavolo di sostegno del sistema di attenuazione; 7. Serbatoio; 8. Telaio di sostegno del serbatoio; 9. Valvola di regolazione della pressione; 10. Controllore. . . . .</i>	261
5.22	<i>Schema costruttivo del serbatoio. . . . .</i>	264
5.23	<i>Compressore alternativo monostadio utilizzato nell'impianto di test. . . . .</i>	267
5.24	<i>Condotto di mandata dell'impianto di test. . . . .</i>	267
5.25	<i>Punti di misura sull'impianto di test. . . . .</i>	269
5.26	<i>Campagna di misura su impianto di test: andamento temporale delle pressioni nel punto PT1. . . . .</i>	270
5.27	<i>Campagna di misura su impianto di test: andamento temporale delle pressioni nel punto PT5. . . . .</i>	270

5.28	<i>Campagna di misura su impianto di test: analisi spettrale su un'acquisizione nel punto PT1.</i>	271
5.29	<i>Campagna di misura su impianto di test: analisi spettrale su un'acquisizione nel punto PT5.</i>	271
5.30	<i>Campagna di misura su impianto di test: analisi spettrale e rappresentazione delle frequenze in multipli della frequenza di rotazione dell'albero del compressore (punto PT1).</i>	272
5.31	<i>Campagna di misura su impianto di test: analisi spettrale e rappresentazione delle frequenze in multipli della frequenza di rotazione dell'albero del compressore (punto PT5).</i>	272
5.32	<i>Campagna di misura sperimentale su impianto di test: spettrogramma del punto PT1.</i>	272
5.33	<i>Campagna di misura sperimentale su impianto di test: spettrogramma del punto PT5.</i>	272
5.34	<i>Campagna di misura sperimentale su impianto di test: velocità istantanea di rotazione dell'albero del compressore.</i>	273
6.1	<i>Risonatore side-branch rettilineo <math>R_{26,0}^{SBR}</math>: schema dell'impianto.</i>	278
6.2	<i>Andamento spettrale oscillazioni di pressione: SS-1550-PT1.</i>	278
6.3	<i>Andamento spettrale oscillazioni di pressione: SS-1550-PT5.</i>	278
6.4	<i>Andamento spettrale oscillazioni di pressione: <math>R_{26,0}^{SBR}</math>-PT1-1550-PT1.</i>	279
6.5	<i>Andamento spettrale oscillazioni di pressione: <math>R_{26,0}^{SBR}</math>-PT1-1550-PT5.</i>	279
6.6	<i>Andamento spettrale oscillazioni di pressione: <math>R_{26,0}^{SBR}</math>-PT2-1550-PT1.</i>	280
6.7	<i>Andamento spettrale oscillazioni di pressione: <math>R_{26,0}^{SBR}</math>-PT2-1550-PT5.</i>	280
6.8	<i>Andamento spettrale oscillazioni di pressione: <math>R_{26,0}^{SBR}</math>-PT3-1550-PT1.</i>	280
6.9	<i>Andamento spettrale oscillazioni di pressione: <math>R_{26,0}^{SBR}</math>-PT3-1550-PT5.</i>	280
6.10	<i>Andamento spettrale oscillazioni di pressione: <math>R_{26,0}^{SBR}</math>-PT4-1550-PT1.</i>	280
6.11	<i>Andamento spettrale oscillazioni di pressione: <math>R_{26,0}^{SBR}</math>-PT4-1550-PT5.</i>	280
6.12	<i>Rapporto tra gli spettri dei segnali di pressione in corrispondenza di PT1, con e senza risonatore side-branch <math>R_{26,0}^{SBR}</math>.</i>	281
6.13	<i>Rapporto tra gli spettri dei segnali di pressione in corrispondenza di PT5, con e senza risonatore side-branch <math>R_{26,0}^{SBR}</math>.</i>	281
6.14	<i>Risonatore side-branch a spirale <math>R_{10,3}^{SBS}</math>.</i>	283
6.15	<i>Andamento spettrale oscillazioni di pressione: SS-1500-PT1.</i>	284
6.16	<i>Andamento spettrale oscillazioni di pressione: SS-1500-PT10.</i>	284

6.17	<i>Andamento spettrale oscillazioni di pressione: <math>R_{24,7}^{SBS}</math>-PT3-1500-PT1.</i>	284
6.18	<i>Andamento spettrale oscillazioni di pressione: <math>R_{24,7}^{SBS}</math>-PT3-1500-PT10.</i>	284
6.19	<i>Sovrapposizione spettri oscillazioni di pressione: SS-PT1.</i>	286
6.20	<i>Sovrapposizione spettri oscillazioni di pressione: SS-PT10.</i>	286
6.21	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{24,7}^{SBS}</math>-PT3-PT1.</i>	286
6.22	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{24,7}^{SBS}</math>-PT3-PT10.</i>	286
6.23	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{14,6}^{SBS}</math>-PT3-PT10.</i>	287
6.24	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{10,3}^{SBS}</math>-PT3-PT10.</i>	287
6.25	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{8,0}^{SBS}</math>-PT3-PT10.</i>	287
6.26	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{6,5}^{SBS}</math>-PT3-PT10.</i>	287
6.27	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{5,5}^{SBS}</math>-PT3-PT10.</i>	287
6.28	<i>Sezione del risonatore di Helmholtz cilindrico a geometria fissa.</i>	288
6.29	<i>Risonatore di Helmholtz cilindrico a geometria fissa: grandezze caratteristiche.</i>	289
6.30	<i>Risonatore di Helmholtz <math>R_{36,3}^H</math>: schema dell'impianto.</i>	290
6.31	<i>Sovrapposizione spettri oscillazioni di pressione: SS-PT4.</i>	291
6.32	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{36,3}^H</math>-PT3-PT4.</i>	292
6.33	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{25,7}^H</math>-PT3-PT4.</i>	292
6.34	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{20,2}^H</math>-PT3-PT4.</i>	292
6.35	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{14,3}^H</math>-PT3-PT4.</i>	292
6.36	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{36,3}^H</math>-PT3-PT4.</i>	293
6.37	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{36,3-2}^H</math>-PT3-PT4.</i>	293
6.38	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{26,8}^H</math>-PT3-PT4.</i>	294
6.39	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{18,8}^H</math>-PT3-PT4.</i>	294
6.40	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{12,4}^H</math>-PT3-PT4.</i>	294
6.41	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{11,0}^H</math>-PT3-PT4.</i>	294
6.42	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{24,6}^H</math>-PT3-PT4.</i>	296
6.43	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{15,5}^H</math>-PT3-PT4.</i>	296
6.44	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{12,3}^H</math>-PT3-PT4.</i>	296
6.45	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{11,0}^H</math>-PT3-PT4.</i>	296
6.46	<i>Dimensionamento risonatore Helmholtz a geometria variabile.</i>	299
6.47	<i>Schema del risonatore di Helmholtz a geometria variabile collegato al condotto principale.</i>	300
6.48	<i>Cilindro pneumatico Festo DNG-50-1000-PPV: vista in sezione.</i>	301
6.49	<i>Foto del risonatore di Helmholtz a geometria variabile realizzato.</i>	301

6.50	<i>Schema trasmissione per risonatore di Helmholtz a geometria variabile.</i>	302
6.51	<i>Assieme complessivo del risonatore di Helmholtz a geometria variabile.</i>	305
6.52	<i>Assieme complessivo del risonatore di Helmholtz a geometria variabile: vista di alcuni particolari della trasmissione.</i>	306
6.53	<i>Schema trasmissione risonatore di Helmholtz a geometria variabile: trasmissione delle forze.</i>	306
6.54	<i>Analisi FEM resistenza meccanica piastra supporto vite a ricircolo: applicazione delle condizioni al contorno e dei carichi.</i>	308
6.55	<i>Analisi FEM resistenza meccanica piastra supporto vite a ricircolo: mesh analisi FEM.</i>	308
6.56	<i>Analisi FEM resistenza meccanica piastra supporto vite a ricircolo: sforzi di von Mises e deformazioni.</i>	308
6.57	<i>Motore brushless Parker S.B.C. SMBA-82-56-03-8-14-S-2D-64-400.</i>	310
6.58	<i>Azionamento elettrico Parker S.B.C. SPD5N.</i>	310
6.59	<i>Trasmissione del moto dal motore alla vite a ricircolo.</i>	311
6.60	<i>Schema elettrico comando risonatore attivo: collegamento all'azionamento.</i>	312
6.61	<i>Schema elettrico comando risonatore attivo: collegamento al controllore.</i>	313
6.62	<i>Sistema di abbattimento indiretto delle oscillazioni di pressione.</i>	314
6.63	<i>Schema concettuale del sistema di controllo per l'abbattimento delle oscillazioni di pressione con il risonatore di Helmholtz a geometria variabile.</i>	316
6.64	<i>Schema di controllo del risonatore a geometria variabile.</i>	317
6.65	<i>Schema a blocchi del sistema di controllo per l'abbattimento delle oscillazioni di pressione mediante il risonatore di Helmholtz a geometria variabile.</i>	318
6.66	<i>Modello semplificato del risonatore di Helmholtz a geometria variabile con trasmissione.</i>	320
6.67	<i>Schema di controllo con regolatore PID.</i>	321
6.68	<i>Luogo delle radici del sistema controllato con regolatore PID al variare del guadagno proporzionale <math>K_P</math>.</i>	322
6.69	<i>Pannello utente per il controllo del sistema di controllo del risonatore di Helmholtz a geometria variabile.</i>	324
6.70	<i>Caratterizzazione sperimentale del risonatore di Helmholtz a geometria variabile.</i>	325
6.71	<i>Caratterizzazione sperimentale del risonatore di Helmholtz a geometria variabile, con correzione.</i>	325
6.72	<i>Sovrapposizione spettri oscillazioni di pressione: SS-PT4.</i>	327

6.73	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{\text{VAR}}^{\text{H}}</math>-PT3-PT4.</i>	327
6.74	<i>Sovrapposizione spettri oscillazioni di pressione: SS-PT5.</i>	327
6.75	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{\text{VAR}}^{\text{H}}</math>-PT3-PT5.</i>	327
6.76	<i>Sovrapposizione spettri oscillazioni di pressione: SS-PT10.</i>	327
6.77	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{\text{VAR}}^{\text{H}}</math>-PT3-PT10.</i>	327
6.78	<i>Sovrapposizione spettri oscillazioni di pressione: SS-PT4.</i>	328
6.79	<i>Sovrapposizione spettri oscillazioni di pressione: <math>R_{\text{VAR}}^{\text{H}}</math>-PT3-PT1.</i>	328
6.80	<i>Grafico riepilogativo abbattimento risonatore di Helmholtz a geometria variabile nel punto di misura PT4.</i>	328
6.81	<i>Grafico riepilogativo abbattimento risonatore di Helmholtz a geometria variabile nel punto di misura PT10.</i>	328
A.1	<i>Schema del sistema di cui si vuole eseguire la caratterizzazione in frequenza.</i>	341
A.2	<i>Schema del sistema di cui si vuole eseguire la caratterizzazione in frequenza, affetto dalla presenza di rumore.</i>	342
B.1	<i>Configurazione interna dell'amplificatore NEXUS.</i>	347
B.2	<i>Rumore in uscita in funzione del guadagno scelto dall'utente.</i>	349
B.3	<i>Curva risposta in frequenza microfono 4944.</i>	350
F.1	<i>Assieme della trasmissione per la movimentazione dello stantuffo del risonatore di Helmholtz a geometria variabile.</i>	478
F.2	<i>Elementi principali della trasmissione per la movimentazione dello stantuffo del risonatore di Helmholtz a geometria variabile.</i>	479
F.3	<i>Particolari della trasmissione per la movimentazione dello stantuffo del risonatore di Helmholtz a geometria variabile.</i>	480





# INTRODUZIONE

La necessità di migliorare la qualità di un prodotto è cresciuta in parallelo all'evoluzione delle tecnologie per l'attenuazione del rumore e, più in generale, delle oscillazioni di pressione emesse da un prodotto o presenti in un particolare impianto. L'esigenza di soddisfare tali requisiti, imposti talvolta dalla legislazione vigente, porta ad un adeguamento e ad un miglioramento in termini di qualità, affidabilità, quindi vita utile dei prodotti presenti sul mercato. Il riconoscimento normativo dei problemi legati all'inquinamento acustico e le conseguenti sanzioni previste per le infrazioni a norme accettate a livello mondiale, hanno dato impeto ad un intero campo di attività commerciali e di ricerca il cui obiettivo è quello di ricercare nuove tecniche e tecnologie per il controllo, e quindi per l'attenuazione del rumore. L'obiettivo del presente lavoro di tesi è quello di studiare e sviluppare, a partire da un'analisi approfondita degli aspetti legati ai fenomeni acustici, due apparati sperimentali per il test di sistemi e algoritmi per il controllo attivo e semi-attivo del rumore. Da questa esigenza sono stati sviluppati dei metodi per il controllo attivo del rumore nei quali, attraverso l'utilizzo di attuatori acustici, si controlla il rumore emesso da una sorgente creando così una "zona di silenzio". Ove l'utilizzo dei sistemi attivi con attuatori acustici risulta impraticabile, occorre utilizzare dispositivi passivi o semi-attivi. Per definire correttamente gli impianti per lo studio ed il test dei dispositivi e degli algoritmi, occorre modellare correttamente l'ambiente sonoro e le sue caratteristiche; a tal fine nella parte iniziale della tesi vengono analizzati in dettaglio i meccanismi che regolano la propagazione e l'abbattimento delle oscillazioni di pressione. La teoria viene affrontata in modo particolare considerando i sistemi in cui la propagazione delle onde è monodimensionale: tale scelta è dettata principalmente dall'elevato numero di casi industriali in cui la propagazione avviene lungo una direzione preferenziale e, conseguentemente, dall'effettiva applicabilità delle tecnologie che verranno sviluppate nella tesi. In particolare nel primo apparato verranno utiliz-

zati i classici attuatori acustici, mentre nel secondo apparato ci si prefigge lo scopo di studiare, progettare e testare dispositivi passivi di tipo risonante. Il principio di abbattimento realizzato nei due apparati è sostanzialmente identico e basato sul principio di sovrapposizione degli effetti, il quale afferma che la somma di due onde di pressione con medesima ampiezza e con fasi opposte è pari a zero: le due onde in questione saranno l'onda che si intende abbattere e l'onda inserita nel sistema dal dispositivo di attenuazione progettato. I due apparati sviluppati nella tesi consentiranno da una parte di validare gli algoritmi per il controllo attivo del rumore, dall'altra di valutare l'efficacia di abbattimento delle oscillazioni di dispositivi passivi progettati in modo da estendere il range di frequenze in cui questi apportano i massimi benefici di abbattimento. Come noto in letteratura, un difetto dei dispositivi passivi è la loro inefficacia nell'abbattimento di oscillazioni di pressione a bassa frequenza: uno degli obiettivi della tesi è quello di studiare a fondo particolari dispositivi passivi, determinando delle configurazioni geometriche che consentano da una parte di estendere il range di frequenze di funzionamento, dall'altra di valutare la migliore caratteristica geometrica che consenta, attraverso la sua variazione, questa estensione di funzionamento. La variazione della geometria del dispositivo è ottenuta mediante un sistema di attuazione che trasforma il dispositivo passivo in un dispositivo semi-attivo. Le particolari configurazioni geometriche che verranno studiate e l'utilizzo del sistema attivo per la variazione della caratteristica geometrica del dispositivo, dovranno consentire l'estensione del range di funzionamento verso le basse frequenze, al fine di coprire il campo di frequenze dove i sistemi di controllo attivo risultano inadeguati.

Nel **Capitolo 1** verranno analizzati la fisica della propagazione del suono, e verrà analizzata la percezione del suono da parte dell'uomo e come questa influisce sulle tecniche di abbattimento del rumore. Dopo una breve delucidazione dei principali elementi di psicoacustica, verrà affrontata la teoria di propagazione del suono, concentrando l'attenzione sulla propagazione delle onde piane in ambienti che favoriscono la propagazione lungo una direzione preferenziale, ossia una propagazione del rumore di tipo monodimensionale. In questi ambienti verranno delineati i processi necessari per realizzare l'abbattimento dell'onda propagante secondo due meccanismi: il meccanismo di cancellazione e il meccanismo di assorbimento. Infine verrà dato spazio allo stato dell'arte delle soluzioni disponibili per l'abbattimento del rumore, eseguendo una classificazione delle tecniche di abbattimento in tecniche passive e attive, evidenziandone pregi e difetti.

Nel **Capitolo 2** verrà applicata la teoria di propagazione delle onde, esposta nel capitolo 1, ad un condotto e si analizzerà l'andamento delle oscillazioni di pressione al variare della loro frequenza, mettendo in evidenza le particolari caratteristiche proprie della propagazione delle onde in un condotto. L'analisi teorica verrà affiancata da quella numerica che consente di validare il modello analitico. Nella seconda parte del capitolo verranno presentate le tecniche per effettuare il controllo attivo del rumore, in particolare si presenterà il sistema di controllo con i suoi principali elementi: gli attuatori, i sensori e infine l'hardware e il software per implementare gli algoritmi di controllo. Di questi elementi si presenteranno le principali caratteristiche che devono essere considerate per effettuare una corretta scelta, caratteristiche che variano in funzione delle performance e della tipologia del sistema di controllo attivo che si intende realizzare. Infine verranno presentati gli algoritmi di controllo implementati che saranno testati in uno degli apparati sperimentali realizzati.

Nel **Capitolo 3** verrà presentato il progetto dell'apparato sperimentale per il test degli algoritmi di controllo implementati nel capitolo 2. Il progetto consiste nella scelta della geometria e nel dimensionamento dell'ambiente acustico che consente di realizzare il controllo attivo del rumore su onde piane che si propagano lungo una direzione, quindi la progettazione di un ambiente acustico che permette la propagazione monodimensionale delle onde piane. Nella seconda fase del progetto si sceglieranno i sensori, gli attuatori e la piattaforma hardware e software per l'esecuzione dei codici di controllo che implementano gli algoritmi realizzati in precedenza. Con la scelta di questi elementi si procederà alla caratterizzazione dell'impianto, alla valutazione e al confronto delle performance di abbattimento del rumore ottenibili mediante gli algoritmi presentati nel capitolo 2.

Nel **Capitolo 4** verrà presentato il controllo indiretto per l'abbattimento delle oscillazioni di pressione. In questo ambito si presenteranno le tecniche di abbattimento passivo, basate sull'utilizzo di dispositivi risonanti che consentono di abbattere il rumore e, più in generale, le oscillazioni di pressione anche in un campo di frequenze dove le tecniche di controllo attivo mostrate nel primo apparato sperimentale possono risultare inadeguate. Nel presente capitolo verrà ripresentata la teoria di propagazione delle onde per giungere allo sviluppo di un metodo sistematico per la scrittura delle equazioni che governano la propagazione delle onde in condotti in cui vi sono delle appendici ad essi collegate. Tali appendici, opportunamente progettate, consentiranno di realizzare l'abbattimento passivo delle oscillazioni di pressione senza l'utilizzo di energia

proveniente dall'esterno. Ottenuto il metodo sistematico di analisi, questo verrà impiegato per l'analisi del comportamento dinamico e dell'abbattimento delle oscillazioni di pressione ottenibili con due tipologie di dispositivi risonanti: i risonatori side-branch e i risonatori di Helmholtz. Note le equazioni che governano la propagazione delle onde in questi dispositivi, verranno analizzate dal punto di vista teorico le performance di abbattimento al variare delle caratteristiche geometriche dei risonatori, al fine di stabilire i range di frequenze di funzionamento e i criteri di buona progettazione. I risultati ottenuti dal modello analitico saranno validati grazie ad una analisi numerica, mediante lo studio del sistema con una modellazione ad elementi finiti.

Nel **Capitolo 5** si procederà al progetto dell'apparato sperimentale per il test dei dispositivi risonanti presentati nel capitolo 4. In particolare, al fine di realizzare un adeguato apparato sperimentale, verrà analizzato il comportamento delle oscillazioni di pressione all'interno di un impianto costituito da un compressore industriale alternativo per la compressione di aria. Nei condotti alimentati dal compressore industriale il flusso di aria presente in essi non è continuo ma soggetto ad oscillazioni dettate dal movimento alterno dei pistoni dei singoli stadi di compressione. A partire dai risultati ottenuti dall'impianto industriale si procederà al progetto di un apparato sperimentale per il test dei dispositivi risonanti presentati in precedenza.

Nel **Capitolo 6** verranno progettati e testati sperimentalmente sull'impianto di test presentato nel capitolo 5, alcuni dispositivi risonanti di tipo side-branch ed Helmholtz a geometria fissa, allo scopo di determinare sperimentalmente le performance di abbattimento di questi sistemi. L'analisi sperimentale consentirà inoltre di valutare le caratteristiche geometriche di questi dispositivi che possono essere modificate da un sistema di attuazione, al fine di adattare alcuni parametri geometrici in funzione della reale oscillazione presente all'interno dei condotti in cui vengono inseriti. Dall'analisi sperimentale verrà scelta la grandezza geometrica che può essere variata. Nella seconda parte del capitolo si affronterà la progettazione del dispositivo risonante a geometria variabile, a partire dalla progettazione della trasmissione, alla scelta degli attuatori e dei sensori, ed infine del circuito elettrico per l'implementazione di un sistema di controllo. Infine si procederà alla definizione e all'implementazione degli algoritmi di controllo e al test sperimentale delle performance di abbattimento del dispositivo risonante a geometria variabile.

La tesi si conclude con sette appendici. Nell'**Appendice A** si riporteranno i complementi analitici come la teoria per la stima delle funzioni di trasferimento di un

sistema e la teoria di minimizzazione basata sul metodo del gradiente: questo metodo sarà alla base degli algoritmi di controllo attivo del rumore implementati nel capitolo 2. A seguire verrà presentata la tecnica di *Cooley-Tukey* per l'analisi spettrale di un segnale e mostrata la similitudine tra la teoria di propagazione delle onde affrontata nel controllo attivo del rumore e la teoria utilizzata nell'abbattimento con i dispositivi passivi.

Nell'**Appendice B** si riporteranno in dettaglio le caratteristiche dei sensori, degli attuatori e dell'hardware utilizzato per l'implementazione dei sistemi di controllo utilizzati negli apparati sperimentali presentati nella tesi.

Nell'**Appendice C** si riporteranno gli script di MatLab utilizzati per la realizzazione del sistema di controllo attivo del rumore.

Nell'**Appendice D** si riporteranno i codici di controllo realizzati nel linguaggio *ANSI C* che implementano gli algoritmi per il controllo attivo del rumore presentati nel capitolo 3.

Nell'**Appendice E** si riporteranno gli script di MatLab utilizzati nello sviluppo del secondo apparato. In particolare si riporterà lo script MatLab che implementa uno dei modelli analitici ottenuti nel capitolo 4, mediante l'utilizzo dell'approccio sistematico per la scrittura delle equazioni che governano un sistema costituito da un condotto e da un risonatore di Helmholtz ad esso collegato. Si riporterà inoltre lo script MatLab realizzato per l'esecuzione della campagna di misura per l'analisi delle oscillazioni di pressione all'interno dei condotti del compressore alternativo industriale, i cui risultati sono mostrati nel capitolo 5.

Nell'**Appendice F** si riporteranno alcuni disegni di assieme relativi al dispositivo risonante a geometria variabile progettato nel capitolo 6.

Nell'**Appendice G** si riporteranno i codici di controllo realizzati nel linguaggio *ANSI C* che implementano gli algoritmi per il controllo del dispositivo risonante a geometria variabile progettato nel capitolo 6.



## Capitolo 1

# La propagazione e il controllo delle onde di pressione

### 1.1 Suono e rumore

Nella vita di tutti i giorni è possibile sperimentare la sensazione legata al senso dell'udito. La percezione del suono è una delle sensazioni che permettono all'essere umano di percepire la natura circostante e di interagire con essa. È quindi necessario, fin dal principio dello studio dei fenomeni legati al suono, tenere in considerazione la natura soggettiva legata a tale percezione. Accanto quindi allo studio relativo alla fisica dei problemi legati alla generazione e propagazione del suono, occorre considerare anche la percezione del suono da parte dell'essere umano attraverso il sistema uditivo.

Il suono è la sensazione prodotta all'orecchio dalle piccole oscillazioni di pressione presenti nel mezzo circostante. Questa sensazione è prodotta in risposta alle vibrazioni della membrana timpanica alle variazioni di pressione. L'ampiezza delle variazioni di pressione viene detta pressione sonora o pressione acustica. Le oscillazioni di pressione sono provocate da movimenti vibratorii indotti da una sorgente; questa può essere costituita da una macchina, un apparecchio in generale, oppure dalle vibrazioni del cono di una cassa acustica. Le vibrazioni della sorgente vengono trasmesse alle particelle adiacenti grazie alle proprietà meccaniche del mezzo di propagazione (ad esempio l'aria). Le particelle iniziano a loro volta ad oscillare trasmettendo il movimento alle altre particelle, poi queste a quelle a loro adiacenti e così di seguito. Il semplice movimento vibratorio del cono si propaga meccanicamente dando luogo all'*onda sonora*.



Figura 1.1: *Onde di pressione prodotte da una superficie piana vibrante.*

Per meglio illustrare il principio di propagazione delle onde si riporta l'esempio di una superficie piana oscillante (fig. 1.1). Se la superficie vibrante si trova allo stato di equilibrio, anche l'aria prossima alla superficie dovrà trovarsi in condizioni di equilibrio alla pressione atmosferica. Si supponga ora che la superficie si deformi, incurvandosi verso destra. L'aria inizialmente di fronte alla superficie viene ora compressa di una piccola quantità. Il numero di molecole di aria di fronte alla superficie vibrante è la stessa, ma il volume occupato da queste è inferiore a causa della presenza della superficie incurvata. Questo porta alla nascita di una fluttuazione di pressione che tende a spostare la superficie vibrante facendola passare dalla posizione di equilibrio, ma per effetto forze di inerzia questa viene superata e viene raggiunta una nuova posizione, in cui la superficie è incurvata verso l'sinistra, dove la velocità dei punti sulla superficie vibrante è nulla. Ora l'aria all'interno è rarefatta, nasce quindi un'ulteriore fluttuazione di pressione che spinge la superficie a riportarsi nella condizione di equilibrio e ricreare uno stato di compressione.

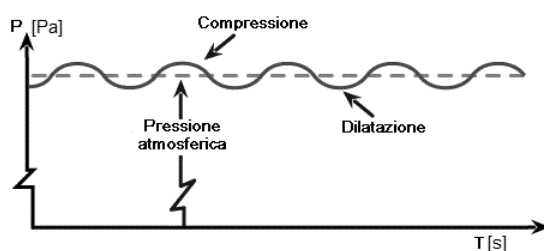


Figura 1.2: *Andamento della pressione sonora rilevata in un punto vicino alla superficie vibrante.*

Andando a misurare l'andamento della pressione in un punto vicino alla superficie vibrante, si otterrebbe un grafico simile a quello in figura 1.2, dove si succedono una serie di stati di equilibrio, compressione, equilibrio, rarefazione. Questa figura assomiglia molto ad un'onda; in particolare viene definita *forma d'onda* la ripetizione di un



determinato andamento della pressione nel tempo. Il movimento della membrana causa quindi compressione e rarefazione dell'aria, che corrispondono a variazioni di pressione acustica: oscillazioni di questo tipo vengono definite longitudinali<sup>1</sup>.

La distribuzione della pressione lungo la direzione di propagazione del suono viene definita *onda sonora*. Nel seguito (par. 1.2) verrà determinata l'equazione delle onde che permette di analizzare il fenomeno di propagazione delle onde e costruire modelli matematici sul comportamento fisico di sistemi reali.

Per avere un'idea dell'entità di queste variazioni di pressione rispetto alla pressione statica dell'ambiente, si pensi al rapporto che esiste tra la pressione media sul fondo di un oceano e la pressione puntuale valutata considerando le fluttuazioni di pressione causate dalle onde superficiali (fig. 1.3). Da ciò si può dedurre quanto le variazioni di pressione dovute alle onde sonore siano piccole, e quanto sia sensibile l'orecchio umano che consente di rilevarle. La pressione sonora è dunque la variazione di pressione rispetto ad un valore di riferimento causato da una perturbazione; il valore della pressione di riferimento viene spesso assunto pari al valore della pressione atmosferica. L'unità di misura della pressione nel sistema internazionale è il Pascal ( $Pa$ )[4]. Risulta utile dare fin da subito delle definizioni di grandezze che verranno utilizzate nel presente studio: si definiscono i concetti di *pressione sonora istantanea*, *pressione sonora efficace* e *livello di pressione sonora*.

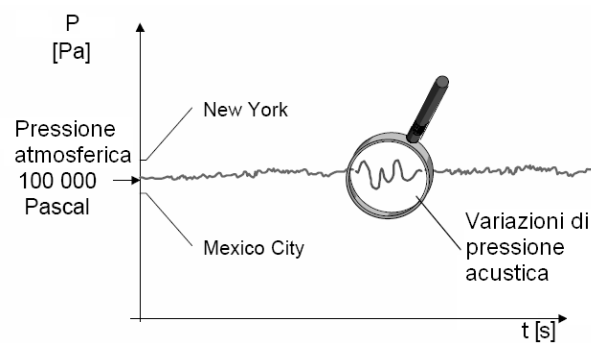


Figura 1.3: *Variazione della pressione atmosferica assoluta in relazione alle onde sonore.*

<sup>1</sup>Le oscillazioni longitudinali sono tipiche della trasmissione del suono nell'aria. Esistono anche oscillazioni in direzione perpendicolare a quella di propagazione del suono, ad esempio, nel caso di una corda percossa, la deformazione che è perpendicolare alla corda, si propaga lungo la corda stessa.

### Pressione sonora istantanea

La pressione sonora istantanea  $p_i$  è la variazione rispetto alla pressione  $p_0$  dell'ambiente, causata da un'onda sonora in una data posizione ed in un dato istante temporale:

$$p_i = \frac{dF}{dA} = \frac{F - F_0}{A - A_0}, \quad (1.1)$$

dove  $F$  è la forza in Newton,  $A$  è l'area in metri su cui agisce la forza  $F$  e il pedice "0" indica i valori di riferimento. La pressione totale  $p_{tot}$  è data da:

$$p_{tot} = p_0 + p_i. \quad (1.2)$$

### Pressione sonora efficace

La pressione sonora efficace  $p_{eff}$  è il valore quadratico medio della pressione sonora istantanea in tutto l'intervallo temporale  $T$ .

$$p_{eff} = \sqrt{\frac{1}{T} \int_0^T p_i^2 dt}. \quad (1.3)$$

### Livello di pressione sonora

La definizione di base del suono è in termini di ampiezza della componente fluttuante della pressione all'interno del mezzo fluido. Viene pertanto definito il *livello di pressione sonora SPL* come:

$$SPL = 10 \log_{10} \frac{p^2}{p_0^2} = 20 \log_{10} \frac{p}{p_0} \text{ dB}. \quad (1.4)$$

Si utilizza il valore quadratico medio delle componenti fluttuanti della pressione poiché la maggior parte dei suoni sono segnali random. Il *SPL* è una misura logaritmica della pressione sonora efficace di onda sonora rispetto ad una sorgente sonora di riferimento. Il valore di  $p_0$  è preso come standard di riferimento pari a  $p_0 = 20 \mu Pa$ , rispetto al quale vengono confrontate le pressioni nell'equazione 1.4. Questo valore non è stato scelto in modo arbitrario, ma rappresenta la soglia di udibilità media dell'essere umano per un suono alla frequenza di  $1000 \text{ Hz}$  che si propaga nell'aria<sup>2</sup>. Il livello di  $0 \text{ dB}$ , corrispondente ad una pressione  $p = p_0$ , è stato definito come la più bassa fluttuazione di pressione che l'uomo riesce normalmente a distinguere.

<sup>2</sup>Per i valori dei livelli di riferimento della pressione in mezzi diversi dall'aria si faccia riferimento alla normativa ANSI S1.1-1994.

Il rumore è un suono non desiderato. Ciò che viene considerato rumore è perciò soggettivo: per un soggetto una canzone può essere considerata un suono, mentre per altre persone può essere fastidiosa e quindi associata a rumore. Un gruppo di suoni tipicamente associati ad una sensazione di rumore sono quelli prodotti da una fabbrica: punzonatrici, seghe, linee di produzione, ecc. Con il passare degli anni si sta sempre maggiormente affermando, sia in ambito civile che industriale, la necessità di regolamentare le emissioni sonore per la tutela della salute. A tale scopo sono state emanate diverse normative per la misura delle grandezze principali in ambito acustico e su di esse si basano le leggi che definiscono i limiti di emissione da rispettare. Anche in ambito normativo si considerano gli aspetti legati alla percezione del suono dell'essere umano e dunque, al fine di comprenderne in fondo l'applicabilità, è necessario disporre delle conoscenze di base nel campo dell'acustica. Nella società attuale si è presa coscienza dei problemi riscontrati nei soggetti sottoposti per lunghi periodi di tempo a rumori oltre certe soglie, o ai problemi riscontrati nei soggetti sottoposti per brevi periodi di tempo a rumori di elevata intensità; pertanto si rende necessario adottare politiche di salvaguardia della salute dei lavoratori indagando sulle sorgenti di rumore e cercando soluzioni per abbattere il rumore emesso da tali sorgenti.

In questo senso sono nate e si stanno attualmente sviluppando tutte le tecniche per la misura dell'emissione sonora e per l'identificazione della sorgente di rumore. La necessità di sviluppare nuove tecniche di misura non può prescindere dalla conoscenza del fenomeno fisico, è quindi necessario poter descrivere i fenomeni attraverso relazioni che contengano grandezze la cui stima possa essere effettuata attraverso opportune campagne di misura. Ad esempio dalla misura della pressione sonora si può passare alla misura di grandezze più complesse, come ad esempio l'intensità sonora che considera anche la velocità delle particelle, ovvero grandezze vettoriali il cui contenuto informativo è maggiore e che consentono di poter effettuare analisi accurate sul comportamento sonoro dei componenti oggetto di studio.

La generazione del rumore è in genere associata alla presenza di più sorgenti, si pensi ad esempio ad una struttura vibrante che trasmette le vibrazioni all'ambiente circostante le quali si comportano come una sorgente di rumore, come mostrato in figura 1.4. Le vibrazioni generate dalle masse in moto alterno del sistema di condizionamento si propagano alle fondazioni e alle pareti circostanti. La vibrazione delle pareti genera onde di pressione che si traducono in rumore percepito all'interno della stanza. Un altro esempio è rappresentato in figura 1.5 dove i moti turbolenti dell'aria indotti da alcuni

meccanismi possono generare rumore. Il movimento delle ventole causa delle turbolenze e delle variazioni di pressione nell'aria circostante che vengono avvertite come rumore.

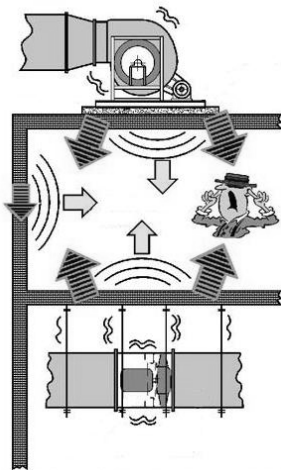


Figura 1.4: *Rumore generato da vibrazioni.*

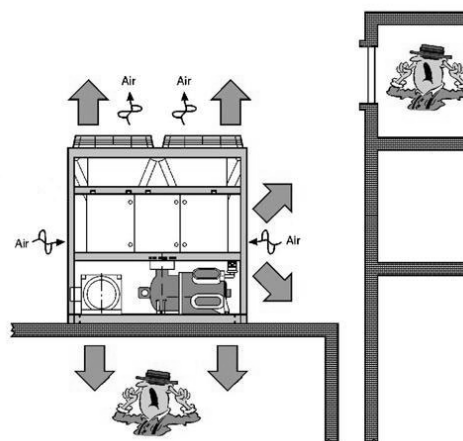


Figura 1.5: *Rumore generato da effetti aerei.*

Dal punto di vista acustico, le situazioni appena descritte possono essere studiate con metodi analitici, numerici, o mediante misure sperimentali. Nel caso in cui si adottino metodi analitici o numerici occorre descrivere il fenomeno analizzato mediante un modello opportuno. La definizione di un modello per il fenomeno di interesse deve poter abbracciare tutti gli aspetti del fenomeno che si intendono considerare, con l'eventuale aumento della complessità del modello che ne può derivare. Nel seguito (par. 1.2) verranno definite le relazioni fisiche che descrivono la propagazione delle onde di pressione, utili per la successiva definizione dei modelli di interesse.

### 1.1.1 Elementi di psicoacustica

La psicoacustica è lo studio della percezione soggettiva umana dei suoni. Più precisamente è lo studio della psicologia della percezione acustica. Come già affermato, i suoni sono prodotti dalle vibrazioni di oggetti che inducono delle vibrazioni nelle molecole di aria in prossimità di essi, questo genera una differenza di pressione e quindi il propagarsi di onde di pressione nell'ambiente circostante: le onde viaggiano con una certa velocità di propagazione che dipende dal mezzo di trasmissione.

Si possono attribuire alcune caratteristiche fondamentali alle onde sonore:

- Ampiezza: grandezza del profilo di variazione di pressione dell'onda sonora.

- Intensità: quantità di energia acustica per unità di area.
- Frequenza: numero di cicli, in termini di variazione di pressione, che si ripetono in un secondo.

L'udito degli esseri umani è sensibile ad un ampio intervallo di frequenze, da circa  $20\text{ Hz}$  a  $20\text{ kHz}$ , e ad un ampio intervallo di intensità, si pensi che il rapporto tra il volume più basso percepibile e quello più alto è quasi  $10^{-6}$ . Data questa grande distanza tra gli estremi del campo udibile, i livelli del suono vengono misurati su una scala logaritmica, le cui unità di misura sono i decibel ( $dB$ ). Essendo una scala logaritmica, cambiamenti relativamente piccoli in decibel possono corrispondere a cambiamenti fisici molto più elevati: un incremento di  $6\text{ dB}$  corrisponde ad esempio a raddoppiare la pressione sonora.

I toni puri sono composti da una sola riga spettrale, in corrispondenza della frequenza a cui viene generata l'onda. Un suono complesso può essere descritto mediante la scomposizione in serie di Fourier, e lo spettro del segnale risulta composto da più righe spettrali. Se si analizza lo spettro del segnale, che risulta essere discreto, si possono evidenziare due aspetti:

- Prima armonica: è la componente fondamentale più bassa del suono.
- Armoniche secondarie: sono le armoniche che presentano frequenze multiple della frequenza dell'armonica fondamentale, e sono derivate dallo sviluppo in serie di Fourier dell'andamento della pressione sonora.

Se si considera la fisiologia dell'orecchio umano, l'ampiezza dell'oscillazione del timpano dovuta a un suono puro determina l'ampiezza dello scostamento della membrana basilare e questa a sua volta determina la sensazione di intensità del suono. L'ampiezza dell'oscillazione del timpano è in stretta relazione con i parametri fisici di variazione di pressione e di flusso di potenza acustica, o intensità.

L'orecchio è sensibile ad un range dinamico estremamente ampio, e l'impraticità della scala lineare ha portato alla definizione di una scala logaritmica espressa in decibel. Vi sono evidenti vantaggi connessi all'utilizzo di questa scala: una scala logaritmica offre innanzitutto una notevole compressione dei valori, inoltre è una scala relativa, perché i valori sono riferiti ad un valore di riferimento che è la soglia di udibilità.

Facendo riferimento a quanto riportato nel paragrafo 1.1, si è introdotta una scala logaritmica che definisce il livello di intensità acustica in funzione del livello di pressione sonora (o sound pressure level, SPL). Sperimentalmente si osserva che suoni puri

continui, caratterizzati da stesso SPL ma a frequenze diverse, producono sensazioni diverse di intensità. Questo indica che l'SPL non è una buona misura dell'intensità percepita se si confrontano suoni puri a frequenza diversa. È stato dunque necessario determinare sperimentalmente i valori di eguale intensità percepita (*curve isofoniche* o *curves of equal loudness*) in base alle risposte dei neuroni posizionati lungo tutta la coclea, al variare della frequenza considerando come riferimento l'SPL a 1000 Hz. In stretta relazione con le curve isofoniche è la definizione di *Phon* (o *Loudness Level, LL*): il *LL* di un suono a frequenza  $f$  è dato dal SPL di un suono a 1000 Hz che determina la stessa percezione di intensità [3].

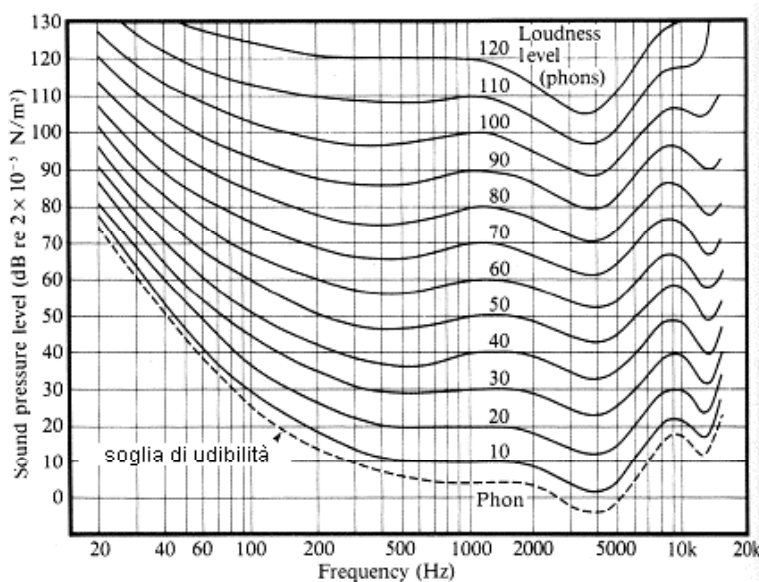


Figura 1.6: *Curve isofoniche.*

Come mostrato in figura 1.6, dire dunque che un suono è a 80 *Phon* vuol dire fornire i valori della curva di isofonia relativa a 80 *SPL* per tutte le frequenze. Da questi audiogrammi si può notare come le curve isofoniche non siano parallele tra loro, specialmente alle basse frequenze; la soglia di massima sensibilità è attorno ai 3000 ÷ 4000 Hz, e corrisponde ad un livello inferiore rispetto a quello ottenuto a 1000 Hz. Si noti che per produrre la stessa sensazione di intensità alle basse frequenze è necessaria molta più energia di quella necessaria per produrre la stessa sensazione alla frequenza di riferimento. Si può notare ancora, osservando la curva che rappresenta la soglia dell'udibile, che la sensibilità dell'orecchio diminuisce notevolmente alle basse frequenze e alle frequenze superiori a 6000 Hz. L'audiogramma, mostrato in figura 1.7 definisce l'area di

udibilità, compresa tra la soglia di udibilità e la soglia di dolore. Il limite superiore è rappresentato dalla soglia di dolore<sup>3</sup>, mentre il limite inferiore è rappresentato dalla minima soglia di udibilità: entrambe le curve appartengono al diagramma delle curve isofoniche e, come si può osservare dalla figura, comprendono tutta la gamma di frequenze e livelli di pressione sonora della musica.

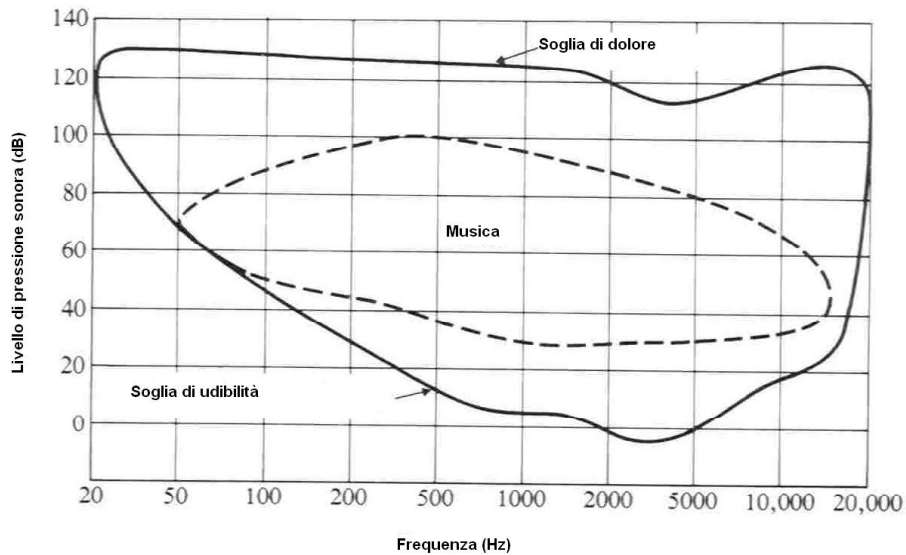


Figura 1.7: Gamma di frequenze e livelli di pressione sonora dell'orecchio umano.

Per comprendere meglio il significato della figura 1.7, nella tabella 1.1 si riportano alcuni esempi di rumori presenti in natura con il corrispondente valore di pressione sonora e livello di pressione sonora [1].

Sorgente	Pressione sonora [ $Pa$ ]	SPL [ $dB$ ]
Limite teorico (1)	101325	194.0937
Esplosione vulcano Karakatoa a 160 km	20000	180
Semplice motore termoacustico (2)	12000	176
Fucile M1 Garand a 1 m	5000	168
Jet a 30 m	630	150
Fucile a 1 m	200	140
Soglia di dolore	100	130
Danni all'udito (3)	20	$\approx 120$

<sup>3</sup>La soglia di dolore si presenta qualora il livello di pressione sonora supera un certo livello, attorno ai 120 dB; al di sopra di questo valore l'orecchio comincia a percepire dolore fisico e per esposizioni prolungate si possono generare danni non reversibili.

Jet a 100 m	$6 \div 200$	$110 \div 120$
Martello pneumatico a 1 m	2	$\approx 100$
Danni all'udito (4)	$6 \cdot 10^{-1}$	$\approx 85$
Strada trafficata a 10 m	$2 \cdot 10^{-1} \div 6 \cdot 10^{-1}$	$80 \div 90$
Automobile a 10 m	$2 \cdot 10^{-2} \div 2 \cdot 10^{-1}$	$60 \div 80$
TV a 1 m	$2 \cdot 10^{-2}$	$\approx 60$
Conversazione	$2 \cdot 10^{-3} \div 2 \cdot 10^{-2}$	$40 \div 60$
Stanza silenziosa	$2 \cdot 10^{-4} \div 6 \cdot 10^{-4}$	$20 \div 30$
Respiro	$6 \cdot 10^{-5}$	10
Soglia di udibilità	$2 \cdot 10^{-5}$	0

Tabella 1.1: Esempi di livelli di pressione sonora e livello di pressione sonora.

Occorrono delle precisazioni per alcune voci, indicate con numeri tra parentesi, della tabella 1.1:

1. Il limite teorico è il limite oltre il quale le onde sonore subiscono una distorsione. Il valore riportato in tabella 1.1 è stato calcolato alla pressione atmosferica pari a  $1 \text{ Atm} = 101325 \text{ Pa}$ . Le onde sonore sono costituite da una compressione che si svolge entro metà del ciclo e da una rarefazione che si svolge nella seconda metà del ciclo. Quando la pressione nella parte di compressione raggiunge valori doppi rispetto alla pressione atmosferica, la pressione nella parte di rarefazione raggiunge valori prossimi al vuoto. A questo punto, per incrementare il livello di pressione sonora è necessario aumentare la pressione nella parte di compressione. Qualsiasi onda sonora che raggiunge questi livelli di pressione non viene più considerata un'onda sonora, ma un'onda d'urto. Queste onde possono svilupparsi ad esempio durante il lancio di missili, nei boom sonici, nella deflagrazione di esplosivi, nei tuoni, nei terremoti e nelle esplosioni vulcaniche.
2. Con questa voce si fa riferimento alle pompe di calore soniche o alle pompe di calore termoacustiche. Questi dispositivi utilizzano onde sonore in un gas pressurizzato per trasferire calore, oppure utilizzano il calore fluente per convertirlo in elettricità attraverso casse acustiche piezoelettriche.



3. I danni all'udito vengono generati dopo un breve tempo di esposizione.
4. I danni all'udito vengono generati dopo un lungo tempo di esposizione.

L'udito umano presenta una gamma di frequenze da  $20\text{ Hz}$  a  $20\text{ kHz}$ , ed è molto sensibile a  $2700\text{ Hz}$  (la prima frequenza di risonanza del canale uditivo) mentre lo è meno alle basse frequenze. Per regolare la risposta di una misura di livello di pressione sonora vengono utilizzate delle curve di ponderazione nel dominio della frequenza. Nella tabella 1.3 vengono mostrati i valori delle curve di ponderazione A e C al variare della frequenza del suono. In generale sono normalizzate tre curve di ponderazione indicate con A, B e C, approssimate a tre curve isofoniche e impiegate a seconda del livello di pressione sonora misurata. La curva più utilizzata è quella di tipo A e il segnale ponderato con questa curva viene espresso tramite la notazione  $dB(A)$  o  $dB(A)$ . Questa pesatura approssima la risposta dell'orecchio umano e modera il livello di pressione sonora, soprattutto alle basse frequenze. La curva di ponderazione di tipo C invece approssima la risposta dell'orecchio umano agli alti livelli di pressione sonora, pertanto presenta un minore abbattimento della pressione sonora percepita sia alle basse che alle alte frequenze. Nonostante questa finestra ponderi correttamente il livello di pressione sonora per alti livelli di pressione sonora, molto spesso viene utilizzata la finestra di pesatura di tipo A perché comparabile con gli standard internazionali di misura.

Frequenze di centro-ottava [ $Hz$ ]	Risposta pesata [ $dB$ ]	
	Pesatura A	Pesatura C
31.5	-39.4	-3
63	-26.2	-0.8
125	-6.1	-0.2
250	-8.6	0
500	-3.2	0
1000	0	0
2000	1.2	-0.2
4000	1	-0.8
8000	-1.1	-3
16000	-6.6	-8.5

Tabella 1.3: Curve di ponderazione A e C al variare della frequenza.

## 1.2 Propagazione delle onde piane

Al fine di approssimare correttamente il problema della definizione di dispositivi atti a ridurre le oscillazioni di pressione, occorre conoscere a fondo i principi che stanno alla base della propagazione delle onde in un generico mezzo. La comprensione del fenomeno propagatorio può essere intrapresa studiando l'equazione delle onde. Per applicare correttamente l'equazione ai dispositivi meccatronici sviluppati in questa tesi è necessario determinare l'equazione delle onde per propagazione di tipo monodimensionale poiché risulta più semplice la comprensione dei fenomeni fisici ed è quindi possibile, grazie ad una migliore comprensione del fenomeno, realizzare sistemi avanzati per l'abbattimento.

Per determinare l'equazione delle onde è necessario scrivere innanzitutto le relazioni fondamentali che legano tra loro pressione e velocità della particella di fluido, e le leggi che descrivono come queste grandezze variano nello spazio e nel tempo [5]. Si applicano quindi l'equazione di Eulero, l'equazione di continuità e l'equazione di stato termodinamico del fluido in cui si propagano le onde.

L'equazione di Eulero coincide fisicamente con la seconda legge della dinamica, che fornisce la relazione tra pressione, velocità e densità attraverso l'equilibrio dinamico. Scrivendo il bilancio di forze sul generico elemento di fluido si ottiene:

$$\vec{dF} = \vec{a} dm, \quad (1.5)$$

dove l'accelerazione  $\vec{a}$  è ottenuta dal limite del rapporto incrementale della velocità  $\vec{v}$  della particella, dopo aver sviluppato quest'ultima in serie di Taylor:

$$\vec{a} = \lim_{dt \rightarrow 0} \frac{\vec{v}(x + dx, y + dy, z + dz, t + dt) - \vec{v}(x, y, z, t)}{dt}. \quad (1.6)$$

Sviluppando in serie di Taylor si ottiene:

$$\begin{aligned} \vec{v}(x + dx, y + dy, z + dz, t + dt) &= \\ &= \vec{v}(x, y, z, t) + \frac{\partial \vec{v}}{\partial x} v_x dt + \frac{\partial \vec{v}}{\partial y} v_y dt + \frac{\partial \vec{v}}{\partial z} v_z dt + \frac{\partial \vec{v}}{\partial t} dt, \end{aligned} \quad (1.7)$$

quindi:

$$\vec{a} = \frac{\partial \vec{v}}{\partial x} v_x + \frac{\partial \vec{v}}{\partial y} v_y + \frac{\partial \vec{v}}{\partial z} v_z + \frac{\partial \vec{v}}{\partial t} = \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \vec{\nabla}) \vec{v}. \quad (1.8)$$

La forza netta agente sull'elemento di volume  $dV$  è:

$$\overrightarrow{dF} = \overrightarrow{\nabla} P dV. \quad (1.9)$$

Sapendo che  $dm = \rho dV$ , si può scrivere l'equazione esatta non lineare:

$$-\overrightarrow{\nabla} P dV = \rho dV \left\{ \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \overrightarrow{\nabla}) \vec{v} \right\}. \quad (1.10)$$

L'equazione 1.10 si può linearizzare per piccole variazioni di densità e di pressione (ipotesi valida nel campo di propagazione acustica). Inoltre si può effettuare l'ipotesi di flusso non soggetto a turbolenza, ovvero che sia verificata la seguente disuguaglianza:

$$|(\vec{v} \cdot \overrightarrow{\nabla}) \vec{v}| \ll \left| \frac{\partial \vec{v}}{\partial t} \right|. \quad (1.11)$$

Assumendo  $P = P_0 + p$  e  $\rho = \rho_0 + \rho'$ , si può scrivere l'equazione di Eulero linearizzata:

$$\overrightarrow{\nabla} P = -\rho_0 \frac{\partial \vec{v}}{\partial t}. \quad (1.12)$$

L'*equazione di continuità* esprime il principio di conservazione della massa, ovvero afferma che la differenza tra la massa entrante e la massa uscente dal sistema deve essere uguale alla variazione di massa complessiva del sistema stesso:

$$\Delta M = M_{in} - M_{out}. \quad (1.13)$$

In un caso monodimensionale si ottiene:

$$\frac{\partial \rho A dx}{\partial t} = (\rho v A)_x - (\rho v A)_{x+dx} = -\frac{\partial(\rho v A)_x}{\partial x} dx, \quad (1.14)$$

e semplificando:

$$\frac{\partial \rho}{\partial t} = \frac{\partial(\rho v)_x}{\partial x}. \quad (1.15)$$

La relazione 1.15 si può estendere al caso di propagazione in un generico campo a più dimensioni:

$$\frac{\partial \rho}{\partial t} + \overrightarrow{\nabla} \cdot \rho \vec{v} = 0. \quad (1.16)$$

L'equazione così ottenuta è non lineare ed è possibile linearizzarla per piccole variazioni di densità. Definendo  $\delta = \frac{\rho - \rho_0}{\rho_0}$  si può procedere alla linearizzazione:

$$\frac{\partial \delta}{\partial t} = -\overrightarrow{\nabla} \cdot \vec{v}. \quad (1.17)$$

L'*equazione di stato* esprime il comportamento termodinamico del fluido. Poiché le variazioni di stato associate alla propagazione delle onde sonore si possono considerare adiabatiche, vale la relazione  $PV^\gamma = cost$  con  $\gamma = \frac{C_p}{C_v}$ :

$$P = p_0 \left( \frac{\rho}{\rho_0} \right)^\gamma. \quad (1.18)$$

Differenziando e linearizzando la relazione 1.18 per piccole variazioni di densità si ottiene:

$$dP = \gamma p_0 \left( \frac{\rho}{\rho_0} \right)^{\gamma-1} \approx \gamma p_0 \frac{\rho - \rho_0}{\rho_0} = \gamma p_0 \delta. \quad (1.19)$$

Derivando l'equazione linearizzata 1.19 rispetto al tempo si ottiene l'equazione di stato:

$$\frac{1}{p_0} \frac{\partial p}{\partial t} = \gamma \frac{\partial \delta}{\partial t}. \quad (1.20)$$

Avendo definito l'*equazione di Eulero*, l'*equazione di continuità* e l'*equazione di stato*, è possibile ora relazionarle e definire la cosiddetta *equazione delle onde*. Mettendo in relazione l'equazione di continuità e di stato si ottiene:

$$\frac{\partial p}{\partial t} = -\gamma p_0 \vec{\nabla} \cdot \vec{v} = -\gamma \text{div} \vec{v}; \quad (1.21)$$

derivando quest'ultima rispetto al tempo si ha:

$$\frac{\partial^2 p}{\partial t^2} = -\gamma p_0 \text{div} \frac{\partial \vec{v}}{\partial t}. \quad (1.22)$$

Applicando l'operatore di divergenza all'equazione di Eulero, è possibile introdurre l'operatore Laplaciano:

$$-\text{div} \vec{\nabla} p = -\nabla^2 p = \rho_0 \text{div} \frac{\partial \vec{v}}{\partial t}. \quad (1.23)$$

Combinando insieme le due relazioni 1.22 e 1.23 si ottiene l'equazione delle onde, la cui unica incognita è la pressione:

$$\nabla^2 p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2}, \quad (1.24)$$

dove:

$$c = \sqrt{\gamma \frac{p_0}{\rho_0}}, \quad (1.25)$$

rappresenta la velocità di propagazione dell'onda sonora nel gas, definita come *velocità del suono*. L'equazione delle onde è un'equazione differenziale del secondo ordine a coefficienti costanti, di tipo iperbolico secondo l'analogia di classificazione delle curve coniche. Essa descrive il comportamento della pressione sonora nello spazio e nel tempo, regola la propagazione di onde sonore in gas omogenei, isotropi e senza perdite dissipative.

L'equazione delle onde 1.24 può essere facilmente riportata al caso di propagazione monodimensionale considerando la seguente relazione:

$$\frac{\partial^2 p}{\partial t^2} - c^2 \frac{\partial^2 p}{\partial x^2} = 0. \quad (1.26)$$

A seconda del tipo di analisi che si intende effettuare è possibile determinare la soluzione dell'equazione delle onde in condizioni di transitorio (*soluzione propagativa*) e in condizioni stazionarie (*soluzione stazionaria*). Di seguito si riporta il procedimento per l'ottenimento della soluzione dell'equazione delle onde nei due casi citati.

L'equazione 1.26 ammette come integrale generale la seguente funzione:

$$p(x, t) = F(x + ct) + G(x - ct), \quad (1.27)$$

dove  $F$  e  $G$  sono due funzioni arbitrarie che descrivono la distribuzione spaziale della pressione sonora nel tempo lungo la direzione  $x$ . Tali funzioni rappresentano entrambe il moto di un'onda viaggiante con velocità  $c$ , in particolare  $F$  rappresenta un'onda che si propaga in direzione delle  $x$  negative (onda progressiva), mentre  $G$  rappresenta un'onda che si propaga nella direzione delle  $x$  positive (onda regressiva). La relazione 1.27 rappresenta la *soluzione propagativa* dell'equazione delle onde, questa è utile soprattutto per descrivere l'andamento della pressione durante i transitori, ossia negli istanti successivi ad una perturbazione. Si può facilmente dimostrare che la relazione 1.27 è soluzione dell'equazione 1.26. Effettuando per comodità un cambio di variabili:

$$\begin{aligned} a &= x + ct, \\ b &= x - ct, \end{aligned} \quad (1.28)$$

sostituendo l'equazione 1.27 nella 1.26 ed esprimendo le derivate parziali come:

$$\begin{aligned} \frac{\partial F}{\partial t} &= \frac{\partial F}{\partial a} \frac{\partial a}{\partial t} \quad \rightarrow \quad \frac{\partial^2 F}{\partial t^2} = \frac{\partial^2 F}{\partial a^2} \left( \frac{\partial a}{\partial t} \right)^2 + \frac{\partial F}{\partial a} \frac{\partial^2 a}{\partial t^2}, \\ \frac{\partial G}{\partial t} &= \frac{\partial G}{\partial b} \frac{\partial b}{\partial t} \quad \rightarrow \quad \frac{\partial^2 G}{\partial t^2} = \frac{\partial^2 G}{\partial b^2} \left( \frac{\partial b}{\partial t} \right)^2 + \frac{\partial G}{\partial b} \frac{\partial^2 b}{\partial t^2}, \end{aligned}$$

$$\begin{aligned}\frac{\partial F}{\partial x} &= \frac{\partial F}{\partial a} \frac{\partial a}{\partial x} &\rightarrow &\frac{\partial^2 F}{\partial x^2} = \frac{\partial^2 F}{\partial a^2} \left(\frac{\partial a}{\partial x}\right)^2 + \frac{\partial F}{\partial a} \frac{\partial^2 a}{\partial x^2}, \\ \frac{\partial G}{\partial x} &= \frac{\partial G}{\partial b} \frac{\partial b}{\partial x} &\rightarrow &\frac{\partial^2 G}{\partial x^2} = \frac{\partial^2 G}{\partial b^2} \left(\frac{\partial b}{\partial x}\right)^2 + \frac{\partial G}{\partial b} \frac{\partial^2 b}{\partial x^2},\end{aligned}$$

dove:

$$\begin{aligned}\frac{\partial a}{\partial t} &= -c &\rightarrow &\frac{\partial^2 a}{\partial t^2} = 0, \\ \frac{\partial b}{\partial t} &= c &\rightarrow &\frac{\partial^2 b}{\partial t^2} = 0, \\ \frac{\partial a}{\partial x} &= 1 &\rightarrow &\frac{\partial^2 a}{\partial x^2} = 0, \\ \frac{\partial b}{\partial x} &= 1 &\rightarrow &\frac{\partial^2 b}{\partial x^2} = 0,\end{aligned}$$

si ottiene che l'equazione 1.27 è soluzione dell'equazione delle onde:

$$c^2 \left( \frac{\partial^2 F}{\partial a^2} + \frac{\partial^2 G}{\partial b^2} \right) - c^2 \left( \frac{\partial^2 F}{\partial a^2} + \frac{\partial^2 G}{\partial b^2} \right) = 0. \quad (1.29)$$

Si consideri a titolo di esempio la funzione  $F(x - ct)$  (analoghe considerazioni possono essere fatte per la funzione  $F(x + ct)$ ), il valore di questa è determinato dall'argomento  $(x - ct)$  e quindi da una serie di valori correlati di  $x$  e  $t$ . Il valore della funzione  $G(x - ct)$  rimane ovviamente lo stesso a pari argomento e quindi un generico incremento  $\Delta x$  produce il medesimo argomento se il tempo viene incrementato di  $\Delta t = \Delta x/c$ , quindi la pressione  $p(x, t)$ , raggiunge lo stesso valore  $\bar{p}$  in differenti punti  $x_1, x_2, \dots, x_n$  in differenti istanti di tempo  $t_1, t_2, \dots, t_n$  correlati da  $\Delta t = \Delta x/c$ :

$$\bar{p} = G(x_1 - ct_1) = G(x_2 - ct_2) = G((x_1 + \Delta x) - c(t_1 + \Delta t)). \quad (1.30)$$

Fisicamente la funzione  $G(x - ct)$  rappresenta un fenomeno propagativo, ossia descrive il profilo di un'onda che si propaga nella direzione delle  $x$  positive con velocità  $c$ . Un discorso analogo vale per la funzione  $F(x + ct)$  che rappresenta invece un'onda che si propaga con velocità  $c$  nella direzione negativa dell'asse  $x$ .

Dovendo invece studiare il comportamento a regime delle variazioni di pressione quando al sistema è applicata una generica forzante, si ha a che fare con un fenomeno non più propagativo, ma stazionario. In un dato punto del sistema la pressione varia con una data ampiezza e una certa frequenza. Imponendo all'equazione 1.26 le condizioni iniziali:

$$p(x, 0) = g(x) = 0, \quad \forall x \in [0, L], \quad (1.31)$$

$$\left( \frac{\partial p}{\partial t} \right)_{x,0} = h(x) = 0, \quad \forall x \in [0, L], \quad (1.32)$$

e le condizioni al contorno, definite secondo le condizioni di *Dirichlet*:

$$p(0, t) = p(L, t) = 0, \quad \forall t, \quad (1.33)$$

è possibile definire una soluzione analitica al problema della propagazione. Considerando la soluzione stazionaria, sarà possibile definire i modi e le pulsazioni dell'ambiente sonoro. Il concetto di modo di vibrare e pulsazione appena definito è analogo a quello che si presenta nell'analisi modale di un sistema meccanico. Ogni sistema vibrante sia esso meccanico, elettrico, o acustico, possiede particolari modi di oscillazione in cui tutte le sue parti componenti vibrano di moto armonico e con la stessa frequenza. Il modo di vibrare acustico rappresenta quindi la deformazione, ossia l'andamento spaziale del modulo della pressione lungo la direzione definita dall'equazione delle onde. Ad ogni modo di vibrare corrisponde una particolare frequenza nota come frequenza propria (o pulsazione propria) del sistema.

Si verifica l'esistenza della soluzione attraverso il metodo di separazione delle variabili. La soluzione  $p(x, t)$  può essere scritta come prodotto di due funzioni indipendenti:

$$p(x, t) = w(t)v(x). \quad (1.34)$$

Definendo le derivate seconde rispetto a  $x$  e a  $t$  dell'ultima relazione trovata, si ha:

$$\frac{\partial^2 p}{\partial t^2} = v(x)\ddot{w}(t), \quad (1.35)$$

$$\frac{\partial^2 p}{\partial x^2} = v''(x)w(t). \quad (1.36)$$

Sostituendo le relazioni appena trovate nell'equazione delle onde 1.26 e separando le variabili si ha:

$$\frac{v''(x)}{v(x)} = \frac{1}{c^2} \frac{\ddot{w}(t)}{w(t)}. \quad (1.37)$$

La relazione precedente è vera solo se i due termini sono uguali ad una costante, pertanto si ha:

$$\frac{v''(x)}{v(x)} = \frac{1}{c^2} \frac{\ddot{w}(t)}{w(t)} = \lambda. \quad (1.38)$$

Si ottiene quindi:

$$v''(x) - \lambda v(x) = 0; \quad (1.39)$$

$$\ddot{w}(t) - \lambda c^2 w(t) = 0. \quad (1.40)$$

Imponendo a  $v(x)$  le condizioni espresse dall'equazione 1.33:

$$v(0) = v(L) = 0, \quad (1.41)$$

si presentano tre casi:

- $\lambda = 0$ ,  $v(x) = A + Bx$ : imponendo le condizioni espresse dall'equazione 1.33 si ha  $A = B = 0$  e si ottiene una soluzione banale;
- $\lambda = \mu^2 > 0$ ,  $v(x) = Ae^{-\mu x} + Be^{\mu x}$ : imponendo le condizioni espresse dall'equazione 1.33 si ha ancora  $A = B = 0$  e si ottiene una soluzione banale;
- $\lambda = -\mu^2 < 0$ ,  $v(x) = A \sin(\mu x) + B \cos(\mu x)$ : imponendo le condizioni espresse dall'equazione 1.33 si ha:

$$v(0) = B = 0, \quad (1.42)$$

$$v(L) = A \sin(\mu L) = 0, \quad (1.43)$$

dove  $A$  può essere scelto arbitrariamente,  $B = 0$  e  $\mu L = m\pi$ ,  $m = 1, 2, \dots$

Solo il terzo caso produce quindi soluzioni non nulle, del tipo:

$$v_m(x) = A_m \sin(\mu_m x),$$

$$\mu_m = \frac{m\pi}{L}.$$

L'equazione definita precedentemente  $\ddot{w}(t) - \lambda c^2 w(t) = 0$  ammette quindi come integrale generale:

$$w_m(t) = C_m \cos(\mu_m ct) + D_m \sin(\mu_m ct), \quad (1.44)$$



si ottengono così soluzioni nella forma:

$$p_m(x, t) = v_m(x)w_m(t) = [a_m \cos(\mu_m ct) + b_m \sin(\mu_m ct)] \sin(\mu_m x), \quad (1.45)$$

da cui risulta che le pulsazioni naturali del sistema sono:

$$\omega_m = \mu_m c = \frac{m\pi}{L} c. \quad (1.46)$$

Le condizioni iniziali sono del tipo:

$$p_m(x, 0) = a_m \sin(\mu_m x), \quad (1.47)$$

$$\frac{\partial p_m}{\partial t}(x, 0) = cb_m \mu_m \sin(\mu_m x). \quad (1.48)$$

La soluzione complessiva dell'equazione delle onde si ottiene mediante il principio di sovrapposizione:

$$p(x, 0) = \sum_{m=1}^{\infty} a_m \sin(\mu_m x) = g(x), \quad (1.49)$$

$$\frac{\partial p}{\partial t}(x, 0) = \sum_{m=1}^{\infty} cb_m \mu_m \sin(\mu_m x) = h(x). \quad (1.50)$$

Assumendo che le funzioni  $g(x)$  e  $h(x)$  siano sviluppabili in serie di Fourier di soli seni, si possono ricavare i termini  $a_m$  e  $b_m$ .

$$a_m = \frac{2}{L} \int_0^L g(x) \sin(\mu_m x) dx; \quad (1.51)$$

$$b_m = \frac{1}{\mu_m c} \frac{2}{L} \int_0^L h(x) \sin(\mu_m x) dx. \quad (1.52)$$

La soluzione stazionaria è dunque nella forma:

$$p(x, t) = \sum_{m=1}^{\infty} [a_m \cos(\mu_m ct) + b_m \sin(\mu_m ct)] \sin(\mu_m x). \quad (1.53)$$

Nota l'equazione delle onde monodimensionale e la sua soluzione è possibile studiare i meccanismi che regolano l'abbattimento, analizzando i fenomeni legati all'interferenza di due onde che si propagano all'interno di un condotto. Per meglio comprendere i principi dell'abbattimento delle onde di pressione si riporta nel paragrafo 1.3 uno studio approfondito che richiama l'attenzione sui due principali meccanismi che si presentano in questo ambito: il *meccanismo di cancellazione* e il *meccanismo di assorbimento*.

Questi meccanismi verranno analizzati considerando la propagazione di onde piane generate da una o più sorgenti monopolo all'interno di un condotto, ossia in un sistema in cui sia valida l'ipotesi di propagazione monodimensionale.

L'ipotesi di monodimensionalità della propagazione delle onde consente di semplificare la trattazione dei meccanismi di abbattimento e di progettare dei dispositivi in cui risulta più semplice la comprensione fisica del problema di abbattimento. I dispositivi che verranno proposti in seguito, sono stati progettati con l'accortezza di evitare la presenza di modi di vibrare trasversali alla direzione di propagazione e il cui contenuto armonico sia al di fuori del range di frequenze che si intende abbattere; in questo modo è possibile considerare trascurabili gli effetti dovuti alle oscillazioni di pressione nelle direzioni ortogonali a quelle di propagazione rendendo quindi valida l'ipotesi di propagazione monodimensionale.

### 1.3 Il processo di abbattimento delle onde piane

Per comprendere le limitazioni associate all'implementazione di un sistema di abbattimento delle onde di pressione nel paragrafo 1.2 si affrontato dal punto di vista analitico lo studio della propagazione delle onde[6]. Volendo affrontare lo studio di un problema di propagazione monodimensionale, occorre considerare un caso in cui tale condizione risulti verificata nel campo di frequenze di interesse. Un esempio di tale condizione può essere rappresentato dalla propagazione delle onde all'interno di un condotto. La conformazione geometrica dell'ambiente sonoro rende possibile trascurare, per frequenze opportunamente basse, l'effetto di propagazione tridimensionale semplificando il modello dell'ambiente sonoro ad una sola dimensione, longitudinale allo sviluppo del condotto. Le onde piane che si propagano all'interno del condotto possono essere, ad esempio, quelle generate da un comune altoparlante (sorgente) comandato a basse frequenze. Con buona approssimazione è possibile considerare gli effetti di eccitazione dei modi di elevato ordine localizzati vicino all'altoparlante (*campo vicino* o *near field*) e con un decadimento di tipo esponenziale con la distanza a partire dalla sorgente [7, 8, 9]. Considerando una frequenza di comando della sorgente inferiore alla frequenza di taglio<sup>4</sup> del primo modo trasversale del condotto è possibile trascurare gli effetti dei modi di elevato ordine e considerare la propagazione monodimensionale [10].

---

<sup>4</sup>Considerando la propagazione in aria a 20°C, le onde acustiche si propagano come onde piane tra una frequenza di 20 Hz e la frequenza di taglio (cut-on frequency), al di sopra della quale le onde acustiche di propagano in più dimensioni.

In questo modo è possibile considerare la lunghezza d'onda del segnale generato dalla sorgente più grande della dimensione della sezione trasversale del condotto, infatti vale la seguente relazione:

$$\lambda = \frac{c}{f} > (4 \div 8)d, \quad (1.54)$$

dove  $c$  rappresenta la velocità del suono,  $f$  la frequenza dell'onda acustica generata dalla sorgente e  $d$  il diametro del condotto (avendo considerato un condotto con sezione circolare). Trascurando quindi gli effetti dell'oscillazione del flusso prodotta dal cono dell'altoparlante nelle prossimità dello stesso, il risultato netto del campo acustico può essere rappresentato da una sorgente piana. Tale sorgente genera effettivamente lo stesso campo acustico di onde piane dell'altoparlante nelle direzioni a valle e a monte dello stesso. Se si pensa all'altoparlante come ad una coppia di pistoni, privi di massa e separati da una distanza infinitesima, costretti ad oscillare da parte a parte grazie all'introduzione di un flusso di volume oscillante  $q(t)$  posto tra i due pistoni<sup>5</sup>, l'altoparlante può essere modellato come una sorgente di onde piane di tipo monopolo [11], come illustrato in figura 1.8.

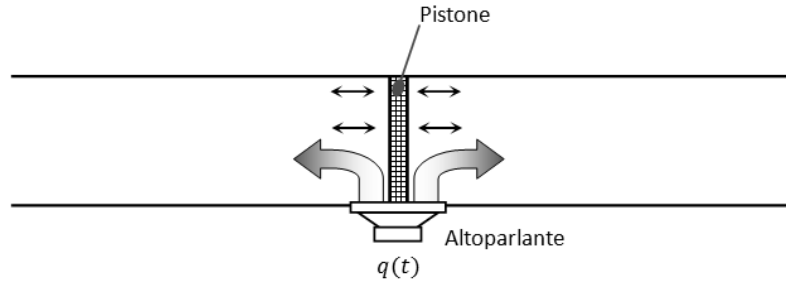


Figura 1.8: Generazione di onde piane in un condotto.

Denominando con  $V(x_0^+)$  il modulo della velocità di particella generata dal pistone nella direzione delle  $x$  positive, e con  $V(x_0^-)$  il modulo della velocità di particella generata dal pistone nella direzione delle  $x$  negative, per l'onda che si propaga lungo la direzione delle  $x$  positive ( $x > x_0$ ) valgono le seguenti relazioni:

$$v_{dx}(x) = V(x_0^+)e^{-jk(x-x_0)}, \quad (1.55)$$

$$p_{dx}(x) = \rho_0 c V(x_0^+)e^{-jk(x-x_0)}, \quad (1.56)$$

<sup>5</sup> $q(t)$  rappresenta il flusso di volume o la velocità di volume, espressa in  $[m^3/s]$ .

dove  $p_{dx}(x)$  è la pressione sonora e  $v_{dx}(x)$  è la velocità di particella nella direzione delle  $x$  positive e  $k$  è il numero d'onda<sup>6</sup>. Per l'onda diretta lungo le  $x$  negative ( $x < x_0$ ) si ha:

$$v_{sx}(x) = V(x_0^-)e^{jk(x-x_0)}, \quad (1.57)$$

$$p_{sx}(x) = -\rho_0 c V(x_0^-)e^{jk(x-x_0)}. \quad (1.58)$$

Nella figura 1.9 viene mostrato l'andamento della pressione acustica e della velocità di particella lungo l'asse  $x$  del condotto. La relazione che lega la velocità di particella alla pressione dinamica è rappresentata dall'equazione di Eulero linearizzata (1.12):

$$v(x) = -\frac{1}{j\omega\rho_0} \frac{dp(x)}{dx} = -\frac{1}{jk\rho_0 c} \frac{dp(x)}{dx}, \quad (1.59)$$

dove  $\omega = 2\pi f$  è la pulsazione dell'onda. Analizzando più in dettaglio la figura 1.9 e le relazioni 1.55, 1.56, 1.57 e 1.58 si possono trarre le seguenti considerazioni:

- Discontinuità della velocità di particella in  $x = x_0$ . L'ampiezza della discontinuità della velocità di particella in  $x = x_0$  è determinata dal flusso di volume "iniettato" tra i due pistoni.
- Andamento continuo della pressione in prossimità della sorgente ( $x = x_0$ ).
- La sorgente di onde piane (pistone) è "trasparente" a qualunque onda incidente, nel senso che i pistoni che costituiscono la sorgente vengono assunti privi di massa e si muovono esattamente secondo le fluttuazioni della velocità di particella associate a qualunque altra onda propagante all'interno del condotto.

---

<sup>6</sup>Si ricorda che il numero d'onda è una grandezza caratteristica delle onde analoga alla frequenza, ma riferita alla periodicità spaziale. Essendo proporzionale all'inverso della lunghezza d'onda si misura in  $m^{-1}$ . Tale parametro viene espresso come:

$$k = \frac{2\pi}{\lambda},$$

dove  $\lambda$  è la lunghezza d'onda. Esprimendo  $\lambda$  in funzione della frequenza dell'onda e della velocità del suono come:

$$\lambda = \frac{c}{2\pi f} = \frac{c}{\omega},$$

dove  $\omega = 2\pi f$  è la pulsazione dell'onda, si ottiene:

$$k = \frac{2\pi\omega}{c}.$$

- Mentre il pistone di sinistra si muove con movimento positivo ( $V(x_0^-) > 0$ ), viene generata a monte un'onda rarefatta.
- Il flusso di volume iniettato tra i due pistoni è assunto indipendente dalla pressione prodotta sui pistoni, ossia viene considerata infinita l'*impedenza acustica* interna della sorgente<sup>7</sup>.

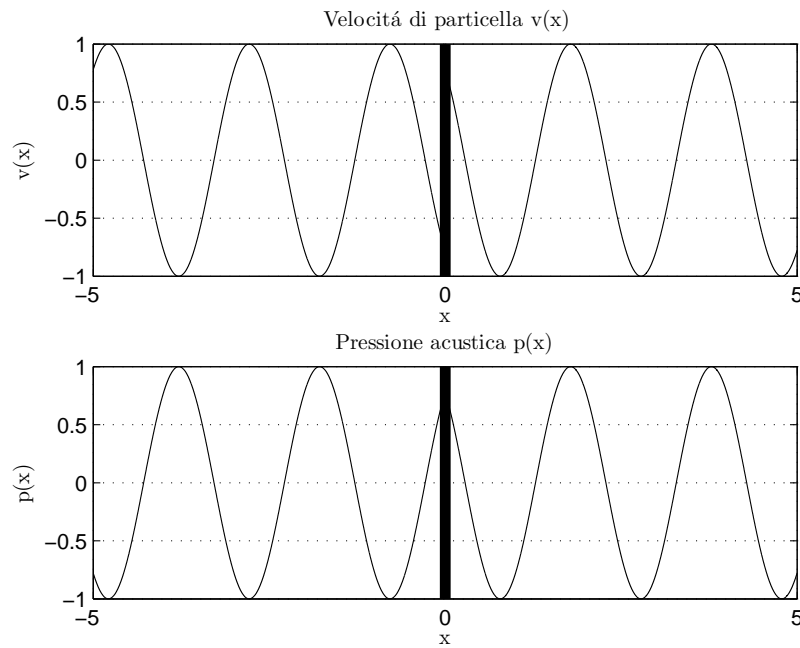


Figura 1.9: Andamento della pressione acustica e della velocità di particella dovuta alla presenza di una sorgente monopolo con generazione di onde piane.

Dalla continuità della pressione acustica in corrispondenza del punto di ingresso del flusso di volume in  $x_0$  si ha:

$$p_{sx}(x_0) = p_{dx}(x_0), \quad (1.61)$$

$$-\rho_0 c V(x_0^-) = \rho_0 c V(x_0^+), \quad (1.62)$$

<sup>7</sup>La pressione dinamica è legata alla velocità di particella attraverso l'impedenza acustica  $R$  del mezzo di propagazione, secondo la nota relazione:

$$p = \rho_0 c v = Rv. \quad (1.60)$$

pertanto, il flusso di volume iniettato dal pistone di destra e di sinistra è equivalente e pari a:

$$-V(x_0^-) = V(x_0^+). \quad (1.63)$$

Data la velocità di volume  $q(x_0)$  generata dalla sorgente è possibile assumere per basse frequenze che questa viene suddivisa equamente nelle due direzioni, ottenendo quindi:

$$-V(x_0^-) = V(x_0^+) = \frac{q(x_0)}{2S}, \quad (1.64)$$

dove  $S$  è la sezione in  $m^2$  del condotto. Il campo di pressione totale può essere espresso in termini della velocità di volume  $q(x_0)$  come:

$$p(x) = \frac{q(x_0)}{2S} \rho_0 c e^{-jk|x-x_0|}. \quad (1.65)$$

Ogni fluttuazione di volume che spinge i due pistoni, viene riprodotta come fluttuazione di pressione esattamente dopo un tempo pari a  $\frac{|x-x_0|}{c}$ . La semplice relazione 1.65 tra le fluttuazioni di pressione e la velocità di volume non è valida in caso di campi acustici tridimensionali, infatti la fluttuazione di pressione è legata alla derivata nel tempo della fluttuazione della velocità di volume, secondo la seguente relazione:

$$\frac{\partial p(x, t)}{\partial x} = -\rho \frac{\partial v(x, t)}{\partial t}. \quad (1.66)$$

### 1.3.1 Meccanismo di cancellazione delle onde piane

Nel presente paragrafo viene analizzata l'interazione tra due onde piane che si propagano all'interno di un condotto (figura 1.10). Le caratteristiche delle due sorgenti sono le medesime di quelle mostrate in precedenza, quindi la generazione delle onde piane viene realizzata grazie a due sorgenti modellizzate come due pistoni in grado di introdurre un flusso di volume simmetrico a destra e a sinistra all'interno del condotto. I pedici  $p$  e  $c$  mostrati in figura 1.10 indicano rispettivamente la sorgente principale e la sorgente di controllo. Alle relazioni viste in precedenza si affianca ora il principio di sovrapposizione degli effetti.

È possibile esprimere la relazione 1.65 in riferimento alle due sorgenti presenti, ottenendo:

$$p_p(x) = q_p \frac{\rho_0 c}{2S} e^{-jk|x|}, \quad (1.67)$$

$$p_c(x) = q_c \frac{\rho_0 c}{2S} e^{-jk|x-L|}. \quad (1.68)$$

Il principio di sovrapposizione degli effetti consente di affermare che la pressione in un generico punto del condotto è equivalente a:

$$p(x) = p_p(x) + p_c(x). \quad (1.69)$$

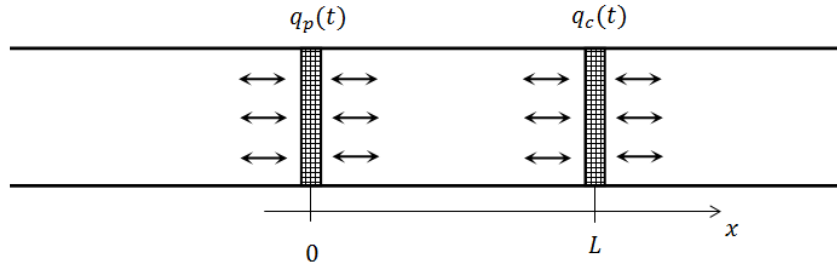


Figura 1.10: Cancellazione di due onde piane in un condotto.

Pensando di voler ottenere una completa cancellazione dell'onda a partire dalla sorgente di controllo, quindi per  $x \geq L$ , si ha:

$$q_p \frac{\rho_0 c}{2S} e^{-jk|x|} + q_c \frac{\rho_0 c}{2S} e^{-jk|x-L|} = 0, \quad x \geq L, \quad (1.70)$$

$$q_p e^{-jk|x|} = -q_c e^{-jk|x-L|}, \quad (1.71)$$

$$q_c = -q_p e^{-jkL}. \quad (1.72)$$

Esplicitando il numero d'onda  $k$  si ha:

$$q_c = -q_p e^{-j\omega \frac{L}{c}}, \quad (1.73)$$

pertanto la velocità di volume della sorgente di controllo  $q_c(t)$  equivale ad un puro ritardo dell'onda generata dalla sorgente principale, infatti:

$$q_c(t) = -q_p \left( t - \frac{L}{c} \right). \quad (1.74)$$

Il processo di cancellazione delle onde piane, appena descritto, può essere anche osservato graficamente nella figura 1.11, dove in rosso è indicata l'onda generata dalla sorgente principale, in azzurro l'onda generata dalla sorgente di controllo e in giallo il risultato netto ottenuto mediante il principio di sovrapposizione, dato dalla somma delle onde generate dalle due sorgenti.

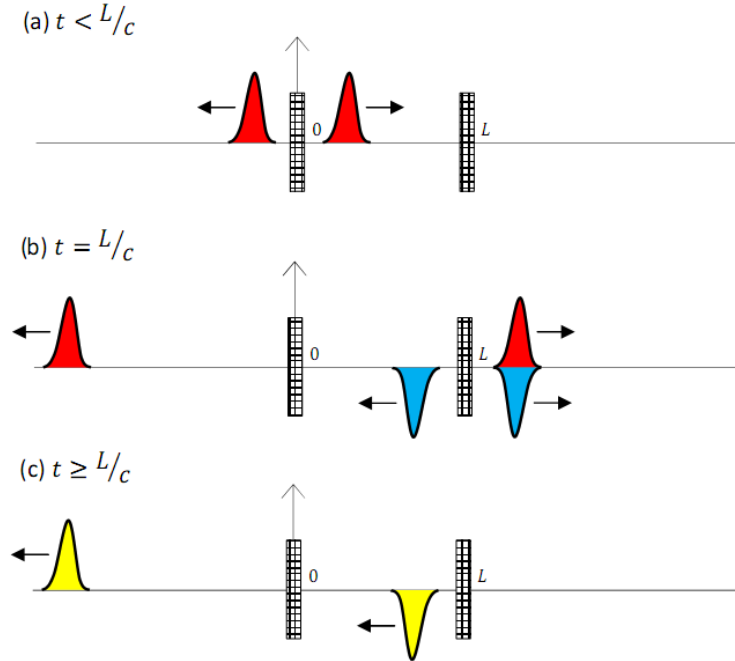


Figura 1.11: *Processo di cancellazione perfetta dell'onda a valle della sorgente di controllo.*

Nella figura 1.11(a) viene mostrata la propagazione delle onde negli istanti precedenti  $t = L/c$ , dove la sorgente principale genera un'onda che viaggia secondo le modalità descritte nel paragrafo 1.3. Non appena l'onda raggiunge la sorgente di controllo, ciò avviene all'istante  $t = L/c$ , la seconda sorgente emette un'onda, nelle medesime modalità della sorgente principale. L'ampiezza dell'onda generata dalla sorgente di controllo, figura 1.11(b) risulta di pari modulo ma segno opposto a quello dell'onda della sorgente principale, permettendo un'esatta cancellazione dell'onda generata dalla prima sorgente nelle posizioni  $x \geq L$ . Dalla figura 1.11(c) è evidente come il risultato di questo processo di cancellazione è quello di una riflessione delle onde da parte della sorgente di controllo: in particolare è importante notare come ad un'onda compressa proveniente dalla sorgente principale, la sorgente di controllo risponda con un'onda rarefatta che si propaga verso la prima sorgente.



Si considera ora il processo di cancellazione dal punto di vista analitico; per semplificare la trattazione si considera un movimento delle onde puramente armonico:

- A valle della sorgente di controllo la pressione è nulla.
- Sfruttando le relazioni 1.69 e 1.72, nella regione di condotto compresa tra le due sorgenti ( $0 \leq x \leq L$ ) si ha:

$$\begin{aligned}
 p(x) &= q_p \frac{\rho_0 c}{2S} e^{-jkx} + q_c \frac{\rho_0 c}{2S} e^{-jk(L-x)} = \\
 &= q_p \frac{\rho_0 c}{2S} e^{-jkx} \underbrace{-q_p e^{-jkL}}_{+q_c} \frac{\rho_0 c}{2S} e^{-jk(L-x)} = \\
 &= q_p \frac{\rho_0 c}{2S} e^{-jkL} \left[ e^{jk(L-x)} - e^{-jk(L-x)} \right].
 \end{aligned} \tag{1.75}$$

Il campo acustico in questa zona è costituito da due onde di uguale ampiezza che viaggiano in direzioni opposte. Le due onde si combinano e danno luogo ad onda stazionaria.

- Nella zona a monte della sorgente principale ( $x \leq 0$ ) si ha:

$$\begin{aligned}
 p(x) &= q_p \frac{\rho_0 c}{2S} e^{jkx} + q_c \frac{\rho_0 c}{2S} e^{-jk(L-x)} = \\
 &= q_p \frac{\rho_0 c}{2S} e^{jkx} \underbrace{-q_p e^{-jkL}}_{+q_c} \frac{\rho_0 c}{2S} e^{-jk(L-x)} = \\
 &= q_p \frac{\rho_0 c}{2S} e^{jkx} \left( 1 - e^{-j2kL} \right) = \\
 &= q_p \frac{\rho_0 c}{2S} e^{jkx} [1 - \cos(2kL) + j \sin(2kL)].
 \end{aligned} \tag{1.76}$$

La relazione 1.76 mette in evidenza che a monte della sorgente principale è presente una sola onda la cui ampiezza è funzione del parametro  $kL$ , quindi dipendente dalla distanza delle due sorgenti e dalla lunghezza d'onda  $\lambda$ . L'ampiezza dell'onda a monte della sorgente principale è nulla se il parametro  $kL$  è un multiplo intero dell'angolo giro ( $kL = n\pi$ ). Ciò equivale a:

$$L = \frac{n\pi}{k} = \frac{n\pi}{\frac{2\pi}{\lambda}} = n \frac{\lambda}{2}, \tag{1.77}$$

quindi per una distanza  $L$  delle sorgenti pari ad un multiplo intero di mezza lunghezza d'onda il campo acustico viene cancellato sia a valle della sorgente di controllo che a monte della sorgente principale.

Nella figura 1.12 si riporta l'andamento della pressione acustica nel condotto nel caso in cui le due sorgenti monopolo vengono poste ad una generica distanza  $L$ . I parametri della simulazione sono i seguenti:

- Frequenza dell'onda di rumore:  $f = 100 \text{ Hz}$ .
- Distanza tra le due sorgenti:  $L = 3.3 \text{ m}$ .

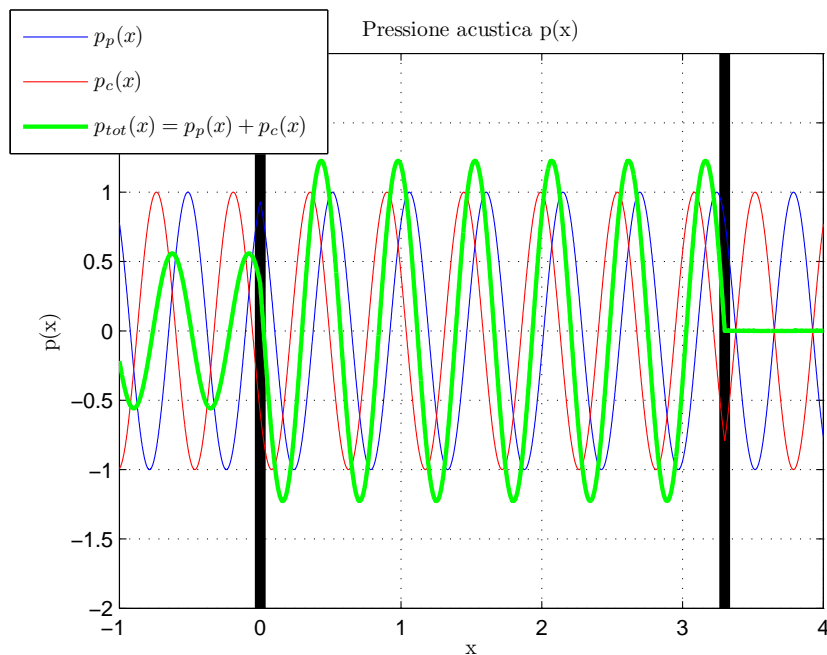


Figura 1.12: Andamento delle pressioni acustiche all'interno del condotto nel caso in cui la distanza delle sorgenti non rispetta la relazione 1.77.

Nella figura 1.13 si riporta l'andamento della pressione acustica nel condotto nel caso in cui le due sorgenti monopolo vengono poste ad una distanza  $L$  che rispetta la relazione 1.77. I parametri della simulazione sono i seguenti:

- Frequenza dell'onda di rumore:  $f = 100 \text{ Hz}$ .
- Distanza tra le due sorgenti:  $L = 4\frac{\lambda}{2} = 1.09 \text{ m}$ .

Per quanto riguarda il processo di cancellazione, risulta evidente come il campo acustico netto ad una data frequenza, prodotto dall'interferenza del campo acustico della sorgente principale e della sorgente di controllo abbia una forte influenza della distanza di separazione delle due sorgenti. La distanza di separazione delle sorgenti comporta

necessariamente delle variazioni delle potenze acustiche in uscita dalle due sorgenti. Al fine di analizzare in dettaglio questo effetto, si determina di seguito la media temporale della potenza in uscita dalle due sorgenti nel caso di perfetta cancellazione del campo acustico a valle della sorgente secondaria.

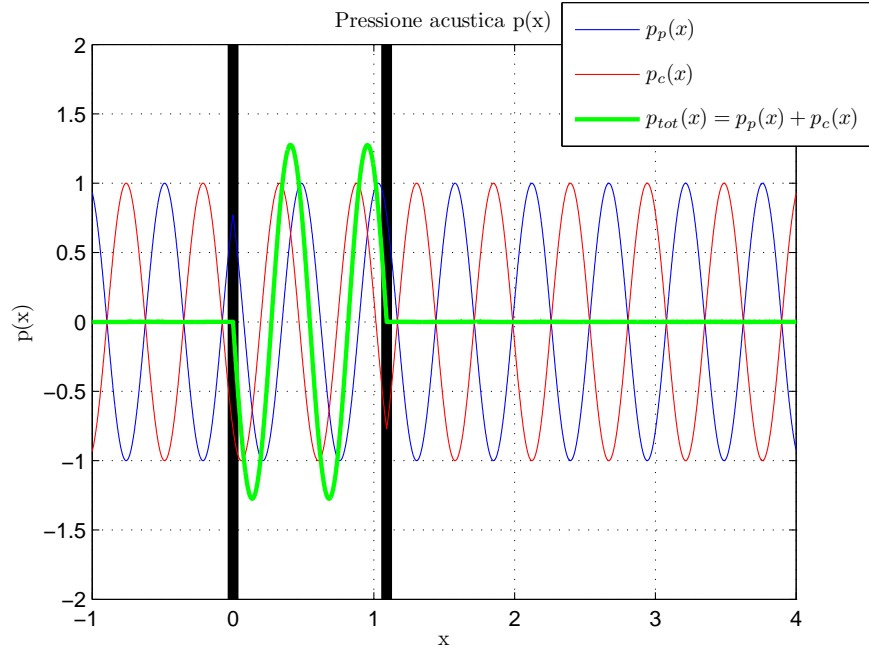


Figura 1.13: *Andamento delle pressioni acustiche all'interno del condotto nel caso in cui la distanza delle sorgenti rispetta la relazione 1.77.*

La potenza sonora può essere determinata come integrale di superficie dell'intensità acustica emessa dalla sorgente. Dato che la superficie è costante e pari alla sezione  $S$  del condotto, si ha:

$$W = \int_S I dS = IS. \quad (1.78)$$

L'intensità acustica è data a sua volta dal prodotto della pressione acustica per la velocità di particella, si può quindi riscrivere ed elaborare la relazione 1.78 nel seguente modo:

$$W = IS = pvS = p \frac{q}{2S} S = \frac{1}{2} pq. \quad (1.79)$$

Il flusso netto di energia in uscita da una sorgente che introduce un flusso di volume  $q(x_0)$  può essere espresso come:

$$W = \frac{1}{2} \Re [p^*(x_0)q(x_0)], \quad (1.80)$$

dove  $p^*(x_0)$  è il complesso coniugato della pressione totale in  $x_0$ , dovuta alla presenza delle diverse sorgenti. Si analizza quindi in dettaglio la potenza acustica nei vari tratti di condotto nel caso di perfetta cancellazione del rumore a valle della sorgente secondaria:

- In corrispondenza della sorgente secondaria ( $x = L$ ) la pressione acustica è nulla, pertanto è nullo il flusso netto di energia:

$$p(L) = 0 \quad \Rightarrow \quad W(L) = W_c = 0. \quad (1.81)$$

La sorgente secondaria in questo caso non immette né assorbe potenza acustica netta, tuttavia per immettere il flusso di volume  $q_c(t)$  essa necessita di elevata potenza elettrica.

- In corrispondenza della sorgente principale ( $x = 0$ ), la pressione acustica è data da:

$$p(0) = q_p \frac{\rho_0 c}{2S} (1 - e^{-j2kL}). \quad (1.82)$$

La potenza acustica vale quindi:

$$W_p^{(c)} = W(0) = \frac{1}{2} \Re \left[ q_p^* \frac{\rho_0 c}{2S} (1 - e^{-j2kL}) q_p \right] = \quad (1.83)$$

$$= \frac{1}{2} \frac{\rho_0 c}{2S} |q_p|^2 \Re [1 - e^{-j2kL}]; \quad (1.84)$$

rielaborando il termine  $\Re [1 - e^{-j2kL}]$  nel seguente modo:

$$\Re [1 - e^{-j2kL}] = 1 - \cos(2kL) = 2 \sin^2(kL), \quad (1.85)$$

si ottiene l'espressione per la potenza netta acustica in  $x = 0$ :

$$W_p^{(c)} = W(0) = \frac{\rho_0 c}{2S} |q_p|^2 \sin^2(kL). \quad (1.86)$$

Nel caso in cui non sia presente la sorgente secondaria (sorgente di controllo), la potenza netta  $W_p^{(nc)}$  in corrispondenza della sorgente principale vale:

$$p(0) = \frac{\rho_0 c}{2S} q_p \quad \Rightarrow \quad W_p^{(nc)} = W(0) = \frac{1}{2} \Re \left[ \frac{\rho_0 c}{2S} q_p^* q_p \right] = \frac{1}{2} \frac{\rho_0 c}{2S} |q_p|^2. \quad (1.87)$$

Nota la potenza netta uscente in  $x = 0$  con e in assenza della sorgente secondaria è possibile determinare il rapporto tra queste due potenze:

$$\frac{W_p^{(c)}}{W_p^{(nc)}} = 2 \sin^2(kL). \quad (1.88)$$

Da quest'ultimo rapporto, non unitario, risulta evidente l'influenza della presenza della sorgente secondaria sulla potenza netta uscente in  $x = 0$ .

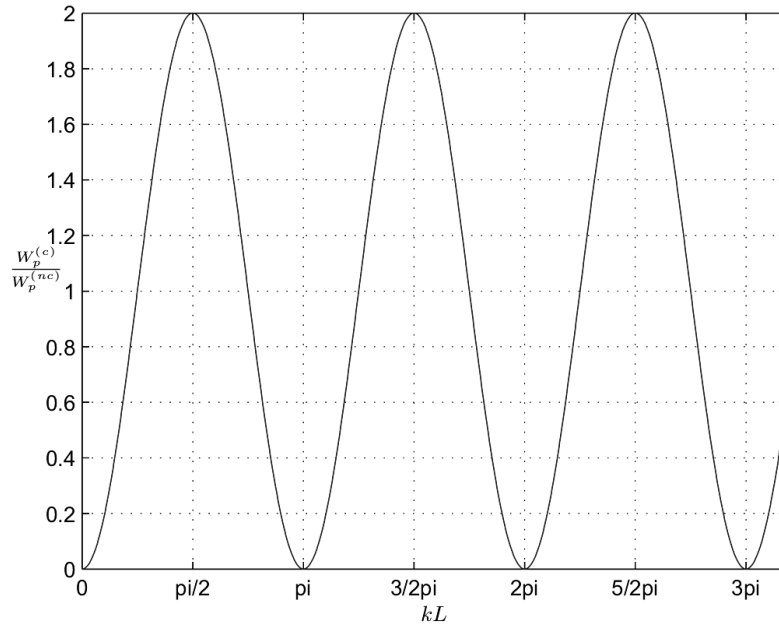


Figura 1.14: *Rappresentazione grafica della relazione 1.88: andamento del rapporto delle potenze in  $x = 0$  con e in assenza della sorgente di controllo.*

La sorgente secondaria è in grado di modificare le fluttuazioni di pressione in  $x = 0$  pertanto è in grado di modificare la potenza della sorgente principale, secondo quanto illustrato nella figura 1.14 che riporta graficamente la funzione 1.88:

- La potenza acustica emessa dalla sorgente principale è raddoppiata alle frequenze dell'onda di rumore corrispondenti a distanze di separazione delle due sorgenti pari a:

$$kL = (2n + 1) \frac{\pi}{2} \quad \Rightarrow \quad L = \frac{1}{4}\lambda, \frac{3}{4}\lambda, \frac{5}{4}\lambda, \text{ ecc.} \quad (1.89)$$

- La potenza in uscita dalla sorgente principale è nulla alle frequenze dell'onda di rumore corrispondenti a distanze di separazione delle due sorgenti pari a:

$$kL = n\pi \quad \Rightarrow \quad L = \frac{1}{2}\lambda, \lambda, \frac{3}{2}\lambda, \text{ ecc.} \quad (1.90)$$

In questa condizione, come visto precedentemente, è nulla la pressione acustica totale a monte della sorgente principale.

- Se le fluttuazioni di pressione in corrispondenza della sorgente principale vengono annullate, quest'ultima non avrà più nulla da respingere, pertanto essa non lavora sul mezzo di propagazione e non emette potenza acustica.

### 1.3.2 Meccanismo di assorbimento delle onde piane

Un secondo meccanismo che si verifica nell'abbattimento delle onde di pressione è quello di assorbimento, infatti è possibile comandare la sorgente secondaria in modo che essa assorba l'energia. Come verrà mostrato in seguito, questo processo è quello che avviene mediante l'utilizzo di sorgenti passive risonanti dove il principale meccanismo di abbattimento è quello di assorbimento dell'energia.

A partire dalle relazioni 1.67 e 1.68 è possibile determinare la pressione e quindi la potenza acustica in corrispondenza della sorgente secondaria:

$$\begin{aligned} p(L) &= p_p(L) + p_c(L) = q_p \frac{\rho_0 c}{2S} e^{-jk|x|} \Big|_{x=L} + q_c \frac{\rho_0 c}{2S} e^{-jk|x-L|} \Big|_{x=L} = \\ &= q_p \frac{\rho_0 c}{2S} e^{-jkL} + q_c \frac{\rho_0 c}{2S}, \end{aligned} \quad (1.91)$$

$$W_c = \frac{1}{2} \frac{\rho_0 c}{2S} \Re \left[ \left( q_p e^{-jkL} + q_c \right)^* q_c \right]. \quad (1.92)$$

Come mostra la relazione 1.92, la potenza acustica è una funzione quadratica del flusso di volume  $q_c$ . Questa relazione può essere rielaborata in modo da ottenere una relazione in forma più utile per le elaborazioni matematiche che seguiranno: dato che la parte reale di un numero complesso è pari a metà della somma del numero stesso e del suo complesso coniugato, possiamo riscrivere la 1.92 come:

$$W_c = \frac{\rho_0 c}{8S} \left[ \left( q_p e^{-jkL} + q_c \right)^* q_c + \left( q_p e^{-jkL} + q_c \right) q_c^* \right]. \quad (1.93)$$

Sviluppando i prodotti di quest'ultima relazione:

$$W_c = \frac{\rho_0 c}{8S} \left[ 2|q_c|^2 + \underbrace{\left( q_p e^{-jkL} \right)^*}_{\hat{q}_p} q_c + \left( q_p e^{-jkL} \right) q_c^* \right], \quad (1.94)$$

e mettendo in evidenza la parte reale ed immaginaria dei singoli termini si ha:

$$W_c = \frac{\rho_0 c}{8S} \left[ 2 \left( q_{cR}^2 + q_{cI}^2 \right) + 2\hat{q}_{pR} q_{cR} + 2\hat{q}_{pI} q_{cI} \right]. \quad (1.95)$$

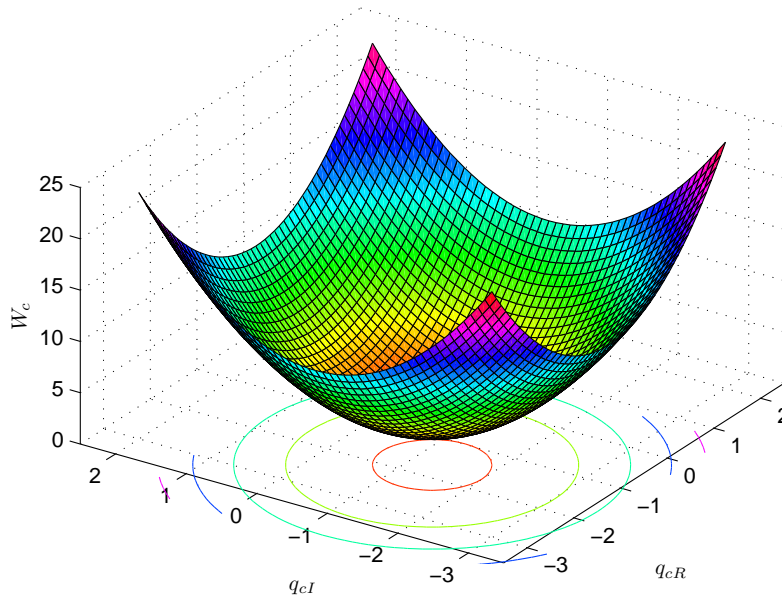


Figura 1.15: *Rappresentazione grafica della relazione 1.92: andamento della potenza acustica al variare del flusso di volume  $q_c$  introdotto dalla sorgente di controllo.*

Nella figura 1.15 viene riportato l'andamento della potenza acustica in funzione della parte reale ed immaginaria del flusso di volume della sorgente di controllo. Nella determinazione della funzione rappresentata in figura si è trascurato il termine  $\frac{\rho_0 c}{8S}$  e si è imposto unitario il flusso di volume generato dalla sorgente principale. Dalla figura è evidente che tale funzione ammette un solo punto di minimo. Il punto di minimo della

funzione 1.95 può essere determinato differenziando tale funzione rispetto alla parte reale ed immaginaria del flusso di volume  $q_c$ , e annullando tali derivate:

$$\frac{\partial W_c}{\partial q_{cR}} = 0 \Rightarrow \frac{\partial [2(q_{cR}^2 + q_{cI}^2) + 2\hat{q}_{pR}q_{cR} + 2\hat{q}_{pI}q_{cI}]}{\partial q_{cR}} = 0 \Rightarrow q_{cR0} = -\frac{1}{2}\hat{q}_{pR}, \quad (1.96)$$

$$\frac{\partial W_c}{\partial q_{cI}} = 0 \Rightarrow \frac{\partial [2(q_{cR}^2 + q_{cI}^2) + 2\hat{q}_{pR}q_{cR} + 2\hat{q}_{pI}q_{cI}]}{\partial q_{cI}} = 0 \Rightarrow q_{cI0} = -\frac{1}{2}\hat{q}_{pI}. \quad (1.97)$$

Da queste relazioni, il minimo della funzione 1.95 può quindi essere scritto come:

$$q_{c0} = q_{cR0} + iq_{cI0} = -\frac{1}{2}\hat{q}_{pR} - i\frac{1}{2}\hat{q}_{pI} = -\frac{1}{2}q_p. \quad (1.98)$$

La potenza acustica minima vale pertanto:

$$W_{c0} = -\frac{\rho_0 c}{16S}|q_p|^2. \quad (1.99)$$

Dalla relazione 1.99 è possibile trarre le seguenti conclusioni:

- Essendo  $W_{c0} < 0$ , in queste condizioni vi è assorbimento della potenza acustica.
- La potenza acustica  $W_{c0}$  è pari ad 1/4 di quella in assenza della sorgente di controllo  $W_p^{(nc)}$ :

$$W_{c0} = -\frac{1}{4}W_p^{(nc)}. \quad (1.100)$$

- Nel dominio del tempo il massimo assorbimento si ha in corrispondenza di un flusso di volume della sorgente di controllo pari a  $q_{c0}(t) = -\frac{1}{2}q_p(t - L/c)$ , ossia metà ampiezza della sorgente principale, invertita e ritardata, secondo quanto mostrato nella figura 1.16. In particolare in rosso è indicata l'onda generata dalla sorgente principale, in azzurro l'onda generata dalla sorgente di controllo e in giallo il risultato netto ottenuto mediante il principio di sovrapposizione, dato dalla somma delle onde generate dalle due sorgenti.

Il risultato finale, mostrato in figura 1.16(c), mostra che l'onda iniziale generata dalla sorgente principale in  $x = 0$ , viene per metà trasmessa e per metà riflessa in corrispondenza della sorgente secondaria. Siccome la potenza è proporzionale al quadrato dell'ampiezza della pressione, considerando unitaria la potenza dell'onda iniziale, la potenza delle due mezze onde ammonta a  $(\frac{1}{2})^2 + (\frac{1}{2})^2 = \frac{1}{2}$ , ossia viene assorbita metà



della potenza dell'onda iniziale, pari ad  $(\frac{1}{4})$  della potenza complessiva dell'onda generata dalla sorgente principale (infatti quest'ultima è costituita da un'onda propagante verso la sorgente secondaria e da un'onda diretta nel verso opposto.)

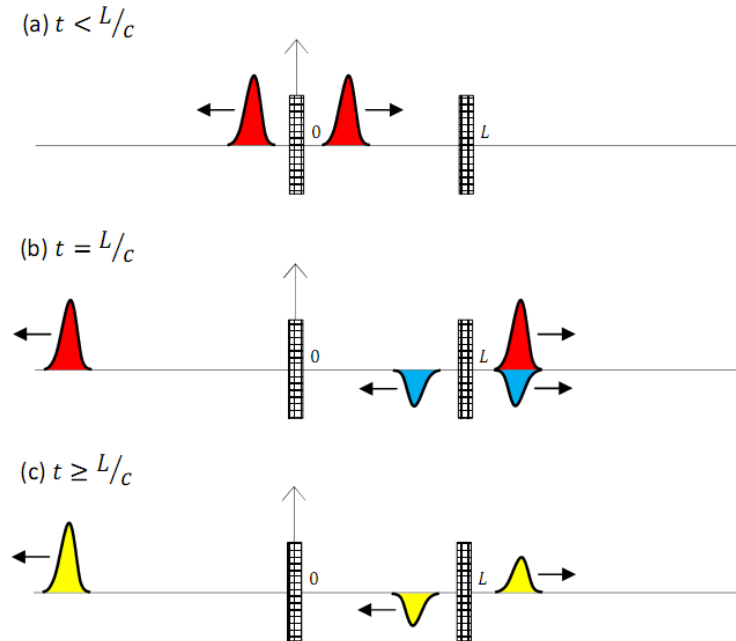


Figura 1.16: *Processo di cancellazione dovuto al massimo assorbimento da parte della sorgente di controllo.*

Concludendo si può affermare che una sorgente di tipo monopolo può assorbire l'onda incidente, ma l'effetto di assorbimento di un'onda piana in un condotto di lunghezza infinita risulta comunque limitato a metà dell'energia acustica incidente. In queste condizioni la sorgente permette di trasmettere e riflettere  $1/4$  dell'energia incidente. Questa strategia è quella che garantisce il massimo assorbimento da parte della sorgente secondaria, ma quanto mostrato mette in luce come questa non sia quella più efficiente per ridurre il rumore a valle della sorgente secondaria. Nel controllo attivo del rumore in un condotto volendo ridurre di fatto il rumore a valle della sorgente secondaria, il meccanismo di assorbimento da parte di quest'ultima sorgente si presenta solitamente come un effetto indesiderato. Nelle condizioni di massimo assorbimento da parte della sorgente secondaria, è possibile determinare la potenza acustica in corrispondenza della sorgente principale ( $x = 0$ ). A partire dalle relazioni 1.67 e 1.68, la pressione acustica in  $x = 0$  vale:

$$\begin{aligned}
p(0) &= p_p(0) + p_c(0) = q_p \frac{\rho_0 c}{2S} e^{-jk|x|} \Big|_{x=0} + q_{c0} \frac{\rho_0 c}{2S} e^{-jk|x-L|} \Big|_{x=0} = \\
&= q_p \frac{\rho_0 c}{2S} + q_{c0} \frac{\rho_0 c}{2S} e^{-jkL}.
\end{aligned} \tag{1.101}$$

La potenza acustica in corrispondenza della sorgente principale vale quindi:

$$\begin{aligned}
W_p &= \frac{1}{2} \frac{\rho_0 c}{2S} \Re \left[ \left( q_p + q_c e^{-jkL} \right)^* q_p \right] = \text{posto } q_c = q_{c0} \\
&= \frac{1}{2} \frac{\rho_0 c}{2S} \Re \left[ \left( q_p - \frac{1}{2} q_p e^{-jkL} e^{-jkL} \right)^* q_p \right] = \\
&= \frac{1}{2} \frac{\rho_0 c}{2S} \Re \left[ \left( 1 - \frac{1}{2} e^{-j2kL} \right)^* \right] |q_p|^2 = \\
&= \frac{1}{2} \frac{\rho_0 c}{2S} |q_p|^2 \left[ 1 - \frac{1}{2} \cos(2kL) \right] = \\
&= W_p^{(nc)} \left[ 1 - \frac{1}{2} \cos(2kL) \right].
\end{aligned} \tag{1.102}$$

L'effetto del massimo assorbimento della sorgente di controllo, modifica la potenza in uscita dalla sorgente principale in un range compreso tra  $0.5W_p^{(nc)}$  e  $1.5W_p^{(nc)}$ , in funzione della distanza di separazione delle due sorgenti relativamente alla frequenza dell'onda che si propaga nel condotto. In queste condizioni, mediando sulla frequenza, la potenza emessa dalla sorgente principale può considerarsi invariata rispetto al caso di assenza della sorgente di controllo.

## 1.4 Soluzioni disponibili per l'abbattimento delle onde di pressione

Il riconoscimento normativo dei problemi legati all'inquinamento acustico e le conseguenti sanzioni previste per infrazioni a norme accettate a livello mondiale hanno dato impeto ad un intero campo di attività commerciali e di ricerca il cui obiettivo è quello di ricercare e sviluppare nuove tecniche e tecnologie per il controllo, e quindi l'attenuazione, del rumore. In questo ambito sono state sviluppate diverse metodologie di controllo del rumore, che si possono differenziare in base al flusso di energia richiesta:

- **Metodi passivi:** sono metodologie di controllo del rumore che non necessitano l'introduzione di energia per poter abbattere il rumore. Si possono suddividere in due famiglie: sistemi fonoassorbenti / fonoisolanti e sistemi risonanti.

- **Metodi attivi:** sono metodologie di controllo che necessitano l'introduzione di energia dall'esterno per poter funzionare.

Con lo scopo di delineare lo stato dell'arte dei metodi di controllo del rumore, nel seguito si elencano alcune soluzioni dei due metodi descritti, evidenziandone pregi e difetti.

### 1.4.1 Metodi passivi

Come accennato nel paragrafo 1.4, all'interno di questa famiglia di metodi è possibile effettuare una ulteriore suddivisione. I sistemi fonoassorbenti / fonoisolanti sono essenzialmente costituiti da opportune schermature poste in punti di interesse dell'ambiente sonoro: in questo caso le proprietà dei materiali utilizzati giocano un ruolo fondamentale nella qualità dell'isolamento acustico. In commercio esistono principalmente due tipologie di materiali adottati in campo acustico per la riduzione del rumore, questi vengono scelti in funzione del loro utilizzo:

- **Materiali fonoassorbenti.** Sono dei materiali che riflettono solo una piccola parte dell'energia incidente, la restante parte viene dissipata ed eventualmente trasmessa. In generale, la capacità di questi materiali di assorbire l'energia incidente aumenta all'aumentare della frequenza e dello spessore dello strato di materiale utilizzato. La caratteristica di questi materiali è quella di non far riflettere il suono all'interno di un ambiente; vengono utilizzati ad esempio nelle casse acustiche. I materiali fonoassorbenti sono utilizzati nei trattamenti acustici degli ambienti chiusi, per ridurre il tempo di riverberazione ed eliminare riflessioni indesiderate.
- **Materiali fonoisolanti.** Sono dei materiali che trasmettono solo una piccola parte dell'energia incidente, la restante parte viene riflessa e assorbita. Questi materiali vengono utilizzati per isolare il rumore all'interno di un ambiente, ossia vengono utilizzati quando si vuole diminuire il rumore trasmesso all'esterno.

Un ulteriore metodo di abbattimento passivo è mediante l'utilizzo di sistemi risonanti, detti anche *filtri acustici*. I filtri acustici vengono progettati per attenuare le pulsazioni sia alle alte che alle basse frequenze, ma solitamente con un unico filtro acustico non è possibile coprire tutto il range di frequenze richiesto. Ogni filtro acustico ha delle caratteristiche specifiche, nel seguito se ne riportano alcuni tipi.

## Risonatore di Helmholtz

I risonatori di Helmholtz sono formati da una cavità messa in comunicazione con l'ambiente per mezzo di un piccolo foro, detto *collo del risonatore*, di dimensioni generalmente trascurabili rispetto a quelle della cavità. Nella figura 1.17 sono raffigurati dei risonatori acustici di Helmholtz di differenti dimensioni.



Figura 1.17: *Risonatori di Helmholtz.*

Quando un'onda sonora va ad incidere sull'ingresso del risonatore, l'aria contenuta nel collo viene posta in oscillazione comportandosi come un pistone oscillante, mentre, l'aria contenuta all'interno della cavità, viene ad essere alternativamente compressa ed espansa comportandosi come una molla. Questo sistema (figg. 1.18 e 1.19), costituito da una massa oscillante data dall'aria nel collo del risonatore, un elemento elastico (aria nella cavità) ed un elemento smorzante dovuto all'attrito sulle pareti del collo, avrà una sua frequenza di risonanza in corrispondenza della quale sarà massima la dissipazione di energia sonora.

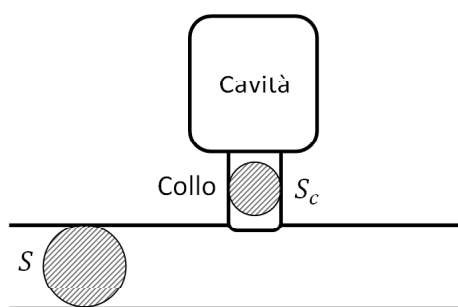


Figura 1.18: *Risonatore di helmholtz semplice.*

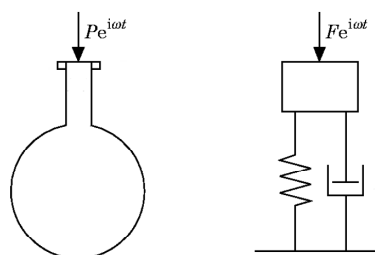


Figura 1.19: *Risonatore di helmholtz semplice. Principio di funzionamento.*

L'assorbimento dei risonatori è elevato in corrispondenza della frequenza di risonanza ma molto ridotto per tutte le altre frequenze. È possibile così costruire dei dispositivi calibrati per assorbire specifiche frequenze. Nella tecnica si usa anche riempire in parte

o tutta la cavità con del materiale assorbente poroso, oppure viene aggiunto questo materiale solo in corrispondenza del collo; queste modifiche allargano lo spettro di frequenze di funzionamento diminuendo però l'effetto in corrispondenza della frequenza di risonanza. Attualmente vengono utilizzati dei pannelli forati rigidi, fissati ad una certa distanza dalla parete; il comportamento acustico di questi pannelli è analogo a quello di un risonatore classico, anche se sono meno efficienti. In questo caso il riempimento dell'intercapedine con materiali porosi migliora effettivamente il comportamento in frequenza della struttura. Il campo ottimale di assorbimento è centrato sulle medie frequenze. Per la costruzione di questi pannelli vengono utilizzati differenti materiali, quali ad esempio, materiali metallici, gesso, legno o materiali plastici [21].

### Accumulatore

Se l'ambiente in cui si propagano le onde di pressione si può considerare monodimensionale, è possibile utilizzare come metodi di abbattimento gli accumulatori. Considerando un ambiente di propagazione come un condotto, gli accumulatori sono costituiti da volumi connessi in serie al condotto (fig. 1.20). Il volume si comporta come una capacità acustica che può ridurre le pulsazioni presenti all'uscita del condotto. Anche in questo caso può essere fatta un'analogia con il caso meccanico, ossia il sistema in esame può essere visto come una molla torsionale a bassa rigidezza in grado di isolare degli elementi del sistema da un'eccitazione torsionale. Essendo la capacità di questi accumulatori molto elevata, la presenza di tale volume determina un abbassamento delle frequenze acustiche naturali del sistema.

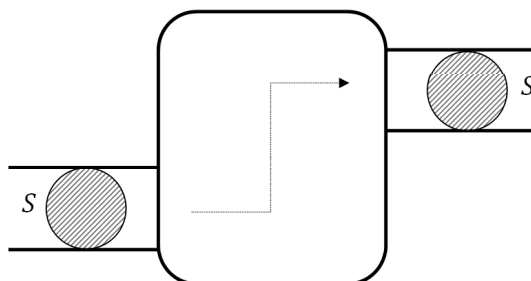


Figura 1.20: Accumulatore per regolarizzare il flusso.

### Filtro acustico passa-basso

Il principio alla base di questi sistemi si basa sul fatto che quando un'onda di pressione incontra un cambiamento di impedenza, associato ad una variazione di volume, parte dell'onda viene riflessa e parte viene trasmessa; l'obiettivo di questi filtri è quello di riflettere il più possibile l'energia incidente e trasmetterne il meno possibile.

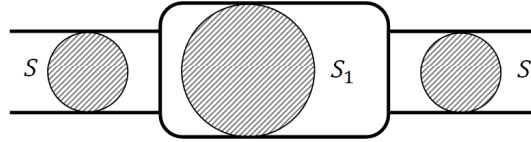


Figura 1.21: *Filtro acustico passa-basso.*

I filtri passa-basso sono molto efficienti a prevenire la trasmissione delle onde acustiche ad elevata frequenza e inefficienti alle basse frequenze, poiché la dimensione di questi sistemi è inversamente proporzionale alla frequenza che si intende controllare. Esistono differenti filtri passa-basso, ad esempio la figura 1.21 mostra la configurazione più semplice, realizzabile inserendo un allargamento in un condotto della macchina. L'allargamento può essere realizzato per mezzo di un condotto aggiuntivo con sezione  $S_1$  superiore alla sezione  $S$  del condotto della macchina in cui si intende inserire il filtro.

### Filtro acustico side-branch

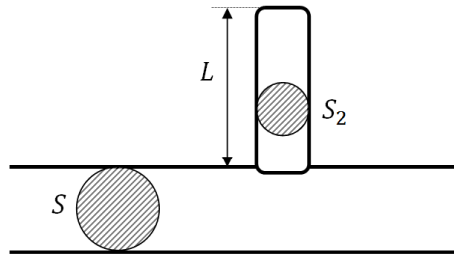


Figura 1.22: *Filtro acustico passa-alto.*

Contrariamente ai filtri passa-basso visti in precedenza, i filtri side-branch possono essere considerati filtri acustici passa-alto. Essi sono efficienti nel prevenire la trasmissione delle onde acustiche a bassa frequenza e risultano poco efficienti alle alte frequenze. Uno dei più semplici filtri passa-alto realizzabili è quello mostrato in figura 1.22, realizzato per mezzo di un condotto di sezione  $S_2$  e lunghezza  $L$ . Altre tipologie di

filtri passa-alto si ottengono inserendo semplici orifizi nei condotti, dove all'aumentare del diametro degli orifizi si ottiene un'attenuazione a più basse frequenze.

### Filtro $\pi$

Da una particolare combinazione di semplici filtri acustici è possibile ottenere i cosiddetti filtri “*volume-choke-volume*”, conosciuti anche come filtri di tipo  $\pi$ . Questi filtri (fig. 1.23) consistono in due volumi connessi tra loro da un condotto di piccolo diametro. I filtri  $\pi$  sono essenzialmente dei filtri passa-basso. Il principio di funzionamento di questi filtri è molto semplice e anche in questo caso può essere fatta l'analogia con il caso meccanico, dove si ha una molla a bassa rigidezza (volume 1) in serie con una elevata massa (choke) e un'altra molla a bassa rigidezza (volume 2). Se la sorgente di eccitazione è posizionata da un lato del sistema è semplice intuire che l'attenuazione avviene alle frequenze superiori a quelle naturali del filtro a partire dal lato opposto del sistema. L'attenuazione aumenta con la frequenza a causa dell'inerzia della massa (choke).

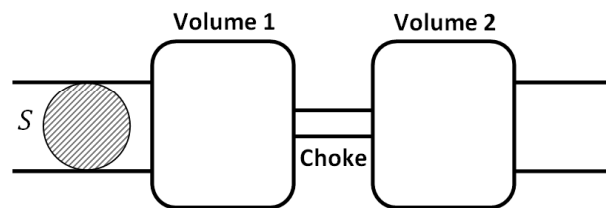


Figura 1.23: Filtro “*volume-choke-volume*”.

### 1.4.2 Metodi attivi

Le caratteristiche degli ambienti sonori e talvolta degli stessi materiali fonoassorbenti fanno sì che questi ultimi non possano essere impiegati in tutte le situazioni in cui vi è la necessità di attenuare il rumore presente in tali ambienti. Da questa esigenza si sono studiati e sviluppati i metodi per il controllo attivo del rumore nei quali, attraverso l'utilizzo di attuatori acustici, si controlla il rumore emesso da una sorgente creando così una “zona di silenzio”. L'interesse attualmente rivolto alle tecniche di riduzione attiva del rumore è motivato dal fatto che, mentre gli interventi “passivi” precedentemente discussi presentano le migliori prestazioni nei campi di frequenza medio alti e comportano spessori e masse sempre crescenti con il diminuire della frequenza da controllare, i dispositivi di riduzione attiva del rumore possono, quando applicabili, fornire risultati

soddisfacenti senza tali inconvenienti proprio nel campo delle basse frequenze, inferiori cioè a  $300 \div 400 \text{ Hz}$ . Va sottolineato che le due tecniche, quella passiva e quella attiva, non sono incompatibili ed alternative tra loro, non è raro infatti il ricorso ad entrambe per la gestione ottimale di fenomeni sonori a largo spettro.

Un sistema di controllo attivo del rumore è composto da:

- **Sensori:** rilevano la sorgente di rumore da abbattere. Tipicamente vengono utilizzati dei microfoni, ma se si conoscono le cause del rumore è possibile utilizzare anche altri sensori, come ad esempio accelerometri.
- **Controllore:** elabora il segnale proveniente dai sensori e determina, in base a specifici algoritmi, il comando degli attuatori.
- **Attuatori:** immettono energia acustica nel sistema. Tipicamente vengono utilizzati attuatori acustici.

I sistemi di controllo attivo del rumore (ANC) sono dunque dei sistemi molto più complessi dei metodi passivi, perché presuppongono oltre ad una buona conoscenza del sistema da isolare, anche la capacità di eseguire delle misure ed elaborare dati durante il loro funzionamento. Tra gli esempi attualmente realizzati a livello industriale o sperimentale si possono citare le cuffie con controllo attivo del rumore e il controllo attivo del rumore all'interno di un autoveicolo.



Figura 1.24: Cuffie con controllo attivo del rumore.

Nel primo caso si tratta di sistemi che isolano l'ascoltatore dai rumori esterni, permettendo una ottima qualità del suono che viene riprodotto. In figura 1.24 viene mostrato un esempio commerciale di tali sistemi. Essi possono essere usati sia per



migliorare la qualità di riproduzione di un brano musicale, ma anche per migliorare la capacità di comunicazione all'interno di ambienti rumorosi, come ad esempio a bordo di un elicottero.

Nel secondo caso, sviluppato da alcune tra le maggiori case automobilistiche, l'obiettivo è migliorare il comfort acustico all'interno della vettura, in particolare si vuole ridurre il rumore generato dal rotolamento dei pneumatici. Anche in questo caso, come mostra la figura 1.25 è possibile individuare nel sistema le tre componenti principali: dei sensori che consentono di effettuare delle misure sull'ambiente acustico e in questo caso anche sulla sorgente di rumore, un sistema di elaborazione delle informazioni rilevate e degli attuatori che immettono energia nel sistema per cancellare le componenti armoniche fastidiose.

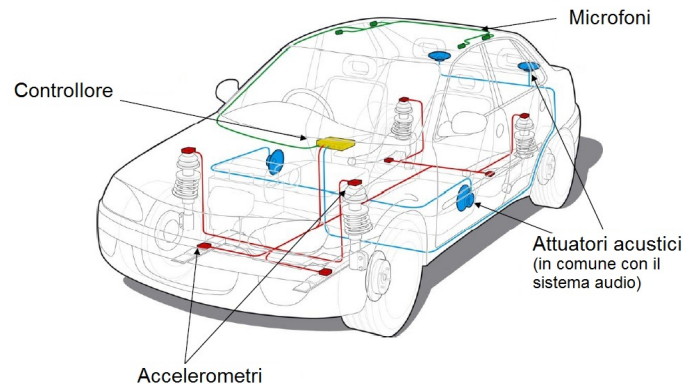


Figura 1.25: Sistema di controllo attivo del rumore all'interno dell'abitacolo di un'automobile.

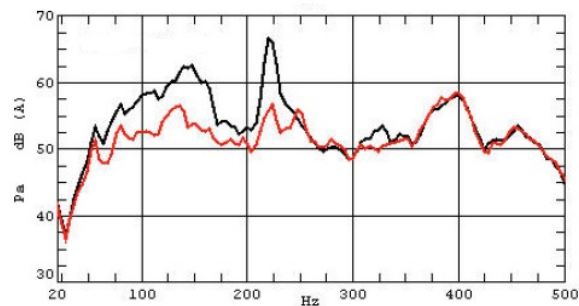


Figura 1.26: Performance di abbattimento del rumore del sistema di controllo attivo.

I risultati di questo sistema possono essere riassunti nel grafico di figura 1.26: con la linea nera viene indicato lo spettro di rumore all'interno dell'abitacolo, con la linea rossa viene indicato lo spettro del rumore che si ottiene con l'inserimento del sistema di controllo attivo. Si nota che questo sistema consente un abbattimento del rumore di circa  $10\text{ dB}$  in un range di frequenze fino a  $300\text{ Hz}$  circa.

Allo stato di evoluzione attuale degli sviluppi applicativi, le tecniche di controllo attivo presentano le seguenti limitazioni:

- Il campo sonoro da gestire deve presentare caratteristiche spaziali non complesse: tipico esempio è la situazione di propagazione del suono in un condotto.
- L'intervento presenta buone efficacia se la lunghezza d'onda del suono è elevata rispetto alle dimensioni del campo sonoro da gestire: pertanto risultano efficaci gli interventi su frequenze di poche centinaia di  $\text{Hz}$ ; solo quando l'obiettivo è la riduzione in una sola ben definita direzione possono essere affrontate anche frequenze relativamente più alte.
- Le difficoltà nella generazione del segnale antagonista sono ovviamente tanto più grandi quanto più estesa è la distribuzione in frequenza del rumore da gestire; i migliori risultati possono essere ottenuti nel caso di toni puri o segnali a banda stretta con eventuali armoniche.
- L'efficacia dell'intervento può essere sostanzialmente migliorata se è possibile rilevare le caratteristiche del rumore da governare prima che questo raggiunga il punto su cui si intende operare; diversamente la regolazione non può che essere basata sui sensori di controllo, il che comporta crescenti imprecisioni nella regolazione con l'aumentare della frequenza del rumore da gestire.
- L'intervento, pur presentando buona efficacia nel punto considerato, può dar luogo ad effetti collaterali di aumento del livello di pressione sonora in altri punti dello spazio, per effetto della cancellazione. Per evitare tali effetti il numero di generatori necessari diventa considerevole, e se il campo è più complesso, con presenza anche di frequenze elevate.

Le prestazioni dei metodi di controllo attivo del rumore dipendono da diversi fattori quali ad esempio la posizione di sensori e attuatori, la qualità e la potenza degli attuatori e le prestazioni richieste agli anelli di controllo. Esse possono quindi essere superiori a quelle fornite dai metodi passivi, ma per poter definire correttamente un impianto

per il controllo attivo del rumore è necessario poter modellare correttamente l'ambiente sonoro e le sue caratteristiche.



## Capitolo 2

# Progettazione di un sistema ANC

### 2.1 Progettazione di un sistema per la generazione di onde piane

La soluzione al problema monodimensionale presentata nel capitolo 1 riassume lo studio fisico delle onde piane progressive, ovvero di un caso speciale di propagazione delle onde in cui tutte le grandezze di interesse acustico variano in funzione, oltre che del tempo, di una sola coordinata cartesiana detta *direzione di propagazione dell'onda*. Di conseguenza sui piani perpendicolari alla direzione di propagazione tutte le grandezze assumono valori costanti, indipendenti dalle altre due coordinate. Una seconda tipologia di onde è costituita dalle onde sferiche, cioè onde che si propagano con simmetria radiale da una sorgente puntiforme in uno spazio omogeneo illimitato. Con queste ipotesi le grandezze acustiche dipendono solo dal tempo e dalla coordinata radiale  $r$  che rappresenta la distanza dal centro di simmetria, dove si trova la sorgente puntiforme. Per le onde sferiche si può utilizzare la scrittura del laplaciano in coordinate sferiche. Tale scrittura deriva direttamente dalla sostituzione delle coordinate cartesiane  $x, y, z$  con le coordinate sferiche  $r, \phi, \theta$ ; trattando in questa sede un problema a simmetria sferica, in esso non si riscontra la dipendenza della soluzione dalle coordinate  $\phi$  e  $\theta$ , pertanto si ottiene nuovamente un problema monodimensionale:

$$p = p(\vec{r}, t), \quad (2.1)$$

$$\vec{r} = r(x, y, z), \quad (2.2)$$

$$\frac{\partial p(\vec{r}, t)}{\partial x} = \frac{\partial p}{\partial r} \frac{\partial r}{\partial x}, \quad (2.3)$$

$$\frac{\partial^2 p(\vec{r}, t)}{\partial x^2} = \frac{\partial^2 p}{\partial r^2} \left( \frac{\partial r}{\partial x} \right)^2 + \frac{\partial p}{\partial x} \frac{\partial^2 r}{\partial x^2}. \quad (2.4)$$

Poiché  $r = \sqrt{x^2 + y^2 + z^2}$  le derivate parziali diventano:

$$\frac{\partial r}{\partial x} = \frac{x}{r}, \quad (2.5)$$

$$\frac{\partial^2 r}{\partial x^2} = \frac{1}{r} \left( 1 - \frac{x^2}{r^2} \right). \quad (2.6)$$

In definitiva si ottiene:

$$\frac{\partial^2 p(\vec{r}, t)}{\partial x^2} = \frac{x^2}{r^2} \frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \left( 1 - \frac{x^2}{r^2} \right) \frac{\partial p}{\partial r}, \quad (2.7)$$

$$\frac{\partial^2 p(\vec{r}, t)}{\partial y^2} = \frac{y^2}{r^2} \frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \left( 1 - \frac{y^2}{r^2} \right) \frac{\partial p}{\partial r}, \quad (2.8)$$

$$\frac{\partial^2 p(\vec{r}, t)}{\partial z^2} = \frac{z^2}{r^2} \frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \left( 1 - \frac{z^2}{r^2} \right) \frac{\partial p}{\partial r}. \quad (2.9)$$

Sommando membro a membro le equazioni 2.7, 2.8 e 2.9 si ottiene il laplaciano in coordinate sferiche:

$$\nabla^2 p = \frac{\partial^2 p}{\partial r^2} + \frac{2}{r} \frac{\partial p}{\partial r} = \frac{1}{r} \frac{\partial^2 (rp)}{\partial r^2}. \quad (2.10)$$

L'equazione delle onde assume dunque la seguente formulazione:

$$\frac{\partial^2 (rp)}{\partial r^2} = \frac{1}{c} \frac{\partial^2 (rp)}{\partial t^2}. \quad (2.11)$$

La soluzione dell'equazione 2.11 deriva dalla soluzione di d'Alembert (mostrata nell'equazione 1.27), ed è la seguente:

$$p(r, t) = \frac{1}{r} F\left(t - \frac{r}{c}\right) + \frac{1}{r} G\left(t + \frac{r}{c}\right), \quad (2.12)$$

che rappresenta la soluzione propagativa, ovvero la sovrapposizione di due onde sferiche progressive smorzate. È possibile definire anche la soluzione stazionaria:

$$p(r, t) = \frac{A_1}{r} \cos k(r - ct) + \frac{A_2}{r} \cos k(r + ct), \quad (2.13)$$

dove  $k = \frac{\omega}{c}$  è detto *numero d'onda* e rappresenta una “frequenza spaziale”. Sostituendo la soluzione mostrata dall'equazione 2.13 nell'equazione di Eulero, si può ricavare la velocità della particella [12]:

$$u(r, t) = \frac{A_1}{\rho_0 c r} \cos k(r - ct) \left[ \frac{1}{rk} \tan k(r - ct) \right] - \frac{A_2}{\rho_0 c r} \cos k(r + ct) \left[ \frac{1}{rk} \tan k(r + ct) \right]. \quad (2.14)$$

Avendo definito le due principali tipologie di onde, le onde piane e le onde sferiche, con lo scopo di costruire un sistema per il controllo attivo del rumore, si è scelto di porsi nella condizione di dover controllare onde piane. Tale scelta è giustificata dal fatto che la generazione, e quindi il controllo di onde sferiche è di per sè molto complessa e richiede l'utilizzo di più attuatori acustici montati in configurazione tale da poter generare tali tipi di onde. La generazione di onde piane è invece più semplice e permette di studiare meglio gli effetti del controllo sul sistema. Inoltre si possono riconoscere meglio gli aspetti legati alla riflessione e rifrazione dell'onda, intervenendo opportunamente. Da qui la scelta di dimensionare un condotto che si comporti come una guida d'onda per la propagazione delle onde piane, e di determinare il posizionamento della sorgente di rumore e dell'attuatore di controllo. Volendo ricondursi al caso di onde piane è necessario introdurre una grandezza, l'*impedenza acustica specifica*, che consentirà successivamente di effettuare delle ipotesi semplificative a riguardo. L'impedenza acustica specifica (impedenza acustica per unità di superficie) è data dal rapporto tra la *pressione sonora* e la *velocità di particella*. Considerando onde di pressione generate da una sfera pulsante si può ridurre il problema al caso monodimensionale, infatti la propagazione viene a dipendere solo dalla distanza  $r$ . L'equazione delle onde sarà quindi:

$$\frac{\partial^2 p}{\partial r^2} - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0. \quad (2.15)$$

L'impedenza acustica viene generalmente rappresentata come una grandezza complessa del tipo:

$$Z_a = \frac{p(r, t)}{u(r, t)} = R + iX, \quad (2.16)$$

dove:

- R è la resistenza acustica (parte reale);
- X è la reattanza acustica (parte immaginaria).

La pressione si determina come visto in precedenza dalla risoluzione dell'equazione delle onde, mentre la velocità della particella si ottiene utilizzando l'equazione di Eulero (eq. 1.12). È possibile esprimere l'equazione 1.12 esplicitando la velocità come:

$$u = -\frac{1}{\rho} \int \frac{\partial p}{\partial r} dt, \quad (2.17)$$

dove il termine  $\frac{\partial p}{\partial r}$  è dato da:

$$\frac{\partial p}{\partial r} = -\frac{A_1}{r^2} e^{i(\omega t - kr)} - ki \frac{A_1}{r} e^{i(\omega t - kr)} - \frac{A_2}{r^2} e^{i(\omega t + kr)} + ki \frac{A_2}{r} e^{i(\omega t + kr)}. \quad (2.18)$$

Dalla risoluzione dell'equazione 2.17 si ha:

$$u = -\frac{1}{\rho} \int \frac{\partial p}{\partial r} dt = \left( -\frac{i}{\omega r \rho} + \frac{k}{\omega \rho} \right) \frac{A_1}{r} e^{i(\omega t - kr)} + \left( \frac{i}{\omega r \rho} + \frac{k}{\omega \rho} \right) \frac{A_2}{r} e^{i(\omega t + kr)}. \quad (2.19)$$

Rielaborando l'equazione precedente si ottiene:

$$u = \left( \frac{\frac{1}{r} + ik}{i\omega \rho} \right) \frac{A_1}{r} e^{i(\omega t - kr)} + \left( \frac{\frac{1}{r} - ik}{i\omega \rho} \right) \frac{A_2}{r} e^{i(\omega t + kr)}. \quad (2.20)$$

Nel caso di onde sferiche l'impedenza acustica specifica dipende dal rapporto  $r/\lambda$ . Solo per valori molto grandi di tale rapporto il termine reattivo tende a zero e quello resistivo tende a  $\rho c$ . L'impedenza può essere quindi utilizzata per stabilire la distanza secondo la quale le onde sferiche generate dalla cassa possono ritenersi onde piane, e quindi riportare il problema al caso monodimensionale e senza riflessioni. Considerando la propagazione di un'onda senza riflessioni, la velocità e la pressione risultano in fase. Considerando la sola propagazione dell'onda nella direzione positiva della coordinata  $r$ , si può riscrivere l'impedenza acustica considerando la sola propagazione di interesse, come:

$$u = \left( \frac{\frac{1}{r} + ik}{i\omega \rho} \right) \frac{A_1}{r} e^{i(\omega t - kr)}. \quad (2.21)$$

L'impedenza acustica è quindi pari a:

$$Z_a = \rho c \left( \frac{k^2 r^2}{1 + k^2 r^2} + i \frac{kr}{1 + k^2 r^2} \right), \quad (2.22)$$



da cui si ottiene il valore del rapporto tra la reattanza e la resistenza acustica:

$$\frac{R}{X} = kr. \quad (2.23)$$

Questo rapporto è utile per poter approssimare l'onda generata da una sorgente sferica ad un'onda piana. Infatti, fissando il rapporto mostrato nell'equazione 2.23 si può calcolare la distanza dalla sorgente sonora sferica alla quale l'onda generata si può approssimare come onda piana. Tale distanza risulta pari a:

$$r = \frac{R}{X} = \frac{R}{X} \frac{c}{2\pi f}. \quad (2.24)$$

Nel diagramma in figura 2.1 vengono riportate le curve che mostrano la distanza della cassa acustica che genera il rumore in funzione del rapporto  $R/X$  al variare della frequenza di forzamento.

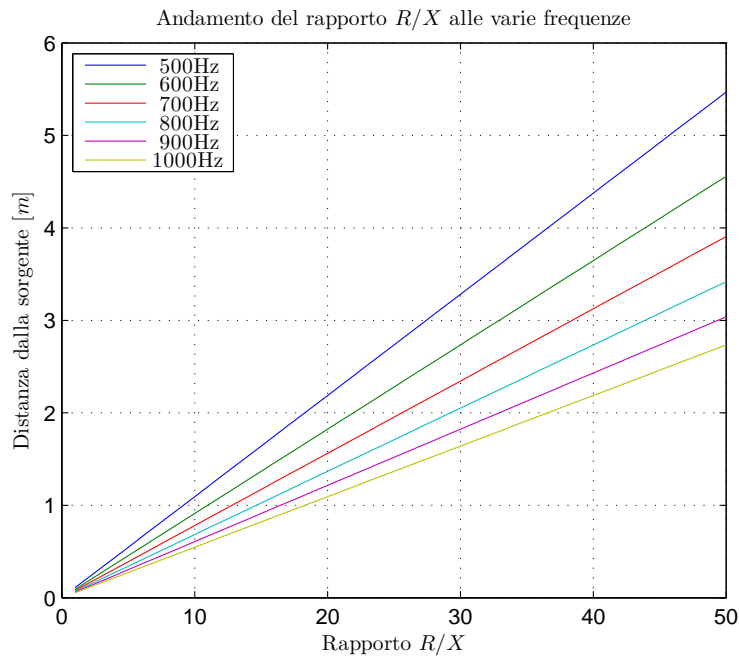


Figura 2.1: *Andamento del rapporto  $R/X$  in funzione della distanza, per varie frequenze.*

Si nota come, idealmente, per ottenere una buona approssimazione da onda sferica ad onda piana, sia necessario porsi ad una distanza di qualche metro. Tuttavia il grafico è puramente indicativo, in quanto le casse acustiche presentano di per se buone doti di direzionalità, che quindi riducono la distanza necessaria per tale approssimazione.

A partire dalla definizione dell'equazione delle onde e imponendo a questa le condizioni iniziali e al contorno è possibile ricavare le pulsazioni e i modi di vibrare dell'ambiente sonoro, nel caso monodimensionale. Essendo interessati alla soluzione a regime del sistema, non si ha più a che fare con un fenomeno propagativo, ma stazionario. Se si impongono le condizioni di Dirichlet al contorno, si ottiene la soluzione mostrata dall'equazione 1.45. Considerando la propagazione delle onde sonore all'interno di un condotto, tali condizioni al contorno equivalgono ad imporre pressione nulla alle estremità del condotto, e quindi un perfetto assorbimento dell'onda incidente. Questa è anche la soluzione per la propagazione delle onde in un condotto avente entrambe le estremità aperte, considerando trascurabile la differenza di pressione causata dalla propagazione delle onde al contorno, rispetto alla pressione statica del mezzo trasmissivo. Dall'equazione 1.45 è possibile definire i modi di vibrare dell'ambiente sonoro come:

$$\psi_m(x) = \sin\left(\frac{m\pi}{L}x\right). \quad (2.25)$$

Dall'equazione 1.46 è invece possibile definire i valori delle pulsazioni naturali dell'ambiente sonoro, che risultano:

$$\omega_m = \mu_m c = \frac{m\pi}{L}c, \quad (2.26)$$

da cui le frequenze naturali sono:

$$f_m = \frac{\omega_m}{2\pi} = \frac{m}{2L}c. \quad (2.27)$$

Imponendo invece come condizioni al contorno le condizioni di Neumann, si impone il valore della derivata della pressione rispetto allo spazio, nel punto considerato. Il significato fisico di queste condizioni è la perfetta riflessione dell'onda sonora ai limiti del condotto, che corrisponde ad un condotto chiuso in entrambe le estremità. Un esempio di queste condizioni è rappresentato dalle equazioni 2.28 e 2.29:

$$\left(\frac{\partial p}{\partial x}\right)_{(0,t)} = 0, \quad (2.28)$$

$$\left(\frac{\partial p}{\partial x}\right)_{(L,t)} = 0. \quad (2.29)$$

Dalle quali è possibile ricavare la soluzione dell'equazione delle onde e i modi di vibrare dell'ambiente sonoro:

$$\psi_m(x) = \cos\left(\frac{m\pi}{L}x\right). \quad (2.30)$$

Le frequenze naturali sono:

$$f_m = \frac{\omega_m}{2\pi} = \frac{m}{2L}c. \quad (2.31)$$

### Definizione di massima dell'ambiente sonoro

Dalla trattazione svolta nel paragrafo precedente si pongono le basi per un dimensionamento di massima di un ambiente sonoro avente le caratteristiche di propagazione di onde piane. Nella sezione 1.4.2 si sono mostrati alcuni metodi di abbattimento delle onde di pressione attivi. Tali sistemi prevedono la rilevazione del segnale del rumore attraverso un sensore, che misura quindi il segnale di riferimento. Il controllore elabora il segnale acquisito dal sensore, tipicamente un microfono, e comanda gli attuatori acustici che generano le opportune onde di pressione per dar luogo al meccanismo della cancellazione. Poiché risulta necessario garantire la causalità del controllo, il segnale di riferimento deve dare con un certo anticipo l'indicazione del rumore. Il tempo necessario al sistema di controllo per acquisire il segnale dal microfono di riferimento, calcolare l'uscita e far muovere la cassa di controllo è denominato come *ritardo di gruppo*. Il ritardo di gruppo dipende in generale da svariate cause tra cui:

- ritardo del microfono;
- ritardo del convertitore ADC (convertitore analogico-digitale);
- ritardo del processore DSP (tempi di calcolo del controllore);
- ritardo del convertitore DAC (digitale-analogico);
- ritardo della cassa di controllo.

Nella figura 2.2 viene schematizzato il sistema in esame composto da:

- un condotto;
- due casse acustiche: una cassa come sorgente di rumore e un'altra è come attuatore per il controllo;
- un microfono: posto a valle dell'attuatore di controllo misura il residuo di rumore presente all'uscita del condotto.

Le grandezze geometriche di interesse sono:

- $L$ : lunghezza del condotto.
- $l_1$ : larghezza della sezione del condotto.
- $d_1$ : distanza tra la sorgente del rumore e l'attuatore acustico di controllo.
- $d_2$ : distanza tra l'attuatore acustico di controllo e il microfono per la rilevazione del rumore residuo.

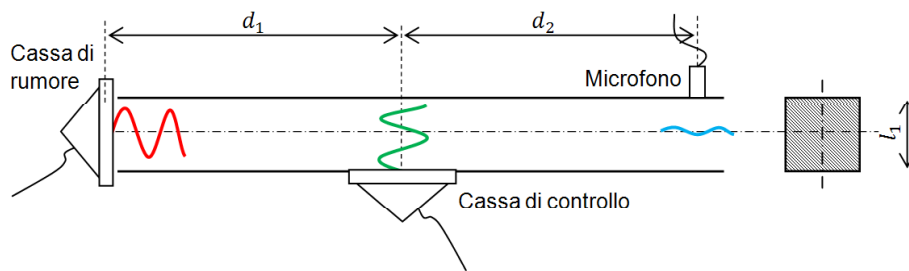


Figura 2.2: Schema del sistema fisico oggetto di studio.

La distanza  $d_1$  è determinata in funzione della velocità del suono e del ritardo di gruppo secondo la seguente disequaglianza:

$$d_1 > c\Delta t_G. \quad (2.32)$$

Considerando un generico hardware di controllo le cui caratteristiche verranno discusse nel paragrafo 2.2.3, si può ritenere che il ritardo di gruppo non sia superiore ad  $2 \text{ ms}$ , pertanto considerando la velocità del suono alla temperatura di  $20^\circ\text{C}$ , ossia  $343.4 \text{ m/s}$ , la lunghezza minima del condotto risulta essere pari a:

$$d_1 > 0.69 \text{ m}. \quad (2.33)$$

Dalla lunghezza minima del condotto appena determinata si devono considerare gli effetti di bordo, che possono sicuramente influenzare il comportamento fisico del sistema. Per ridurre la presenza di questi effetti la lunghezza del condotto viene fissata pari a  $L = 2 \text{ m}$ , in questo modo è possibile tenere in considerazione uno spazio aggiuntivo utile per l'inserimento e il bloccaggio dei diversi componenti.

Per avere un controllo semplice da attuare è necessario garantire che all'interno del condotto arrivi un disturbo sotto forma di onde piane. Si vuole evitare di introdurre

modi di ordine elevato poichè non hanno una distribuzione di pressione costante sulla sezione, ma dipendente dalla forma dell'ambiente sonoro e quindi la presenza di questi non consente di ricondursi al caso di propagazione monodimensionale. L'ipotesi di flusso monodimensionale comporta l'assunzione di  $l_1 \ll \lambda$ , pertanto, fissato il valore del rapporto tra la lunghezza d'onda e il diametro del tubo (inteso come lunghezza caratteristica della sezione del tubo, in funzione della forma della sezione) ed il valore della frequenza massima controllabile, si può definire il diametro minimo del tubo. Ad esempio scegliendo come frequenza massima della sorgente di rumore un valore di  $1200 \text{ Hz}$  e fissando un rapporto minimo  $\frac{\lambda}{l_1} \geq 2$ , si ottiene:

$$\frac{\lambda}{l_1} = \frac{c}{fl_1} \geq 2, \quad (2.34)$$

da cui il diametro minimo del tubo vale:

$$l_{1,min} = \frac{c}{2f} = 143 \text{ mm}. \quad (2.35)$$

Si approssima il valore ottenuto a  $l_1 = 150 \text{ mm}$ , scegliendo come forma della sezione una sezione quadrata. In questo modo risulta più semplice il montaggio delle casse acustiche e data la maggiorazione di  $l_1$  introdotta dall'approssimazione è possibile scegliere delle casse acustiche aventi il cono leggermente più grande.

Per la generazione del rumore sottoforma di onde piane è necessario mantenere una certa distanza tra la cassa acustica e l'imbocco del condotto. Tale distanza viene stimata in base al valore assunto dal rapporto tra la parte reale e la parte immaginaria dell'impedenza acustica specifica. Fissando il rapporto  $\frac{R}{X} = 5$  si ottiene secondo la relazione 2.24 una distanza pari a  $0.23 \text{ m}$  considerando come frequenza di generazione  $f = 1200 \text{ Hz}$ , e una distanza di  $0.91 \text{ m}$  considerando una frequenza di generazione di  $f = 300 \text{ Hz}$ . Dai risultati ottenuti si evince che, per la corretta esecuzione delle prove in sperimentali sarebbe opportuno disporre la cassa acustica preposta alla generazione del rumore su un supporto mobile, che possa essere avvicinato o allontanato dall'imboccatura del tubo a seconda delle esigenze.

Le dimensioni geometriche del condotto sono riassunte nella tabella 2.1. Nella tabella 2.2 si possono osservare invece le frequenze proprie teoriche del condotto.

In riferimento all'equazione 2.25 nella figura 2.3 sono indicati i primi tre modi di vibrare dell'ambiente sonoro.

$L$	$2 \text{ m}$
$l_1$	$150 \text{ mm}$
$d_1$	$1 \text{ m}$
$d_2$	$0 \div 1 \text{ m}$

Tabella 2.1: *Caratteristiche geometriche teoriche del condotto.*

Frequenze di risonanza	
Indice	frequenza [Hz]
1° modo	85.75
2° modo	171.5
3° modo	257.25
4° modo	343
5° modo	428.75
6° modo	514.5

Tabella 2.2: *Frequenze proprie teoriche del condotto.*

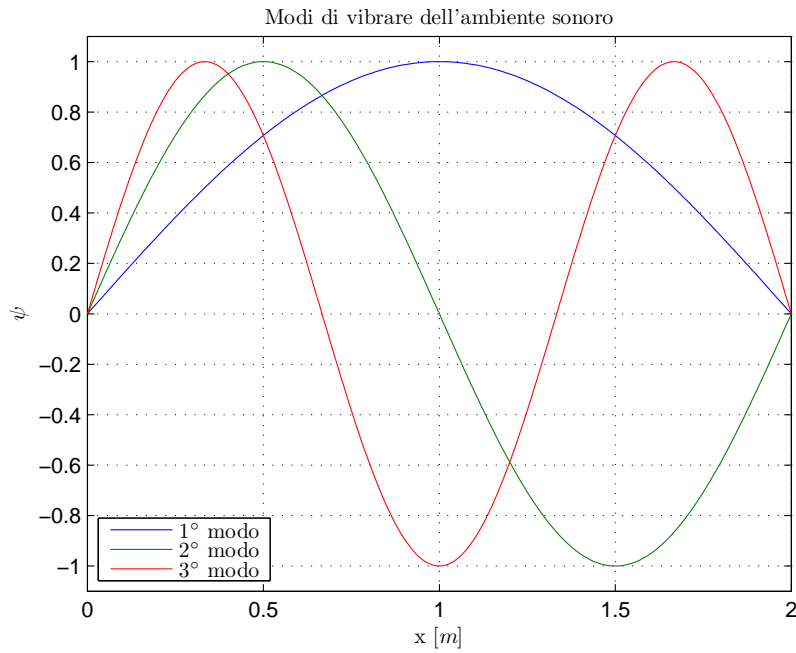


Figura 2.3: *Modi di vibrare dell'ambiente sonoro.*

### 2.1.1 Modellazione numerica del condotto

Con lo scopo di validare il dimensionamento del condotto, si definisce un modello ad elementi finiti dell'ambiente sonoro, avvalendosi del software FEM ANSYS. Nella figura 2.4 è mostrato il modello ad elementi finiti ottenuto, avente le seguenti caratteristiche:

- Dimensioni:  $2000 \times 150 \times 150 \text{ mm}$ ;

- Fluido: Aria;
- Densità:  $1.2 \frac{kg}{m^3}$ ;
- Velocità di propagazione del suono:  $343 \frac{m}{s}$ ;

Per eseguire l'analisi acustica si sceglie di utilizzare gli elementi finiti tridimensionali denominati FLUID30. questi elementi finiti hanno forma di parallelepipedi e presentano otto nodi, i loro gradi di libertà sono costituiti dalla pressione e dagli spostamenti nelle tre direzioni. La dimensione di questi elementi finiti è determinata dalla massima frequenza che si vuole analizzare con il modello. Infatti, ponendo di voler analizzare una frequenza massima pari a:

$$f_{maxANALISI} = 500 \text{ Hz}, \quad (2.36)$$

la dimensione massima degli elementi  $d$  dovrà rispettare la seguente condizione:

$$d \leq \frac{1}{10} \lambda_{min} = \frac{1}{10} \frac{c}{f_{maxANALISI}}, \quad (2.37)$$

ovvero dovrà essere almeno dieci volte più piccola della lunghezza d'onda minima di analisi. Tale condizione è necessaria per eseguire una corretta analisi in frequenza e per visualizzare correttamente i risultati ottenuti. Imponendo la condizione 2.37 si ottiene:

$$d \leq \frac{1}{10} \frac{343 \frac{m}{s}}{500 \text{ Hz}} = 0.069 \text{ m} \quad \implies \quad \hat{d} = 0.05 \text{ m}. \quad (2.38)$$

Dall'equazione 2.38 si ottiene la dimensione degli elementi, pari a  $\hat{d}$ . Per l'ambiente sonoro avente le dimensioni elencate, il numero di elementi per definire la geometria è pari a 360.

Al modello appena descritto vengono applicate le seguenti condizioni al contorno:

- Spostamento nullo nei nodi intermedi.
- Pressione nulla all'uscita del condotto.

All'ingresso del condotto viene invece applicata una variazione armonica di pressione di ampiezza pari a  $1 \text{ Pa}$  e frequenza variabile in un range da 0 a  $450 \text{ Hz}$ .

I risultati dell'analisi armonica sono mostrati nelle figure 2.5, 2.6, 2.7 e 2.8. Nella figura 2.5 è mostrato l'andamento della pressione alla frequenza  $f = 85.75 \text{ Hz}$ , pari alla frequenza di eccitazione del 1° modo di vibrare. L'andamento della pressione rispecchia l'andamento teorico mostrato in figura 2.3.

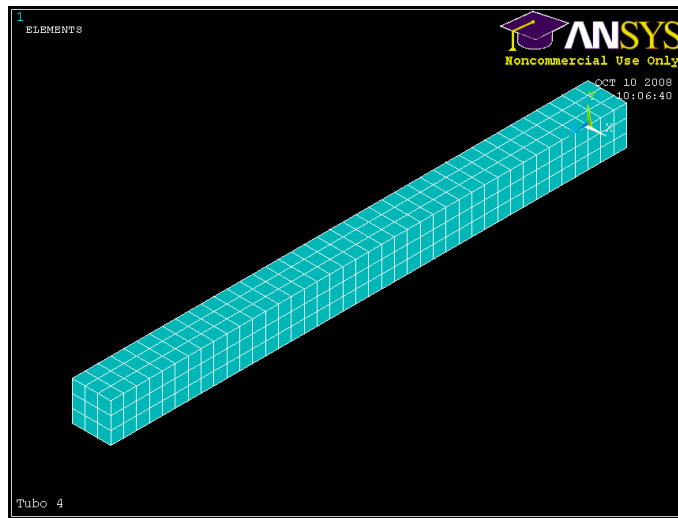


Figura 2.4: Definizione del modello dell'ambiente sonoro mediante metodo ad elementi finiti.

Analogamente le stesse osservazioni valgono per le figure 2.6 e 2.7, in cui sono mostrati gli andamenti delle pressioni per frequenze di eccitazione del condotto pari a 171.5 e 257.25  $Hz$ . In figura 2.8 è mostrato lo spettro di risposta in frequenza di un punto posto al centro della sezione del condotto, alla distanza di 0.25  $m$  dall'uscita del condotto, rispetto alla forzante armonica applicata alla sezione di ingresso del condotto a frequenza variabile. Nella figura si notano i picchi di pressione in corrispondenza delle frequenze di risonanza del 3°, 4° e 5° modo.

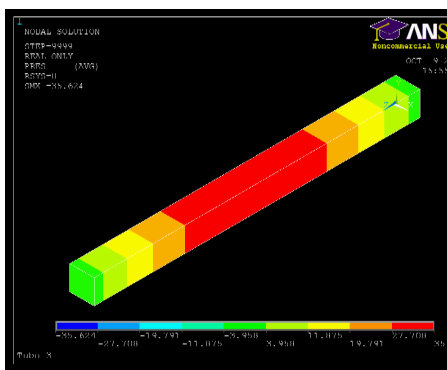


Figura 2.5: Analisi armonica dell'ambiente sonoro: 1° modo di vibrare.

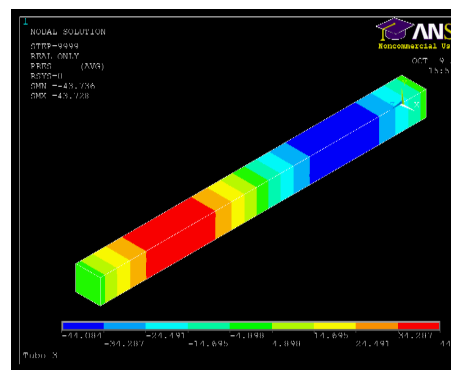


Figura 2.6: Analisi armonica dell'ambiente sonoro: 2° modo di vibrare.



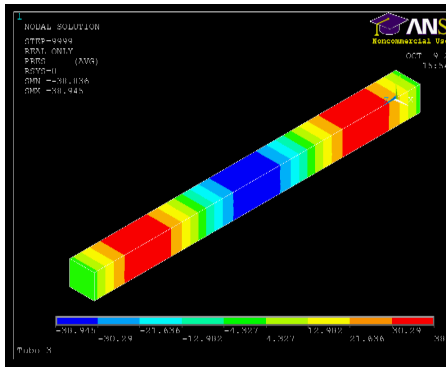


Figura 2.7: *Analisi armonica dell'ambiente sonoro: 3° modo di vibrare.*

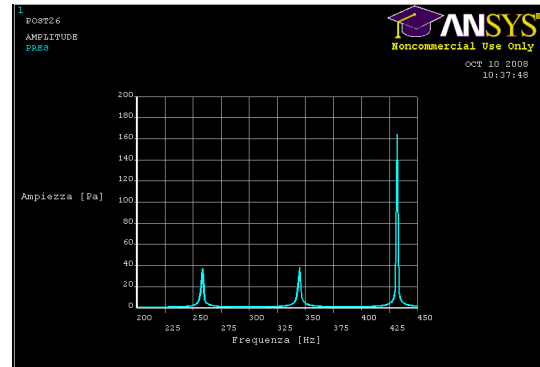


Figura 2.8: *Analisi armonica dell'ambiente sonoro: risposta in frequenza da 200 a 450 Hz.*

## 2.2 Sistema di controllo per l'abbattimento del rumore

L'approccio tradizionale per l'abbattimento del rumore prevede tecniche di isolamento passivo come l'utilizzo di materiali fonoassorbenti, cabine di protezione, barriere e sistemi risonanti (par. 1.4.1). Il vantaggio principale di tali dispositivi, oltre al fatto di non richiedere energia dall'esterno per il loro funzionamento, risiede nell'elevato livello di attenuazione del rumore in un ampio range di frequenze tuttavia, risultano particolarmente ingombranti, costosi e inefficaci nell'abbattimento di onde a basse frequenze. Nello sforzo per superare questi problemi, la tecnica del controllo attivo del rumore, attualmente sta ricevendo interessi considerevoli grazie anche allo sviluppo di micro-controllori low-cost. Il controllo attivo del rumore utilizza sistemi elettroacustici o elettromeccanici che cancellano il rumore della sorgente primaria, ossia il rumore indesiderato, basandosi sul principio di sovrapposizione degli effetti, come mostrato in figura 2.9.

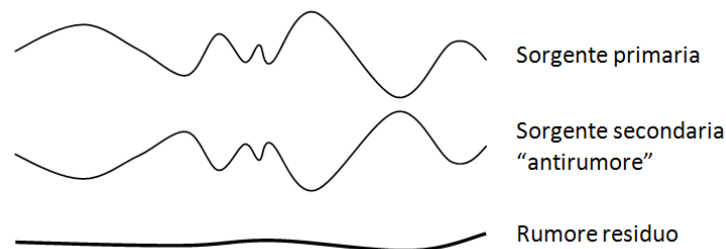


Figura 2.9: *Principio di sovrapposizione degli effetti.*

I sistemi di controllo attivo del rumore riducono efficacemente il rumore alle basse frequenze, dove i sistemi di abbattimento passivo risultano inadeguati o tendono ad essere molto costosi e ingombranti (norma UNI EN ISO 11690-2 [25]), in questo senso può essere conveniente, qualora l'applicazione lo consenta, combinare gli effetti di attenuazione in alta frequenza dei sistemi passivi con la tecnologia attiva per coprire il più ampio range di frequenze. Le performance nella cancellazione del rumore dipendono dalle caratteristiche degli elementi che costituiscono il sistema di controllo, quindi dall'accuratezza del sistema nella determinazione dell'ampiezza e della fase del segnale di *antirumore*.

Sebbene il principio di sovrapposizione delle onde acustiche è noto sin dal 1878<sup>1</sup>, la tecnologia del controllo attivo del rumore non ha riscontrato ad oggi una significativa implementazione. La lentezza dello sviluppo di tale tecnologia è in parte legata alla mancanza della conoscenza dei principi fisici coinvolti e alla difficoltà di combinare la multidisciplinarietà della tecnologia del controllo attivo che comprende un ampio numero di discipline tecniche tra cui i sistemi di controllo, l'elaborazione dei segnali, l'elettronica, l'acustica e le vibrazioni e in parte è dovuta al tempo necessario allo sviluppo di potenti micro-controllori. La figura 2.10 mostra gli elementi su cui si basa la tecnica del controllo attivo del rumore illustrando chiaramente la necessità di una sinergia tra diversi campi dell'ingegneria. In particolare, data la presenza di una sorgente di rumore e stabilita una zona (nota come *zona di silenzio desiderata*) entro la quale si vuole ottenere una cospicua riduzione del rumore generato dalla sorgente, si vuole realizzare un sistema di controllo costituito da sensori, attuatori e controllore. Il controllore in funzione dei segnali rilevati dai sensori presenti nel sistema dovrà, mediante degli opportuni algoritmi, comandare gli attuatori al fine di abbattere il rumore, rilevato dai sensori, che perviene nella zona di silenzio. L'applicazione del controllo attivo del rumore necessita di importanti considerazioni al fine di realizzare un sistema di successo; tali considerazioni riguardano la tipologia di sorgente, il tipo rumore, le caratteristiche del campo di propagazione e degli elementi costituenti il sistema di controllo.

---

<sup>1</sup>Le prime osservazioni sul principio di cancellazione del rumore risalgono al 1878 grazie a Sir William Thomson con l'invenzione dei primi telefoni (*Bell telephone*), la tecnica di abbattimento del rumore non fu più sviluppata fino al 1930 quando l'ingegnere francese H. Coanda documentò e poi brevettò l'idea di cancellazione del rumore grazie all'interferenza distruttiva (principio di sovrapposizione degli effetti). La sua idea si basa su un sistema elettro-acustico, costituito da un microfono, un amplificatore e un altoparlante per generare un segnale con fase opposta a quella del rumore da cancellare.

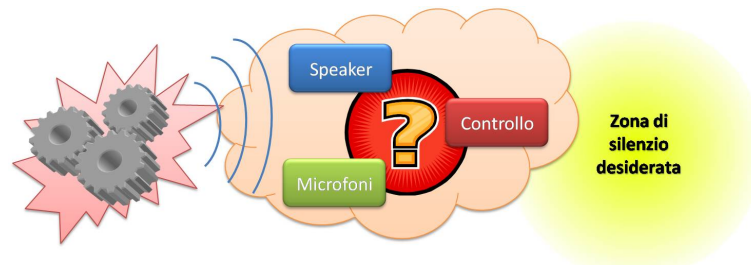


Figura 2.10: Elementi di un sistema di controllo attivo del rumore.

A seconda della tipologia di rumore e di campo acustico possono essere utilizzate differenti tecniche di controllo del rumore. Esistono principalmente due tipologie di rumore, rumore a banda larga (*broadband*) e rumore a banda stretta (*narrowband*). Il rumore di tipo *broadband* è ad esempio quello causato da fenomeni turbolenti, i quali sono totalmente casuali e hanno un'energia distribuita uniformemente su una certa banda di frequenze. Classici esempi di rumori *broadband* sono quelli del traffico stradale o quelli di tipo impulsivo come un'esplosione. I rumori di tipo *narrowband* concentrano la loro energia attorno a specifiche frequenze e sono spesso correlati alla presenza di parti rotanti. Esempi di rumori *narrowband* sono quelli generati dai motori a combustione interna, dalle turbine, dai compressori, dalle pompe, etc.

### 2.2.1 Schema di controllo

Le tecniche di controllo attivo del rumore, anche dette ANC (*active noise control*), si basano principalmente su due schemi di funzionamento: tecniche di tipo *feedforward* (figura 2.11) e tecniche di tipo *feedback* (figura 2.12).

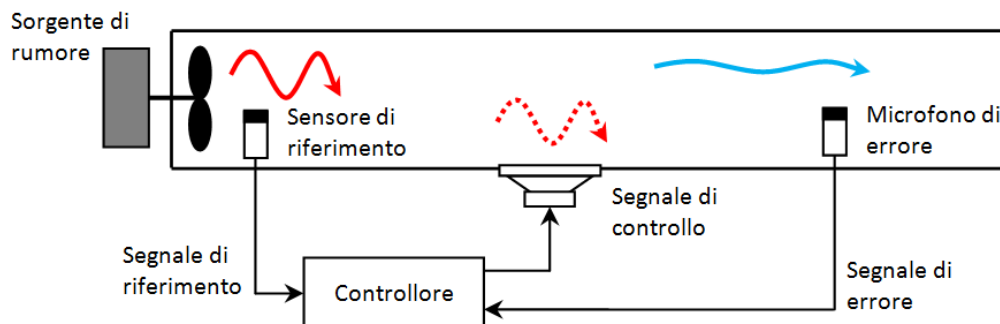


Figura 2.11: Sistema di controllo attivo del rumore in *feedforward*.

Il sistema di controllo attivo del rumore in feedforward è costituito da un sensore di riferimento (solitamente un microfono) che permette di rilevare il rumore emesso dalla sorgente, da una sorgente di controllo che genera il segnale di antirumore e da un microfono di errore che misura l'efficacia di funzionamento del controllore. La sorgente di controllo costituita solitamente da un'altoparlante viene comandata dall'elettronica la quale filtra e campiona il segnale letto dal microfono di riferimento e mediante un algoritmo determina il segnale di controllo. Il segnale del microfono di errore viene rilevato ed inviato all'elettronica, la quale mediante un opportuno algoritmo modifica i parametri del regolatore al fine di minimizzare il rumore misurato dal microfono di errore.

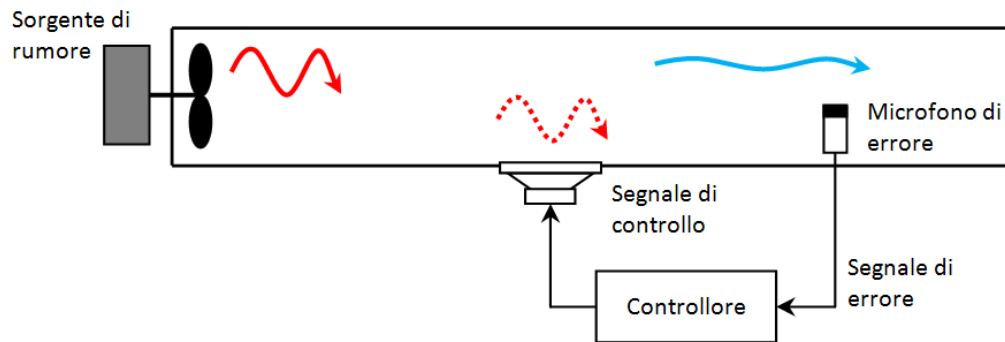


Figura 2.12: *Sistema di controllo attivo del rumore in feedback.*

Il tempo necessario al controllore per la determinazione del segnale di controllo deve essere inferiore, nel caso di rumore a banda larga, al tempo che impiega il rumore a propagarsi dal sensore di riferimento alla sorgente di controllo, mentre nel caso di rumore tonale il massimo tempo di processamento può essere molto superiore poiché il segnale è ripetitivo. Rispetto ai controllori in feedback, quelli in feedforward presentano spesso uno svantaggio legato al feedback dell'uscita della sorgente di controllo verso il sensore di riferimento. Sebbene questo effetto può essere considerato all'interno del controllore, attraverso l'algoritmo di controllo e i filtri, spesso si instaurano delle instabilità nel controllo. Esistono particolari accorgimenti che consentono di evitare o per lo meno ridurre tale problema. È possibile ad esempio, nel caso di rumore generato da macchine rotanti, utilizzare un sensore di riferimento non acustico come una dinamo tachimetrica. Il sistema di controllo dovrà quindi essere in grado di ricostruire il segnale di rumore di riferimento a partire dal segnale di velocità angolare. Lo svantaggio di utilizzare questa tecnica è ovviamente l'impossibilità di attenuare le armoniche del rumore non multiple

della frequenza di rotazione della macchina. In questo caso infatti, le armoniche rilevate dal sensore di errore saranno scorrelate dal segnale di riferimento e l'uscita del sistema di controllo non potrà tenerne conto. Pertanto l'utilizzo delle dinamo tachimetriche è utile nel caso in cui il contenuto spettrale del rumore è concentrato sulle armoniche o sulle sotto-armoniche della frequenza di rotazione della macchina.

Il sistema di controllo attivo del rumore di tipo feedback (figura 2.12) è costituito dai medesimi componenti del controllo feedforward tranne che per il microfono di riferimento che, in questo caso, non è presente. Mentre i sistemi di controllo in feedforward si affidano a misure predittive per generare un appropriato segnale di antirumore, lo scopo dei sistemi di controllo in feedback è quello di attenuare il rumore acquisito dal sensore di errore, quando questo è posto a valle dell'attuatore acustico di controllo. I sistemi di controllo in feedback determinano il segnale di controllo sulla base del solo segnale del sensore di errore, in modo da minimizzare il rumore acquisito da quest'ultimo, mentre i sistemi di controllo in feedforward utilizzano il segnale del sensore di errore solo in aiuto al controllore al fine di ottimizzare le sue performance nella minimizzazione del segnale di errore. La banda in frequenze entro la quale risulta efficace l'utilizzo di un controllore in feedback risulta limitata dal ritardo che vi è tra l'uscita del segnale di controllo da parte dell'altoparlante e la lettura del segnale di errore (ingresso del controllore). La larghezza di banda è direttamente proporzionale al reciproco di questo ritardo. Lo sfasamento associato a questo ritardo porta il sistema di controllo, alle alte frequenze, da feedback negativo a feedback positivo rendendo pertanto instabile il sistema. Questo è il principale svantaggio associato all'utilizzo di un controllore in feedback, infatti quando questo sfasamento eccede i  $180^\circ$  il feedback positivo non fa che incrementare il livello di rumore che risulterà limitato solo dalla potenza dell'altoparlante e dalla sua amplificazione. Per limitare i ritardi e quindi massimizzare le performance e la stabilità di un controllore in feedback si dovrebbe posizionare la sorgente di controllo molto vicina al sensore di errore, tuttavia anche questa soluzione presenta lo svantaggio che il sensore di errore è vicino al "near field" dell'altoparlante e quindi può capitare che la pressione acustica lontano dal sensore di errore ("far field") non venga significativamente ridotta; in queste condizioni è possibile ottenere una cancellazione solo in prossimità del sensore di errore a causa della non uniformità del "near field". Questo non è ovviamente un problema in alcuni sistemi come ad esempio nelle cuffie attive dove l'orecchio umano è molto vicino al sensore di errore.

A fronte dei consistenti vantaggi di abbattimento, le tecniche di controllo attivo del rumore presentano ad oggi le seguenti limitazioni:

- Il campo sonoro da controllare non deve presentare caratteristiche spaziali particolarmente complesse e un andamento pressochè stazionario nel tempo. Classico esempio di propagazione in campo non particolarmente complesso è quello della propagazione del rumore all'interno di un condotto, dove la propagazione delle onde avviene lungo una direzione preferenziale.
- Il controllo attivo del rumore risulta efficace se la lunghezza d'onda del rumore è elevata rispetto alle dimensioni del campo acustico (ambiente in cui si propagano le onde).
- La generazione del segnale di controllo è tanto più complessa quanto più estesa è la distribuzione in frequenza del rumore da controllare. I risultati migliori sono quelli che possono essere ottenuti in presenza di rumori costituiti da toni puri o a banda stretta.
- Le non linearità dei trasduttori, delle sorgenti di controllo e in generale dell'hardware utilizzato nel sistema di controllo, comportano l'inevitabile presenza di un ritardo di fase che cresce all'aumentare della massima frequenza del rumore che si intende controllare. Questo ritardo può portare il sistema di controllo in condizioni di instabilità. Il fenomeno dell'instabilità è quello che maggiormente limita l'utilizzo dei sistemi di controllo basati sullo schema in feedback (figura 2.12).
- Il controllo attivo del rumore può portare elevati benefici nella posizione di silenzio desiderata, ma altrove può comportare un incremento del livello di pressione acustica. Per ridurre tale effetto può essere necessario incrementare il numero delle sorgenti di controllo, ma se il campo da controllare è molto complesso difficilmente si riesce ad evitare questo fenomeno.

La disposizione fisica degli elementi che costituiscono il sistema di controllo è un elemento fondamentale per la realizzazione di efficiente sistema di attenuazione del rumore. Nella presente tesi viene proposto il controllo del rumore basato sulla tecnica di feedback poiché si intende abbattere rumori con contenuto spettrale concentrato attorno ad una specifica frequenza o comunque rumori di tipo mono-armonico o multi-armonico di tipo stazionario.

### 2.2.2 Sensori e attuatori per il controllo

Uno degli elementi fondamentali per la buona riuscita di un sistema di controllo attivo del rumore riguarda la corretta scelta dei sensori e degli attuatori acustici. I sensori più utilizzati in ambito acustico sono i microfoni mentre, per quanto riguarda gli attuatori vengono solitamente utilizzati comuni altoparlanti. Quando si seleziona un microfono, prima di tutto è importante capire quali sono i requisiti necessari al sistema di controllo e come questi parametri influiscono sulle prestazioni richieste al microfono. In particolare ogni microfono è ottimizzato per effettuare particolari misure; i parametri di maggior interesse per la scelta di un sensore da utilizzare in un sistema di controllo attivo del rumore sono il tipo di campo sonoro, la risposta in frequenza e il range dinamico, ossia del livello di rumore in grado di essere rilevato senza che il sensore vada in saturazione. Talvolta questi parametri non possono essere scelti in modo indipendente ma devono essere scelti congiuntamente. Infatti la risposta in frequenza e il range dinamico di un microfono hanno un andamento opposto come mostrato nelle figure 2.13 e 2.14.

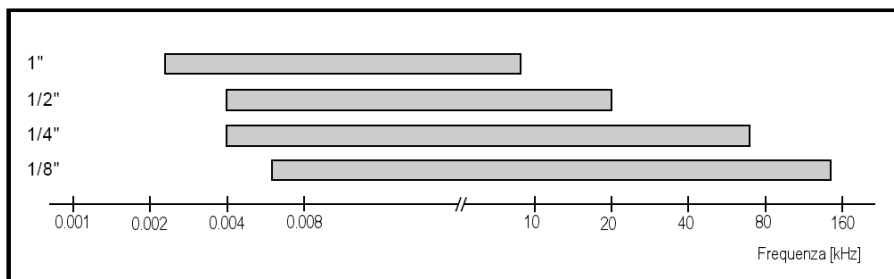


Figura 2.13: *Limiti della risposta in frequenza di quattro microfoni di differenti taglie: 1/8, 1/4, 1/2 e 1".*

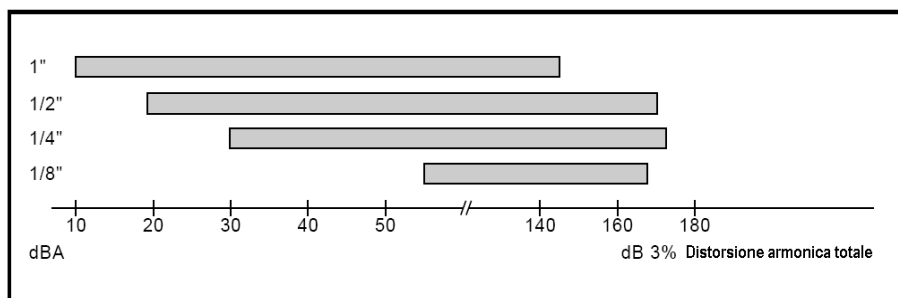


Figura 2.14: *Range dinamico di quattro microfoni di differenti taglie: 1/8, 1/4, 1/2 e 1".*

Nel caso in cui viene richiesto un sensore in grado di rilevare un rumore ad elevata frequenza e un'ampiezza dinamica molto elevata occorrerà privilegiare uno o l'altro parametro poiché, ad una maggiore frequenza rilevabile dal microfono, corrisponde una minore ampiezza che può essere rilevata prima che il sensore vada in condizione di saturazione. Nel caso del controllo attivo del rumore, tale scelta si riconduce solitamente alla sola ampiezza dinamica, poiché la minima e la massima frequenza del rumore che potrà essere attenuato con questa tecnologia sarà sicuramente compresa all'interno del range di frequenze in grado di essere rilevate da un comune microfono.

Vengono di seguito riassunte le principali caratteristiche utili alla scelta dei sensori acustici per il controllo attivo del rumore:

- Risposta in frequenza: sebbene i microfoni siano ottimizzati per particolari misure, presentano comunque un ampio intervallo in cui è possibile operare. La risposta in frequenza deve essere considerata in relazione ad altri criteri di selezione, come ad esempio il campo sonoro. La figura 2.15 riporta la risposta in frequenza di un classico microfono utilizzato per misurazioni acustiche.

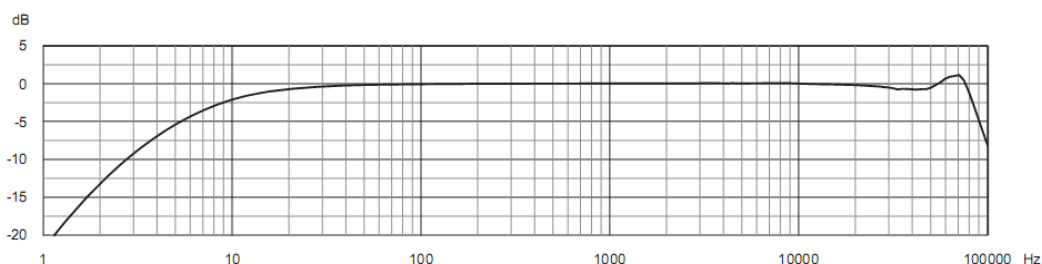


Figura 2.15: *Risposta in frequenza di un classico microfono per misurazioni acustiche (Fonte: Bruel & Kjaer).*

- Campo sonoro: per misure effettuate lontano da superfici riflettenti, o in ambienti chiusi acusticamente ben smorzati, i microfoni a campo libero sono la scelta migliore. Alternativamente, per misure effettuate in piccole cavità o vicino a superfici riflettenti devono essere scelti microfoni a campo di pressione. Per misure in ambienti chiusi dove c'è molto riverbero, esistono dei microfoni a campo di pressione adattati per avere una buona risposta anche con incidenza qualsiasi rispetto al campo sonoro, questo poiché questi microfoni hanno una risposta più costante in frequenza rispetto ai microfoni a campo libero. A tal fine per



il controllo attivo del rumore all'interno di condotti si scelgono solitamente dei microfoni a campo di pressione.

- Intervallo dinamico: il limite inferiore dell'intervallo dinamico è dettato essenzialmente dal rumore inerente del microfono e del preamplificatore (combinazione del rumore elettrico e termico proveniente dal microfono). Il limite superiore è dettato dal massimo livello di pressione sonora in cui si abbia il 3 % di distorsione armonica totale. Per quanto riguarda il controllo attivo del rumore il problema principale può presentarsi nel caso in cui il rumore che si vuole rilevare col microfono ha un'ampiezza molto bassa pertanto occorre fare particolare attenzione al limite minimo dell'intervallo dinamico in grado di essere rilevato dal microfono; il limite massimo difficilmente viene raggiunto nei comuni sistemi di controllo del rumore a meno che il microfono venga posizionato in zone molto prossime alla sorgente di rumore.

Come per la scelta dei sensori, una delle caratteristiche più importanti nella scelta degli attuatori acustici è il range di frequenze del suono che questi dispositivi sono in grado di generare. Nelle tecniche di controllo attivo nei condotti gli attuatori utilizzati per la generazione del segnale di controllo sono solitamente i classici altoparlanti. La classificazione degli altoparlanti disponibili commercialmente viene solitamente effettuata in funzione del range di frequenze come:

- Tweeter: altoparlante in grado di riprodurre suoni ad elevata frequenza, tipicamente nel range  $2000 \div 20000 \text{ Hz}$ , anche se ne esistono alcuni in grado di rispondere fino alla frequenza di  $45 \text{ kHz}$ .
- Mid-range: altoparlante in grado di riprodurre suoni nel range  $300 \div 5000 \text{ Hz}$ .
- Woofer: altoparlante in grado di coprire il range di frequenze acustiche basse, medio-basse, generalmente nel range  $40 \div 1000 \text{ Hz}$ .
- Full-range: altoparlante progettato per riprodurre suoni in un range di frequenze il più ampio possibile, al fine di coprire con un unico dispositivo tutto il range di frequenze udibili dall'uomo. Generalmente questi altoparlanti sono in grado di riprodurre suoni nel range  $100 \div 20000 \text{ Hz}$ . A causa delle loro caratteristiche costruttive questi altoparlanti hanno delle performance inferiori rispetto ai precedenti e solitamente non vengono utilizzati nei dispositivi di riproduzione di alta fedeltà.

Un altro importante parametro nella scelta di un altoparlante è la sua potenza intesa come potenza supportata. In generale è possibile affermare che un'altoparlante con potenza maggiore è in grado di produrre un suono di ampiezza superiore solamente nel caso in cui l'amplificatore, cui viene collegato l'altoparlante, è in grado di fornire tale potenza. Un parametro più utile per valutare l'efficienza di un altoparlante è la sua sensibilità, espressa solitamente in  $dB$ , che indica il livello di pressione sonora prodotta con 1 *Watt* di potenza elettrica ad 1 *m* di distanza. Generalmente una cassa di media efficienza ha una sensibilità compresa tra 87 e 90  $dB$ . Parallelamente alla potenza supportata da un altoparlante è necessario considerare la sua impedenza elettrica. Solitamente l'impedenza nominale degli altoparlanti è pari a 4 o 8 *Ohm*, tuttavia l'andamento dell'impedenza reale di un altoparlante varia in funzione della frequenza del suono che viene generato. Occorre pertanto fare attenzione che, ad una minore impedenza dell'altoparlante, consegue una maggiore corrente che scorre nel circuito di amplificazione e che potrebbe pertanto danneggiarlo. Ovviamente ad una minore impedenza dell'altoparlante corrisponde una maggiore efficienza poiché, a parità di potenza fornita all'altoparlante, questo sarà in grado di emettere un suono con un livello di pressione acustica superiore.

Da quanto appena mostrato risulta necessario che, nella scelta di un sensore o di un attuatore acustico, occorre effettuare la scelta delle caratteristiche che risultano di maggior interesse al fine di realizzare un sistema di controllo con delle performance soddisfacenti.

### 2.2.3 Hardware di controllo

Le caratteristiche necessarie all'hardware di controllo vengono scelte in funzione del tipo di algoritmi che si intendono utilizzare per la realizzazione del sistema di controllo. Le prestazioni di abbattimento del rumore sono fortemente legate alle prestazioni dell'hardware che viene utilizzato, in particolare con alcune tipologie di hardware è possibile eseguire algoritmi basati su filtri digitali adattativi che possono richiedere prestazioni di calcolo estremamente elevate. Gli algoritmi di controllo del rumore che si intendono implementare e testare verranno formulati per un abbattimento di rumori di tipo stazionario che non variano dal momento in cui viene attivato il sistema di controllo, questo consente di utilizzare degli hardware di controllo con prestazioni di calcolo nettamente inferiori rispetto a quelli necessari per l'implementazione di filtri digitali adattativi a scapito di una minore flessibilità del sistema di controllo. Le

prestazioni dell'hardware di controllo, di interesse per gli algoritmi di controllo che verranno implementati in seguito, possono essere valutate in funzione delle seguenti caratteristiche:

- **Frequenza processore:** questo parametro indica la frequenza di clock del processore ossia il tempo minimo garantito per eseguire un singola operazione. Maggiore è la frequenza del processore è minore è il tempo necessario per eseguire le operazioni dell'algoritmo di controllo. Questa caratteristica si rileva estremamente importante per gli algoritmi che utilizzano filtri adattativi dove viene solitamente richiesta l'esecuzione di un elevato numero di operazioni in un tempo molto ridotto. La frequenza del processore non assicura la corretta esecuzione del codice di controllo entro un determinato periodo di tempo. È necessario avvalersi di un ambiente software che garantisca il determinismo temporale delle operazioni. Tale ambiente software è costituito dal sistema operativo Real-Time. In tale ambiente viene elaborato il codice di controllo e viene assicurata l'esecuzione del ciclo di controllo entro un determinato intervallo di tempo. In questo modo viene garantita la banda passante del sistema di controllo.
- **Numero di canali analogici:** maggiore è il numero di canali a disposizione e maggiore potrà risultare la complessità dell'algoritmo di controllo, quindi sarà possibile in generale conseguire performance di abbattimento superiori. Infatti un numero superiore di ingressi analogici consente di realizzare un sistema di controllo multi-input (MI) che tiene conto, ad esempio, del valore di pressione acustica in più punti rilevata grazie all'utilizzo di più microfoni. Oltre alla disponibilità di differenti ingressi analogici, un hardware per il controllo dispone di uno o più canali per le uscite analogiche in modo da comandare opportunamente una o più sorgenti di controllo. Anche in questo caso, un numero superiore di uscite analogiche consente di comandare più attuatori realizzando un sistema di controllo multi-output (MO).
- **Fondoscala adattabile:** questa caratteristica consente di adattare il fondoscala  $FS$  dei convertitori in funzione dei sensori e degli attuatori utilizzati. Solitamente il valore del fondoscala può essere adattato in range prefissati (tipici range sono:  $0 \div 10 \text{ Volt}$ ,  $-10 \div 10 \text{ Volt}$ ,  $-5 \div 5 \text{ Volt}$ ,  $-2.5 \div 2.5 \text{ Volt}$ ).
- **Risoluzione dei canali analogici:** tale parametro è legato al numero di bit  $n_b$  del convertitore (ADC o DAC) dei canali. Maggiore è il numero di bit del ca-

nale è maggiore è la risoluzione. Per un convertitore ADC la risoluzione viene determinata utilizzando la seguente relazione:

$$r = \frac{FS}{2^{n_b}}. \quad (2.39)$$

- **Frequenza di campionamento:** questa caratteristica si riferisce ai canali analogici di acquisizione. In particolare la frequenza di campionamento  $f_s$  è la frequenza alla quale è possibile campionare i valori discreti del segnale analogico fornito dai sensori. Maggiore è la frequenza di campionamento, maggiore è la frequenza rilevabile dal segnale fornito dal sensore, dando la possibilità all'algoritmo di controllo di riconoscere segnali con contenuto spettrale ad alta frequenza. La massima frequenza rilevabile dal sensore deriva dal teorema sul campionamento di Shannon il quale afferma che un segnale analogico variabile con continuità può essere campionato e successivamente riprodotto in modo esatto da valori discreti nel tempo attraverso un algoritmo di interpolazione se la frequenza di campionamento è almeno pari al doppio della banda del segnale. Il teorema di Shannon si riconduce pertanto alla seguente relazione dove  $f_{sig}$  è la massima frequenza che si intende rilevare dal segnale acquisito dal sensore:

$$f_s \geq 2f_{sig}. \quad (2.40)$$

- **Condizionamento:** alcuni hardware di controllo consentono di acquisire i segnali dei sensori con un opportuno condizionamento atto a soddisfare differenti requisiti. Uno dei requisiti fondamentali è l'acquisizione del segnale del sensore garantendo l'assenza di componenti in frequenza superiori alla frequenza di campionamento dell'hardware. La presenza di queste componenti causano il problema dell'aliasing che si riconduce in una errata conversione digitale del segnale analogico del sensore. Per evitare questo fenomeno alcuni hardware dispongono di banchi di filtri passa-basso che consentono di attenuare le componenti in frequenza del segnale del sensore al di sopra della frequenza di taglio del filtro utilizzato riducendo il problema dell'aliasing. Qualora nell'hardware di controllo non siano disponibili questi filtri è necessario cautelarsi realizzando opportuni filtri analogici da posizionare tra i sensori e i canali di ingresso analogico dell'hardware di controllo.

L'hardware di controllo viene fornito con un opportuno software che consente la comunicazione tra l'utente e l'hardware, la scrittura del codice di controllo nei comuni linguaggi di programmazione e la successiva conversione del codice in linguaggio macchina sulla RAM dell'hardware di controllo.

## 2.3 Algoritmi di controllo

Gli algoritmi di controllo che verranno implementati fanno riferimento allo schema di controllo in feedback mostrato in figura 2.12. La sorgente di rumore, costituita da un comune altoparlante, genera delle onde piane di pressione che vengono convogliate all'interno del condotto. La cassa acustica di controllo, situata nella mezzeria del condotto viene comandata in modo tale da abbattere il rumore generato dalla prima sorgente. Per la realizzazione dell'algoritmo di controllo vengono presentati quattro differenti algoritmi basati sulla minimizzazione di una funzione obiettivo. Verranno presi in considerazione diversi funzionali, basati su differenti grandezze come il valore quadratico medio (RMS) della pressione sonora, o l'ampiezza spettrale delle armoniche che compongono il segnale da abbattere. Tali grandezze vengono determinate a partire dalla misura della pressione sonora attraverso il microfono di errore posto all'estremità del condotto. Il segnale rilevato dal microfono di errore viene inviato al sistema di controllo che provvede, attraverso un opportuno algoritmo di minimizzazione di un funzionale, a determinare il segnale da fornire alla cassa di controllo.

Prima di procedere alla descrizione delle quattro tecniche di controllo è necessario analizzare dal punto di vista analitico l'influenza dei parametri di controllo sulla funzione obiettivo che si intende minimizzare. Lo studio analitico consente quindi di prevedere il comportamento di un possibile apparato sperimentale e scegliere in modo opportuno i passi per lo sviluppo degli algoritmi. Nei paragrafi seguenti verranno proposti i quattro algoritmi e la loro analisi attraverso lo studio delle funzioni obiettivo in un opportuno ambiente di modellazione analitica.

Le tecniche di controllo che si intendono sviluppare prevedono l'abbattimento di rumori stazionari di tipo mono-armonico o multi-armonico. Il fatto di considerare il rumore come costituito da una o più armoniche permette di ridurre notevolmente il numero dei parametri di controllo infatti, considerando il rumore costituito da una o più sinusoidi, i parametri del segnale di controllo saranno le ampiezze e le fasi delle singole sinusoidi riconosciute all'interno del segnale di rumore. Essendo il segnale di controllo costituito dalla somma di toni puri, l'algoritmo di controllo dovrà essere in

grado di determinare le ampiezze e le fasi del segnale da fornire all'altoparlante di controllo al fine di minimizzare la funzione obiettivo prescelta.

### 2.3.1 Tecnica di controllo ARMS

Prescindendo dalla fisica del problema, che risulta alquanto complessa, si vuole mostrare la possibilità di abbattere un rumore che si propaga all'interno di un generico condotto considerando un semplice problema di ricerca del minimo di una funzione. Per questa tecnica di controllo la funzione da minimizzare è rappresentata dal valore quadratico medio (*RMS*) della somma dei segnali, costituiti da due toni puri, generati dalle casse di rumore e di controllo. Per semplicità questa tecnica viene denominata ARMS (Analisi dell'RMS).

Al fine di sviluppare correttamente l'algoritmo di minimizzazione del rumore è necessario studiare l'andamento dell'RMS al variare dei parametri del segnale di controllo. In particolare una volta noto il segnale di rumore e quindi noti i suoi parametri, si vogliono trovare i parametri da assegnare al segnale di controllo facendo in modo di collocarsi in una posizione di minimo della funzione RMS (Appendice A.1). Si consideri ad esempio di generare con la cassa di rumore un tono puro  $s_{rum}$  costituito da un segnale sinusoidale dato dalla seguente relazione:

$$s_{rum} = A_{rum} \sin(2\pi f_{rum}t + \phi_{rum}), \quad (2.41)$$

dove:

- $A_{rum}$  è l'ampiezza del segnale di rumore;
- $f_{rum}$  è la frequenza del segnale di rumore;
- $\phi_{rum}$  è la fase del segnale di rumore;
- $t$  è il tempo.

I parametri da assegnare al segnale di controllo sono ampiezza  $A_{cont}$  e fase  $\phi_{cont}$  e vengono scelti in modo da minimizzare il valore quadratico medio della somma dei segnali di rumore e di controllo. Si assume la frequenza del segnale di controllo pari alla frequenza del segnale di rumore, poiché alla base del problema si vuole annullare il rumore in ingresso al sistema fornendo un segnale di controllo che risulta essere in controfase al primo. Il segnale rilevato dal microfono di errore  $s_{rum+cont}^{i,j}$  risulterà pari

alla somma del segnale generato dalla cassa di rumore e quello generato dalla cassa di controllo:

$$s_{rum+cont}^{i,j} = A_{rum} \sin(2\pi f_{rum}t + \phi_{rum}) + A_{cont}^i \sin(2\pi f_{cont}t + \phi_{cont}^j), \quad (2.42)$$

dove gli apici  $i$  e  $j$  identificano differenti valori dei parametri di controllo  $A_{cont}^i$  e  $\phi_{cont}^j$ . Il valore quadratico medio si ottiene come:

$$RMS^{i,j} = \sqrt{\frac{\int_0^T \left(s_{rum+cont}^{i,j}\right)^2 dt}{T}} \approx \sqrt{\frac{\sum_{t=0}^T \left(s_{rum+cont}^{i,j}\right)^2}{T}}, \quad (2.43)$$

dove l'approssimazione permette l'implementazione a tempo discreto della relazione 2.43 in un comune sistema di controllo.

Prima di procedere allo sviluppo dell'algoritmo è interessante eseguire una mappa analitica del dominio entro cui l'algoritmo di controllo si troverà ad operare. Nell'appendice C.1 viene riportato lo script MatLab per la realizzazione della mappatura in oggetto. La mappatura viene effettuata considerando, per il segnale di rumore una sinusoide a parametri costanti, mentre per il segnale di controllo un segnale sinusoidale con parametri variabili in determinati range. I valori dei parametri assegnati ai segnali di rumore e di controllo per la modellazione della funzione 2.43 sono i seguenti:

- Segnale di rumore: sinusoide con ampiezza pari a  $A_{rum} = 1 Pa$ , fase pari a  $\phi_{rum} = 60^\circ$  e frequenza pari a  $f_{rum} = 1000 Hz$ .
- Segnale di controllo: sinusoide la cui ampiezza variabile nel range  $-1.5 \leq A_{cont} \leq 1.5 Pa$  e con fase nel range  $0^\circ \leq \phi_{cont} \leq 360^\circ$ .

Nelle figure 2.16 e 2.17 viene mostrato l'andamento analitico della funzione obiettivo 2.43 in funzione dei parametri di controllo  $A_{cont}$  e  $\phi_{cont}$ . Dall'andamento del valore dell'RMS si nota che esistono due soluzioni (ossia due minimi), in particolare si può notare che il grafico presenta una simmetria rispetto all'asse delle ampiezze secondo la seguente relazione:

$$RMS(\bar{A}_{cont}, \bar{\phi}_{cont}) = RMS(-\bar{A}_{cont}, \bar{\phi}_{cont} + k\pi) \quad k \in N; \quad (2.44)$$

questo può essere compreso analiticamente considerando che un'onda sinusoidale ad ampiezza positiva è pari ad un'altra onda con stessa ampiezza in modulo ma di segno opposto e con sfasamento di  $180^\circ$  rispetto alla prima onda:

$$1. \Rightarrow A_{cont} = -|\bar{A}|, \quad \phi_{cont} = \bar{\phi} \quad (2.45)$$

$$2. \Rightarrow A_{cont} = |\bar{A}|, \quad \phi_{cont} = \bar{\phi} + 180^\circ. \quad (2.46)$$

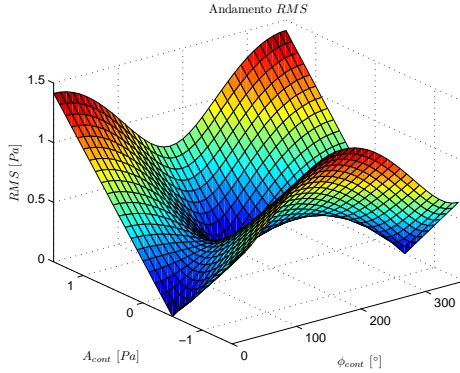


Figura 2.16: Mappatura analitica dell'RMS - 3D.

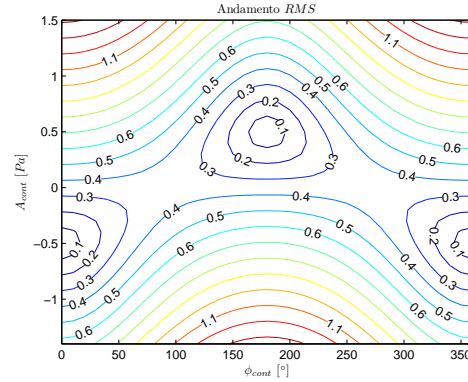


Figura 2.17: Mappatura analitica dell'RMS - isolinee.

Dall'andamento della funzione RMS mostrata in figura 2.16, si nota come questa funzione oltre alla zona in cui si ha effettivamente il minimo, presenta un altro punto a derivata nulla in corrispondenza di un valore nullo dell'ampiezza del segnale di controllo che comporta alcuni problemi nella determinazione del punto di minimo, infatti il *metodo del gradiente* non ha caratteristiche globali ma solo locali. Il rischio nell'utilizzo del metodo del gradiente è quello di trovare un punto che non è un minimo della funzione ma solo un punto in cui il gradiente è nullo. Questo problema è da tenere in considerazione qualora la funzione da minimizzare presenti punti di minimo locale o punti di sella. Risulta inoltre importante scegliere con la dovuta attenzione il punto iniziale nel quale andare a valutare il gradiente, questo permette al metodo di localizzare il vero punto di minimo della funzione.

Essendo l'RMS una funzione a due parametri, è necessario valutare il gradiente sia nella direzione dell'ampiezza che in quella della fase (Appendice A.1). Naturalmente, non essendo possibile in questo caso valutare analiticamente la funzione RMS, e pertanto il suo gradiente, è necessario calcolare il gradiente della funzione per mezzo delle seguenti formule alle differenze finite:



$$\left. \frac{\partial RMS}{\partial \phi_{cont}} \right|_{\bar{A}_{cont}, \bar{\phi}_{cont}} \cong \frac{RMS(\bar{A}_{cont}, \bar{\phi}_{cont} + \Delta \phi_{cont}) - RMS(\bar{A}_{cont}, \bar{\phi}_{cont})}{\Delta \phi_{cont}}, \quad (2.47)$$

$$\left. \frac{\partial RMS}{\partial A_{cont}} \right|_{\bar{A}_{cont}, \bar{\phi}_{cont}} \cong \frac{RMS(\bar{A}_{cont} + \Delta A_{cont}, \bar{\phi}_{cont}) - RMS(\bar{A}_{cont}, \bar{\phi}_{cont})}{\Delta A_{cont}}. \quad (2.48)$$

I gradienti sono stati calcolati in corrispondenza dei valori dei parametri di controllo  $\bar{A}_{cont}$  e  $\bar{\phi}_{cont}$ . Le grandezze  $\Delta A_{cont}$  e  $\Delta \phi_{cont}$  rappresentano gli incrementi da fornire alle variabili di controllo al fine di localizzare il minimo della funzione RMS.

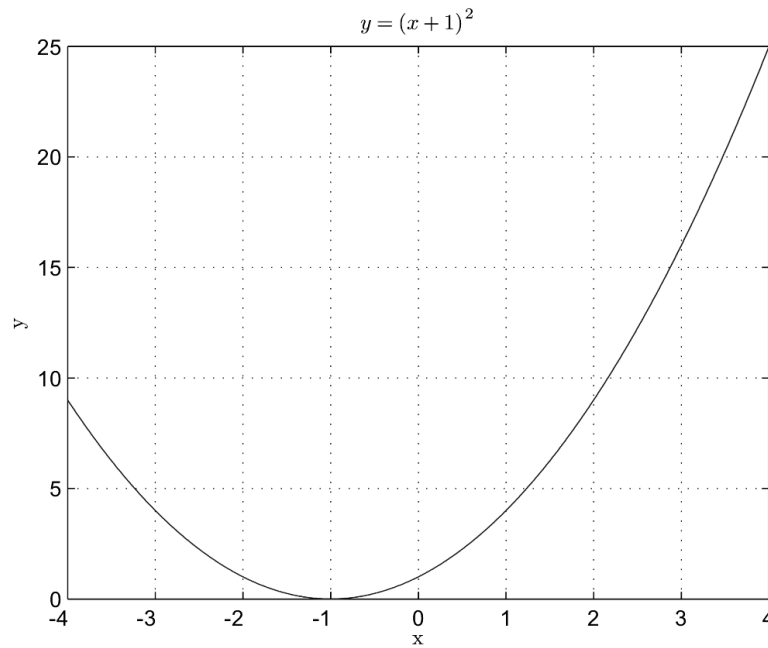


Figura 2.18: Funzione  $y = (x + 1)^2$ .

Si consideri a titolo di esempio la ricerca del minimo della seguente funzione ad un solo parametro: (figura 2.18):

$$y = f(x) = (x + 1)^2, \quad x \in (-\infty, +\infty). \quad (2.49)$$

Nota la funzione è possibile valutare analiticamente il suo gradiente, ad esempio nel punto  $\bar{x} = 4$ :

$$\nabla f(\bar{x}) = 2(\bar{x} + 1) = 10. \quad (2.50)$$

Ipotizzando di non conoscere analiticamente la funzione è possibile utilizzare, per il calcolo del gradiente, la seguente formulazione alle differenze finite:

$$\nabla f(\bar{x}) \cong \frac{f(\bar{x} + \Delta x) - f(\bar{x})}{\Delta x} = 10.1; \quad (2.51)$$

avendo fissando un incremento  $\Delta x = 0.1$ . È evidente che il valore che si ottiene dalla relazione 2.51 è un'approssimazione della 2.49, questa risulta tanto più accurata tanto più piccolo è il valore dell'incremento  $\Delta x$ , tuttavia l'incremento da dare alle variabili deve risultare apprezzabile rispetto al rumore, presente in un sistema reale, al fine di ottenere una valutazione corretta del gradiente. Nella realtà questi problemi possono riguardare la presenza di errori nella misurazione, l'utilizzo di variabili a virgola mobile, oppure a problemi legati all'utilizzo di buffer nella generazione del segnale, ecc. Tutti questi problemi limitano la soglia inferiore degli incrementi sulle variabili di controllo. A partire dalle condizioni iniziali  $A_{cont}^{k=0}$  e  $\phi_{cont}^{k=0}$  e muovendosi iterativamente lungo la direzione decrescente della funzione RMS si determina il punto di ottimo. I valori successivi di ampiezza e fase ottenuti dalla ricerca del minimo possono essere valutati, secondo quanto mostrato nell'appendice A.1, con le seguenti relazioni:

$$\phi_{cont}^{k+1} = \phi_{cont}^k - G_2^k \nabla RMS^\phi, \quad (2.52)$$

$$A_{cont}^{k+1} = A_{cont}^k - G_2^k \nabla RMS^A, \quad (2.53)$$

dove  $G_2^k$  è il guadagno che viene tarato ad ogni passo di iterazione per garantire la convergenza del metodo di ricerca del minimo. Questo parametro deve essere tale da garantire la convergenza dell'algoritmo e una adeguata velocità nel raggiungimento della convergenza, evitando il "saltellamento" attorno alla soluzione di minimo. A partire da un valore di guadagno iniziale  $G_1$  si calcola il valore del guadagno  $G_2$  applicando delle restrizioni sulla variazione delle grandezze e monitorando l'andamento dei gradienti in funzione del passo di iterazione. Per motivi di semplicità di notazione, i valori dei gradienti vengono da ora indicati con la seguente nomenclatura:

$$\nabla RMS^\phi = \left. \frac{\partial RMS}{\partial \phi_{cont}} \right|_{A_{cont}, \phi_{cont}}, \quad (2.54)$$

$$\nabla RMS^A = \left. \frac{\partial RMS}{\partial A_{cont}} \right|_{A_{cont}, \phi_{cont}}. \quad (2.55)$$

### 2.3.2 Tecnica di controllo AS

La tecnica di minimizzazione del rumore basata sul calcolo del valore quadratico medio del segnale rilevato da un microfono di errore posizionato all'uscita del condotto può risultare molto influenzata da fattori esterni, come ad esempio la presenza di rumore elettrico sulla misura, la presenza di armoniche non correlate al rumore generato dalla sorgente, oppure la presenza di eventuali rumori non stazionari e impulsivi provenienti dall'esterno che causano degli incrementi o delle variazioni temporanee del valore della funzione RMS. Le variazioni dell'RMS possono quindi determinare un errato calcolo del gradiente che comporta necessariamente un maggiore tempo per il raggiungimento del valore di soglia di rumore che si desidera ottenere. Al fine di ridurre i problemi legati all'utilizzo di un funzionale basato sul valore quadratico medio del rumore rilevato dal microfono è possibile utilizzare un metodo differente che isola, mediante un'analisi spettrale, la componente in frequenza di interesse che si intende abbattere. Questo metodo, definito per semplicità AS (Analisi Spettrale), consiste nell'effettuare un'analisi spettrale sul segnale acquisito dal microfono e, successivamente, valutare l'altezza del picco corrispondente alla frequenza del rumore che si desidera abbattere.

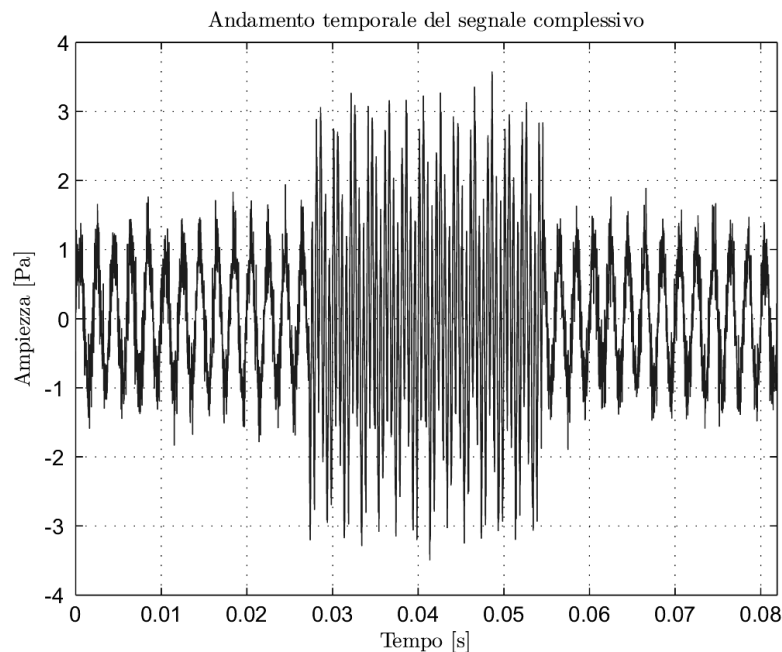


Figura 2.19: Segnale complessivo acquisito dal sistema.

Per mettere meglio in evidenza i problemi legati alla presenza dei fenomeni esposti in precedenza, si consideri ad esempio il segnale riportato in figura 2.19, esso è costituito dalla somma di tre differenti segnali (figura 2.20):

- Segnale principale: si suppone che il segnale generato dalla sorgente di rumore sia costituito da una sinusoide con frequenza di  $500\text{ Hz}$  e ampiezza pari ad  $1\text{ Pa}$ .
- Rumore bianco: per simulare una reale lettura di un microfono che risulta affetta dalla presenza di rumore elettrico, è stato aggiunto al segnale principale del rumore incorrelato (rumore bianco).
- Segnale secondario: può capitare che al vero segnale di rumore generato dalla sorgente (segnale principale), vada ad aggiungersi il segnale di una seconda sorgente. Questo può essere presente ad intervalli non regolari che potrebbe causare dei problemi non indifferenti sulla determinazione del valore quadratico medio.

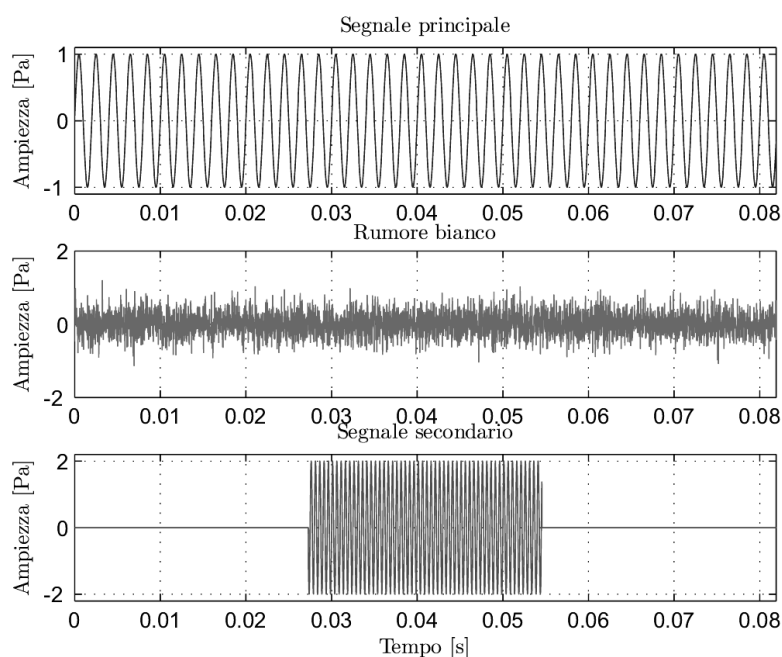


Figura 2.20: *Segnali costituenti il segnale complessivo.*

Analizzando la figura 2.21 è possibile pensare di sfruttare un'analisi spettrale del segnale letto dal microfono per ottenere una misura dell'ampiezza del vero segnale di rumore presente nel sistema; infatti per calcolare l'ampiezza del segnale di rumore è possibile andare a valutare il valore del picco presente nello spettro. Si vuole far notare

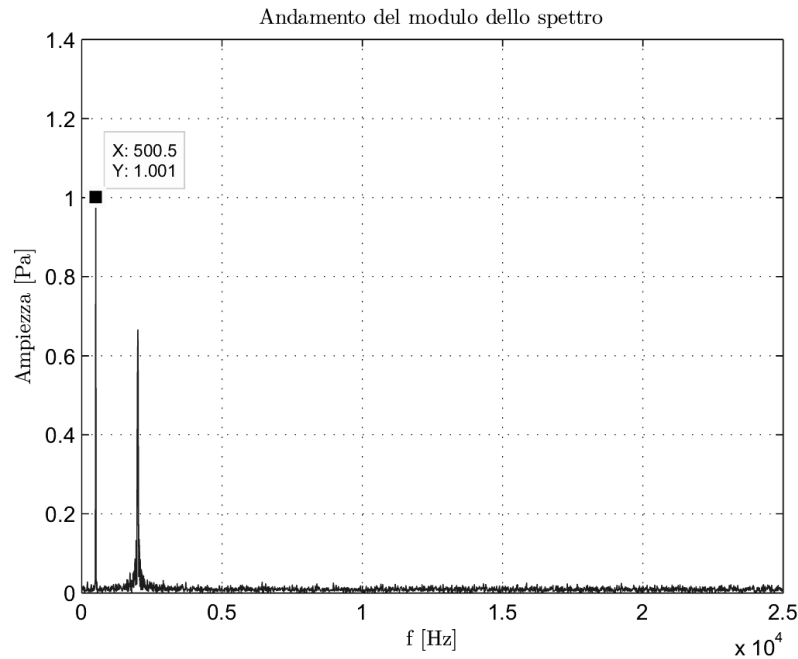


Figura 2.21: *Spettro del segnale complessivo.*

che questo picco ha praticamente lo stesso valore dell'ampiezza del segnale sinusoidale. Nella tabella 2.3 viene effettuato un confronto tra il metodo basato sul calcolo del valore quadratico medio e quello basato sull'analisi spettrale. In particolare si nota la forte discrepanza tra il valore quadratico medio che fornirebbe l'algoritmo di controllo nel caso in cui si considera il segnale completo, rispetto al valore quadratico medio che si otterrebbe sul segnale in assenza di disturbi esterni. Considerando invece la tecnica basata sull'analisi spettrale, si nota come questa non sia praticamente influenzata dalla presenza di possibili disturbi esterni, infatti i valori determinati con questa tecnica sono pressoché identici.

<b>Tecnica di controllo</b>	<b>Segnale principale</b>	<b>Segnale completo</b>
<b>RMS [Pa]</b>	0.707	1.134
<b>Ampiezza spettro [Pa]</b>	0.998	1.001

Tabella 2.3: *Confronto tra la misura dell'RMS e dell'ampiezza spettrale.*

Da queste osservazioni si vuole sfruttare la tecnica di minimizzazione col metodo del gradiente applicandola direttamente al valore del picco rilevato dallo spettro del segnale. Un'ulteriore prerogativa nell'utilizzo di un metodo basato sull'analisi spettrale è il fatto

che questo risulta completamente indifferente alla presenza di più armoniche nel segnale analizzato. Questa constatazione permette di applicare ricorsivamente questo metodo per abbattere il rumore alle diverse componenti spettrali che costituiscono il segnale di rumore, abbattendo quindi segnali di rumore multi-armonici.

Il processo per determinare il segnale da fornire alla cassa acustica di controllo al fine di minimizzare l'ampiezza del picco dello spettro del segnale è lo stesso utilizzato nella minimizzazione dell'RMS (par. 2.3.1), a patto di sostituire il gradiente della funzione RMS con il gradiente dell'ampiezza spettrale dell'armonica considerata. L'identificazione del picco dello spettro che si intende abbattere e la determinazione della sua ampiezza si basano su un'analisi spettrale del segnale utilizzando la trasformata di Fourier. La tecnica di calcolo della trasformata e i problemi relativi alla determinazione dello spettro vengono discussi nell'appendice A.2. Dato che la funzione obiettivo è differente da quella utilizzata nella tecnica di controllo precedente, è opportuno determinare una nuova mappatura del dominio di interesse, considerando l'andamento dell'ampiezza spettrale alla frequenza  $f_{rum} = f_{cont}$  in funzione delle grandezze del segnale di controllo.

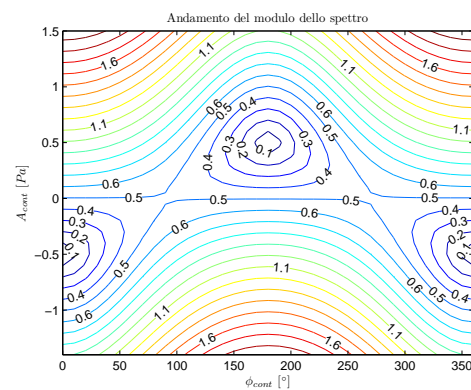
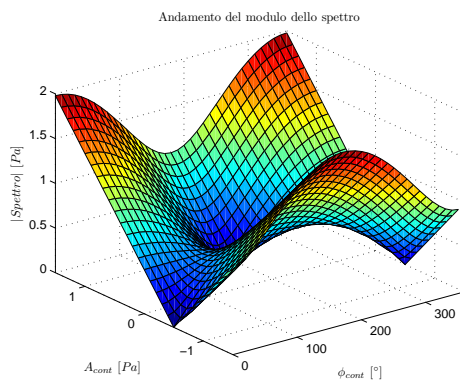


Figura 2.22: Mappatura analitica dell'ampiezza spettrale - 3D.

Figura 2.23: Mappatura analitica dell'ampiezza spettrale - isolinee.

Nella figura 2.22 si riporta l'andamento della somma del segnale di rumore generato dalla sorgente e del segnale di controllo, al variare dei parametri ampiezza e fase di quest'ultimo. L'andamento della funzione obiettivo è identico a quello della tecnica di controllo basata sulla minimizzazione dell'RMS, rendendo di fatto identico anche il processo di minimizzazione e di ricerca del minimo della funzione obiettivo. Le figure sono state ottenute utilizzando lo script MatLab riportato nell'appendice C.2.

Dal grafico si nota che, anche in questo caso, si hanno due minimi e un punto a

derivata nulla in corrispondenza di un valore nullo dell'ampiezza del segnale di controllo che comporta alcuni problemi nella determinazione del punto di minimo, di conseguenza è importante scegliere con la dovuta attenzione il punto iniziale nel quale andare a valutare il gradiente, questo permette al metodo di localizzare il vero punto di minimo della funzione.

I gradienti calcolati col metodo delle differenze finite e valutati al variare della fase e dell'ampiezza sono ottenuti dalle seguenti relazioni:

$$\left. \frac{\partial P_f}{\partial \phi_{cont}} \right|_{\bar{A}_{cont}, \bar{\phi}_{cont}} \cong \frac{P_f(\bar{A}_{cont}, \bar{\phi}_{cont} + \Delta \phi_{cont}) - P_f(\bar{A}_{cont}, \bar{\phi}_{cont})}{\Delta \phi_{cont}}, \quad (2.56)$$

$$\left. \frac{\partial P_f}{\partial A_{cont}} \right|_{\bar{A}_{cont}, \bar{\phi}_{cont}} \cong \frac{P_f(\bar{A}_{cont} + \Delta A_{cont}, \bar{\phi}_{cont}) - P_f(\bar{A}_{cont}, \bar{\phi}_{cont})}{\Delta A_{cont}}, \quad (2.57)$$

dove  $P_f$  è l'ampiezza del picco rilevato dallo spettro in corrispondenza della frequenza  $f$  di analisi, ossia della frequenza analizzata del segnale di rumore. Indicando, per semplicità di esposizione, i gradienti come:

$$\nabla P_f^\phi = \left. \frac{\partial P_f}{\partial \phi_{cont}} \right|_{\bar{A}_{cont}, \bar{\phi}_{cont}}, \quad (2.58)$$

$$\nabla P_f^A = \left. \frac{\partial P_f}{\partial A_{cont}} \right|_{\bar{A}_{cont}, \bar{\phi}_{cont}}, \quad (2.59)$$

è possibile definire i valori assunti dalle grandezze di controllo al passo successivo dell' algoritmo di controllo:

$$\phi_{cont}^{k+1} = \phi_{cont}^k - G_2^k \nabla P_f^\phi, \quad (2.60)$$

$$A_{cont}^{k+1} = A_{cont}^k - G_2^k \nabla P_f^A, \quad (2.61)$$

dove  $G_2^k$  è un opportuno guadagno da tarare in modo da garantire la convergenza del metodo del gradiente.

### 2.3.3 Tecnica di controllo MFS

Come affermato precedentemente uno dei principali vantaggi del metodo di minimizzazione del rumore basato sulla misurazione dell'ampiezza dell'armonica di interesse dello spettro, consiste nel poter utilizzare il metodo in modo ricorsivo al fine di ridurre le differenti armoniche che costituiscono il segnale di rumore. Questo non è possibile dalla misura dell'RMS poiché con questa funzione non si riescono a separare le armoniche

del rumore e non è quindi possibile analizzare in modo separato i contributi dati dalle singole frequenze.

Nello spettro del segnale, analizzato nell'esempio riportato nel paragrafo 2.3.2 alla figura 2.21, si nota la presenza di due righe spettrali, il che permette di giustificare un utilizzo ricorsivo del metodo con l'analisi spettrale, per minimizzare passo passo l'ampiezza del picco in corrispondenza delle diverse frequenze. Si tratta dunque di un processo di minimizzazione Multi-Frequenza con approccio Seriale, per semplicità denominato MFS. Le relazioni per il calcolo del gradiente mostrate nel paragrafo 2.3.2 sono le stesse e vengono applicate in modo ricorsivo. I gradienti vengono calcolati col metodo delle differenze finite e valutati al variare della fase e dell'ampiezza del segnale di controllo:

$$\left. \frac{\partial P_{f_i}}{\partial \phi_{cont}^i} \right|_{\bar{A}_{cont}^i, \bar{\phi}_{cont}^i} \cong \frac{P_{f_i}(\bar{A}_{cont}^i, \bar{\phi}_{cont}^i + \Delta \phi_{cont}^i) - P_{f_i}(\bar{A}_{cont}^i, \bar{\phi}_{cont}^i)}{\Delta \phi_{cont}^i}, \quad (2.62)$$

$$\left. \frac{\partial P_{f_i}}{\partial A_{cont}^i} \right|_{\bar{A}_{cont}^i, \bar{\phi}_{cont}^i} \cong \frac{P_{f_i}(\bar{A}_{cont}^i + \Delta A_{cont}^i, \bar{\phi}_{cont}^i) - P_{f_i}(\bar{A}_{cont}^i, \bar{\phi}_{cont}^i)}{\Delta A_{cont}^i}, \quad (2.63)$$

dove  $P_{f_i}$  è l'ampiezza del picco rilevato dallo spettro in corrispondenza dell' $i$ -esima frequenza  $f_i$  di analisi, ossia della frequenza analizzata del segnale di rumore. Indicando, per semplicità di esposizione, i gradienti come:

$$\nabla P_{f_i}^\phi = \left. \frac{\partial P_{f_i}}{\partial \phi_{cont}^i} \right|_{\bar{A}_{cont}^i, \bar{\phi}_{cont}^i}, \quad (2.64)$$

$$\nabla P_{f_i}^A = \left. \frac{\partial P_{f_i}}{\partial A_{cont}^i} \right|_{\bar{A}_{cont}^i, \bar{\phi}_{cont}^i}, \quad (2.65)$$

è possibile definire i valori assunti dalle grandezze di controllo al passo successivo dell'algoritmo di controllo:

$$\phi_{cont}^{(k+1),i} = \phi_{cont}^{k,i} - G_2^{k,i} \nabla P_{f_i}^\phi, \quad (2.66)$$

$$A_{cont}^{(k+1),i} = A_{cont}^{k,i} - G_2^{k,i} \nabla P_{f_i}^A, \quad (2.67)$$

dove  $G_2^{k,i}$  è un opportuno guadagno, che varia in funzione della frequenza  $i$ -esima di analisi, da tarare in modo da garantire la convergenza del metodo del gradiente.



### 2.3.4 Tecnica di controllo MFP

Ipotizzando una completa indipendenza nell'abbattimento delle singole armoniche che costituiscono il segnale di rumore è possibile pensare di effettuare il calcolo del gradiente contemporaneamente su tutte le frequenze di interesse. Tale approccio è dunque di tipo Multi-Frequenza con minimizzazione di tipo Parallelo, per semplicità denominato MFP. Questa ipotesi permette quindi di applicare il metodo precedente in modo simultaneo e non più consecutivo a tutte le frequenze delle armoniche contenute nel segnale di rumore con il conseguente vantaggio di un sostanziale abbattimento del tempo necessario all'algoritmo per terminare la minimizzazione, infatti in questo caso il tempo necessario  $T_{ALGO_P}$  è pari a quello impiegato per effettuare la minimizzazione della frequenza che richiede il maggior numero di iterazioni, secondo la seguente relazione:

$$T_{ALGO_P} = \max(T_{f_1}, T_{f_2}, \dots, T_{f_N}), \quad (2.68)$$

dove  $T_{f_i}$  indica il tempo necessario per minimizzare il rumore alla frequenza  $i$ -esima e  $N$  è il numero di armoniche che compongono il segnale di rumore. Questo tempo è sicuramente inferiore del tempo  $T_{ALGO_S}$  richiesto all'algoritmo che procede alla minimizzazione in modo consecutivo, dove questo è dato dalla sommatoria dei tempi richiesti algoritmo per la minimizzazione delle singole frequenze, secondo la seguente relazione:

$$T_{ALGO_S} = \sum_{i=1}^N T_{f_i} \gg T_{ALGO_P}. \quad (2.69)$$

Le formule alla base del metodo sono le stesse utilizzate per l'algoritmo MFS, pertanto valgono le relazioni mostrate nel paragrafo 2.3.3. La differenza rispetto all'algoritmo MFS è nella valutazione dei picchi spettrali: nell'algoritmo MFS viene effettuata valutando e abbattendo un picco per volta, mentre nell'algoritmo MFP vengono valutati ad ogni iterazione di controllo tutti i picchi spettrali di ampiezza maggiore ad una determinata soglia. Il calcolo del gradiente avviene quindi per tutti i picchi spettrali contemporaneamente e il controllo agisce simultaneamente su tutte le frequenze del segnale del rumore da controllare.



## Capitolo 3

# Analisi sperimentale degli algoritmi ANC

### 3.1 Apparato sperimentale

Nella sezione 2.1 si sono poste le basi teoriche per l'analisi della propagazione delle onde sonore lungo una direzione preferenziale e si è individuata un possibile ambiente sonoro che consentisse, mediante l'utilizzo di attuatori acustici comuni, l'ottenimento di questo tipo di propagazione. L'ambiente sonoro è un condotto di sezione quadrata, le cui dimensioni teoriche sono riassunte nella tabella 2.1. Per poter eseguire la verifica sperimentale degli algoritmi mostrati nel capitolo 2 si procede alla realizzazione fisica del condotto e dei sistemi di supporto per gli attuatori acustici. Nella figura 3.1 viene mostrato lo schema del sistema ANC che si intende realizzare. L'apparato sperimentale è costituito da un condotto a sezione quadrata, avente lunghezza longitudinale  $L_{tot} = 2 \text{ m}$  e lunghezza del lato interno della sezione pari a  $l_1 = 150 \text{ mm}$ . Esso viene realizzato in lastre di polimetilmetacrilato (PMMA) dello spessore di  $15 \text{ mm}$ ; questo materiale è stato scelto per la facilità della lavorazione. Esso viene poi fissato ad un supporto realizzato con barre d'alluminio, che consente di sospendere la base di appoggio del condotto.

La cassa acustica di controllo viene montata in corrispondenza della mezzeria del condotto, in modo che le onde sonore da lei generate risentano il meno possibile degli effetti di bordo. Il fissaggio della cassa acustica avviene per mezzo di una opportuna flangia di fissaggio, anch'essa realizzata in PMMA, che consente una adeguata flessibilità dell'apparato di test nel caso si voglia sostituire il tipo di cassa acustica utilizzata per il controllo.

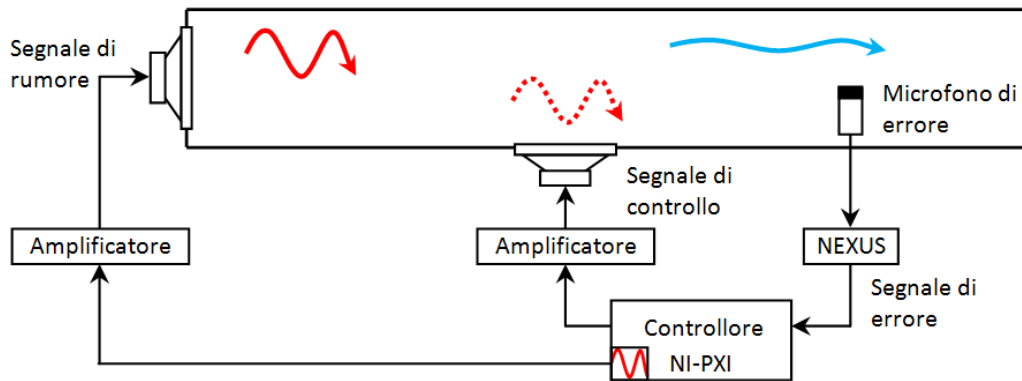


Figura 3.1: *Controllo attivo del rumore: schema del sistema ANC.*

Il microfono per la rilevazione del residuo di rumore presente all'uscita del condotto viene posto alla distanza di  $100\text{ mm}$  dall'uscita del condotto. Tuttavia vengono predisposte altre sedi di montaggio dei microfoni lungo tutta la lunghezza del condotto, ogni  $100\text{ mm}$ . Queste sedi possono servire per effettuare misure in parallelo a quelle rilevate dal sensore integrato nella catena di controllo. La cassa acustica per la generazione del rumore viene invece montata su un telaio in alluminio appositamente disegnato in modo tale che il centro del cono sia sempre posto in corrispondenza dell'asse principale del condotto. Tale supporto non è vincolato al condotto, pertanto è possibile variarne a piacere la posizione, secondo le considerazioni teoriche effettuate nella sezione 2.1. Anche in questo caso la cassa acustica è montata su una flangia in modo da poter consentire, qualora vi sia l'esigenza, la sostituzione della cassa acustica.

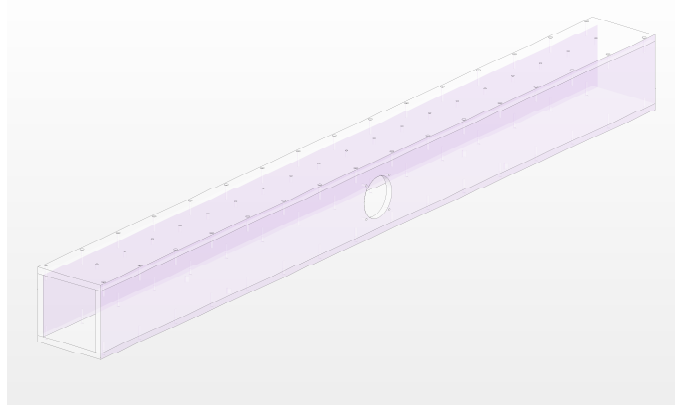


Figura 3.2: *Controllo attivo del rumore: modello 3D del condotto.*

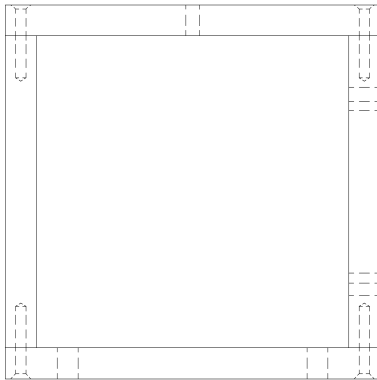


Figura 3.3: *Controllo attivo del rumore: sezione del condotto.*

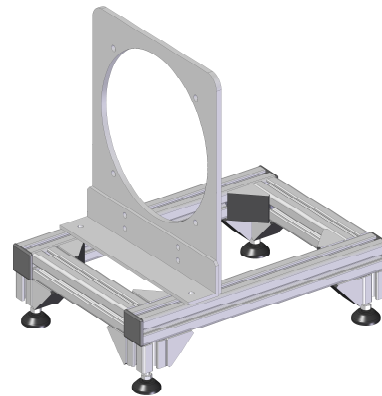


Figura 3.4: *Controllo attivo del rumore: modello 3D del supporto per le casse acustiche preposte alla generazione di rumore.*

Nelle figure 3.2, 3.3, 3.4 e 3.5 si riportano alcuni particolari dei modelli 3D realizzati per la costruzione effettiva del condotto. Nella figura 3.2 è mostrato il modello del condotto reale, in cui si può notare come sul lato superiore siano presenti 19 fori per l'alloggiamento dei microfoni di misura, e lateralmente in posizione centrale sia presente il foro per il montaggio della cassa acustica di controllo. Nella figura 3.3 è mostrata la sezione del condotto, che permette di capire la modalità di realizzazione e assemblaggio mediante viti. Nella figura 3.4 è mostrato il supporto per le casse acustiche realizzato mediante profilati in alluminio. Il supporto è costituito da una lamiera che funge da flangia per il montaggio delle casse acustiche. Nella figura 3.5 è mostrato il modello complessivo dell'apparato di test, in cui si può notare come il supporto della cassa per la generazione di rumore sia svincolato dal supporto del condotto. Questo permette di adeguare la distanza della cassa in funzione della tipologia di prova effettuata, in modo tale da garantire che all'interno del condotto vi sia una propagazione delle onde sonore di tipo monodimensionale.

Per una visione più dettagliata del progetto del condotto si rimanda ai relativi disegni tecnici. Nella figura 3.6 si mostra la realizzazione fisica del sistema. Nella figure 3.7 e 3.8 è mostrato il particolare del supporto della cassa acustica.

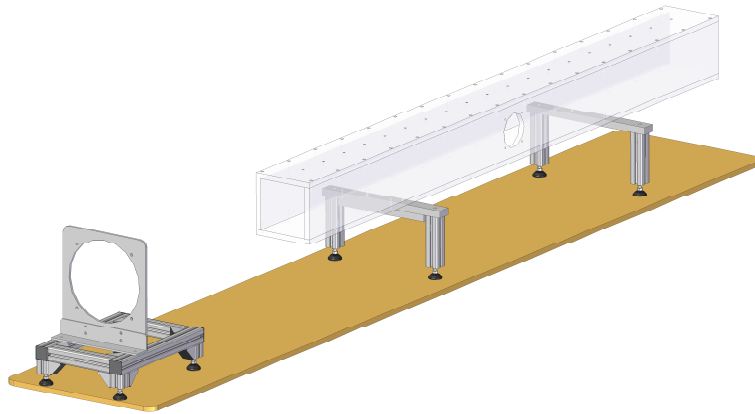


Figura 3.5: *Controllo attivo del rumore: modello 3D dell'apparato di test.*

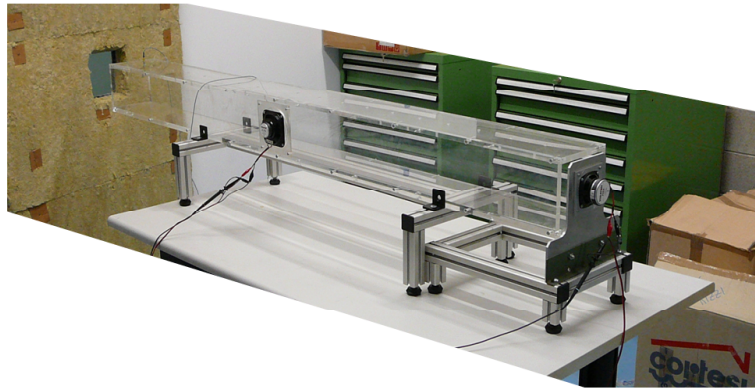


Figura 3.6: *Controllo attivo del rumore: apparato sperimentale di test.*

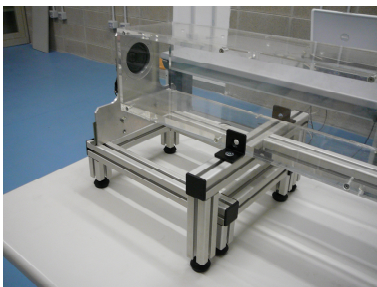


Figura 3.7: *Controllo attivo del rumore: supporto per la cassa di generazione del rumore.*

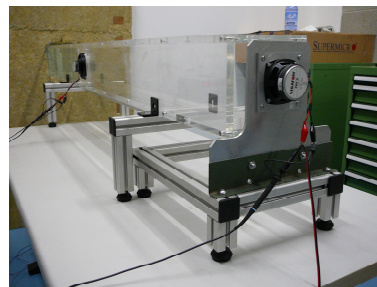


Figura 3.8: *Controllo attivo del rumore: vista posteriore del sistema reale.*

## 3.2 Sensori, attuatori e hardware per il controllo attivo

Avendo dimensionato e realizzato opportunamente il condotto, occorre definire fisicamente i componenti del sistema di controllo, come i sensori per la rilevazione del rumore, gli attuatori acustici, l'hardware e il software di controllo. In questa sezione vengono mostrati i criteri di scelta e le principali caratteristiche tecniche di tali componenti, in accordo alle considerazioni effettuate nella sezione 2.2.

### 3.2.1 Scelta di sensori e attuatori

La scelta dei microfoni viene effettuata secondo le considerazioni tecniche legate alla risposta in frequenza del sensore, al tipo di campo sonoro che si intende misurare e ai limiti dell'intervallo dinamico del sensore, oltre che alla risposta in fase e al tipo di polarizzazione del sensore. Per lo studio si sono utilizzati due microfoni prodotti da Bruel & Kjaer. Entrambi i microfoni sono del modello 4944; si tratta di microfoni pre-polarizzati, il cui impiego risulta più semplice poiché non è richiesta ad ogni utilizzo la polarizzazione del microfono. I parametri principali dei microfoni sono riportati nell'appendice B.3. Per l'utilizzo di questi microfoni con un qualunque sistema di acquisizione e/o controllo è necessario un amplificatore e un condizionatore del segnale letto dal microfono. A questo scopo è stato inserito nella catena di misura anche un amplificatore NEXUS per microfoni DeltaTron fornito da Bruel & Kjaer, per le cui caratteristiche si può far riferimento all'appendice B.2.

La scelta degli attuatori viene effettuata, riferendosi a componenti facilmente reperibili in commercio<sup>1</sup>. La cassa acustica scelta per la generazione del rumore è la seguente:

- Full-range “Visaton R 10 S 8 Ohm”:
  - Potenza nominale: 20 W,
  - Potenza picco: 30 W,
  - Impedenza: 8  $\Omega$ ,
  - Range frequenze: 100 Hz - 13 kHz,
  - $f_c$  (frequenza di risonanza): 160 Hz,
  - SPL: 90 dB.

---

<sup>1</sup>Le casse acustiche sono state scelte tra quelle disponibili dal fornitore RS electronics.

Questa cassa acustica è stata scelta in modo da poter generare agevolmente tutta la gamma di frequenze del campo udibile. Sebbene essa non offra prestazioni elevate rispetto a casse acustiche che coprono solo una parte del campo udibile, come woofer, middle range e tweeter, essa consente di avere comunque una buona qualità del rumore generato. La scelta è stata basata sulla ricerca di un attuatore che, all'interno del campo di frequenza di utilizzo, presentasse una funzione di trasferimento con un livello di pressione sonora il più possibile costante. A questo scopo si possono vedere le scheda tecnica della cassa acustica all'appendice B.4.

Per la scelta della cassa acustica di controllo anche in questo caso ci si è focalizzati sulla scelta di una cassa acustica per medie frequenze, che consente di effettuare il controllo in un ampio campo di frequenze. Le caratteristiche di questa cassa acustica sono le seguenti:

- Full-range “Visaton FRS 8 8 Ohm”
  - Potenza nominale: 30  $W$  (6000  $Hz$ ),
  - Potenza picco: 50  $W$  (6000  $Hz$ ),
  - Impedenza: 8  $\Omega$ ,
  - Range frequenze: 100  $Hz$  - 20  $kHz$ ,
  - $f_c$  (frequenza di risonanza): 130  $Hz$ ,
  - SPL: 82  $dB$ .

Anche in questo caso la scelta è stata dettata dalla volontà di disporre di una funzione di trasferimento più costante possibile nella gamma di frequenze di utilizzo della cassa.

### 3.2.2 Hardware e software utilizzato

Per il test degli algoritmi di controllo si è scelto di utilizzare un hardware prodotto da National Instruments. Esso consente di essere configurato in modo modulare con le schede di input/output di interesse per l'applicazione specifica. In particolare si è scelto l'utilizzo della versione NI PXI con tre differenti moduli:

- Modulo controllore: costituito dalla scheda NI PXI-8106, un controller embedded a prestazioni avanzate basato su un processore Intel Core 2 Duo. Questa scheda è il target del sistema di controllo real-time e su questo è stato installato il sistema operativo real-time (RTOS) LabView Real-Time Module.



- Modulo di acquisizione: per l'acquisizione del segnale del microfono di errore viene utilizzata la scheda di acquisizione NI PXI-6123, costituita da 8 ingressi analogici a 16 bit con campionamento simultaneo. Il campionamento simultaneo è reso possibile grazie all'utilizzo di un convertitore ADC (Analog to Digital Converter) su ciascun canale della scheda. Questo tipo di campionamento diviene molto utile nel caso di controllo multi-input in cui si rende necessario nel sistema di controllo utilizzare più sensori, infatti disponendo del campionamento simultaneo, si ha una maggiore frequenza di campionamento aggregata rispetto alle tradizionali schede con un solo convertitore per tutti i canali di acquisizione. Per il tipo di algoritmi che si intende implementare questa caratteristica non risulta necessaria, tuttavia si è scelta questa scheda essendo già disponibile presso il laboratorio. La frequenza di acquisizione massima della scheda è pari a  $500\text{ kHz}$  a canale permettendo quindi di effettuare tutte le misure di interesse per il campo acustico in cui viene utilizzato l'algoritmo di controllo.
- Modulo di generazione: per la generazione dei segnali analogici da fornire alla cassa di controllo e di rumore si utilizza la scheda NI PXI-6733, costituita da 8 canali di output analogico a 16 bit con aggiornamento simultaneo. La frequenza massima di output della scheda è pari a  $740\text{ kHz}$  a canale permettendo quindi di comandare la cassa di controllo nel range di frequenze di interesse.

Per le caratteristiche dettagliate dell'hardware si rimanda all'appendice B.1. La scrittura degli algoritmi di controllo è stata effettuata nell'ambiente di programmazione ANSI C, mediante il software LabWindows CVI fornito dal costruttore dell'hardware. Il codice implementato viene compilato in una libreria dinamica (DLL) e inserita all'interno di un loop deterministico creato nell'ambiente LabVIEW. L'utilizzo di questa procedura di programmazione ha consentito di svincolarsi dall'ambiente LabVIEW per quanto riguarda lo sviluppo del codice, creare codici personalizzati di qualunque complessità in ANSI C e snellire lo schema a blocchi su LabVIEW, permettendo quindi di sfruttare al meglio le prestazioni offerte dall'hardware. Il linguaggio di programmazione grafica di LabVIEW permette quindi di sviluppare le applicazioni di controllo embedded LabVIEW Real-Time su un comune computer, mediante l'utilizzo di DLL create con LabWindows CVI, e quindi scaricarle ed eseguirle sul target hardware NI PXI-8106.

I canali di acquisizione della scheda NI PXI-6123 non sono provvisti di filtri analogici pertanto non consentono di effettuare il condizionamento del segnale atto ad evitare il problema dell'aliasing; a tal fine si rende necessario l'utilizzo di un condizionatore/am-

plificatore esterno in grado di alimentare i sensori di misura, amplificare il segnale ed effettuare un filtraggio passa-basso alla frequenza di taglio di interesse per l'algoritmo di controllo. L'amplificatore utilizzato è il *NEXUS Conditioning Amplifier 2693-A-0S4*, le cui caratteristiche sono riportate nell'appendice B.2.

Le prestazioni di abbattimento vengono valutate utilizzando la piattaforma hardware e software *Bruel & Kjaer PULSE 3560-B*, che consente di effettuare tutte le misure di interesse, in particolare tale strumentazione viene utilizzata per rilevare il segnale di un secondo microfono posizionato a valle del microfono di errore. Questa strumentazione, con ingressi analogici a 24 bit con frequenza di campionamento fino a  $460\text{ kHz}$ , è stata inoltre utilizzata per determinare le funzioni di trasferimento del sistema condotto e degli attuatori utilizzati nel sistema di controllo. Le caratteristiche principali di questo hardware vengono riassunte nell'appendice B.1.

Per fornire all'utente uno strumento utile al controllo del sistema, al reporting e all'analisi dei dati, alla rilevazione e visualizzazione di possibili problemi nell'hardware e nel software di controllo è stata creata una semplice interfaccia in ambiente LabView (fig. 3.9).



Figura 3.9: *Interfaccia utente, GUI - host.*

Questa interfaccia è realizzata per mezzo dello schema a blocchi riportato in figura 3.10, dove viene effettuata la comunicazione delle variabili tra l'ambiente grafico dell'interfaccia utente, gestita da un comune PC con sistema operativo Windows denominato *host*, e il sistema real-time (*target*), costituito dal controllore NI PXI, mostrato in figura 3.11. La visualizzazione dei processi e del carico della CPU necessari all'esecuzione del sistema di controllo vengono visualizzati a video grazie ad un monitor collegato direttamente al sistema real-time. L'host è in continua comunicazione ad intervalli di tempo prefissati con il target real-time per mezzo di una rete LAN; il target invia e riceve dati dall'host che provvede alla visualizzazione grafica mediante l'interfaccia utente e al salvataggio dei dati di monitoraggio dell'algoritmo su file.

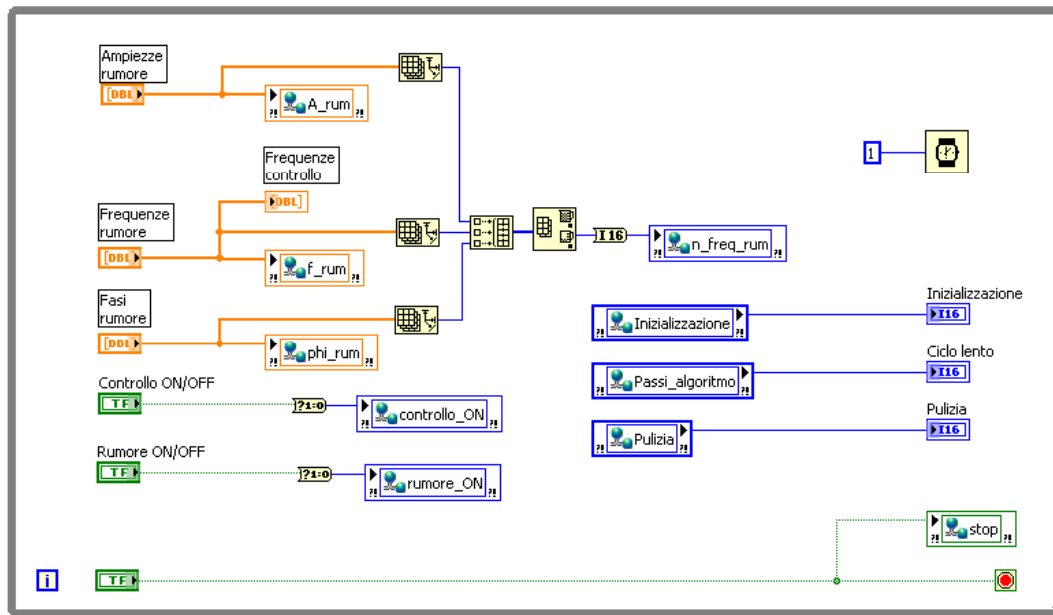


Figura 3.10: *Interfaccia utente, schema a blocchi - host.*

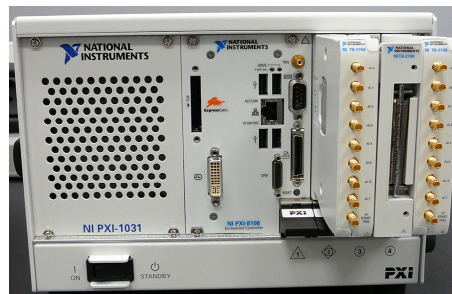


Figura 3.11: *Controllore NI PXI.*

Il compito del target è quello di eseguire il codice di controllo creato senza provvedere ad alcuna visualizzazione in tempo reale dei risultati. Un'eventuale visualizzazione o un uso improprio del sistema potrebbe causare la perdita del determinismo temporale e quindi la perdita delle prestazioni tipiche del sistema real-time. Per rendere minimo il carico di CPU occupato dall'algoritmo di controllo, per sfruttare al meglio le prestazioni dell'hardware a disposizione e semplificare eventuali operazioni matematiche la cui implementazione potrebbe risultare molto complessa in LabVIEW, si è scelto di non utilizzare l'ambiente LabVIEW per la realizzazione dei codici ma ci si è avvalsi, come accennato precedentemente, dell'ambiente di programmazione ANSI C che permette di realizzare codici anche di notevole complessità. Con questa metodologia si

sono inoltre realizzate una serie di funzioni per la gestione delle schede di cui è fornito il sistema PXI. Queste funzioni, che verranno mostrate meglio in seguito, comprendono ad esempio l'impostazione delle caratteristiche dell'hardware quale ad esempio il tempo di aggiornamento delle uscite, le funzioni per la creazione dei task, o la creazione e l'assegnamento ai task dei canali analogici e digitali, infine è anche possibile gestire la partenza, la pulizia e la terminazione di un task. Il codice C viene importato all'interno dell'ambiente LabVIEW sotto forma di una libreria dinamica (*DLL*).

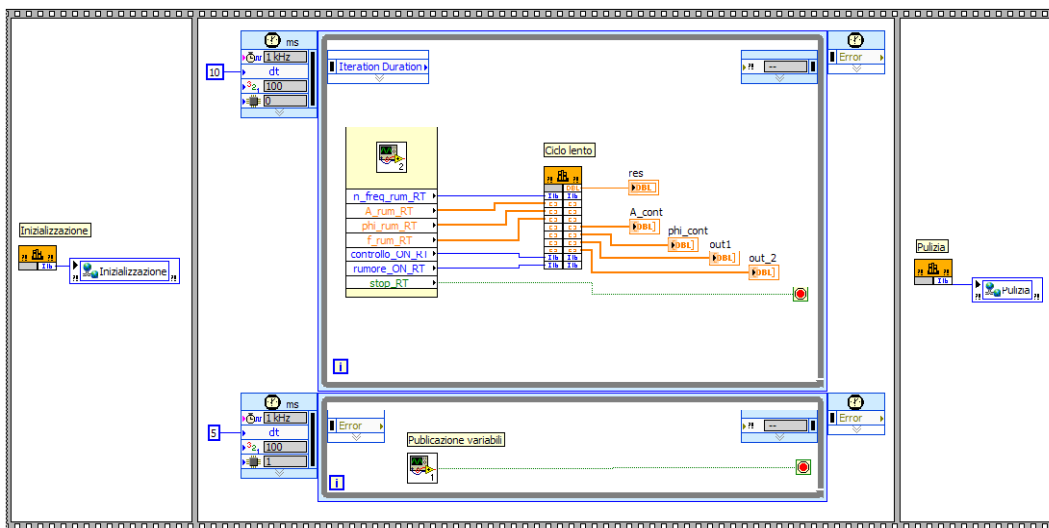


Figura 3.12: Schema a blocchi del sistema ANC - target.

Nella figura 3.12 viene visualizzato lo schema a blocchi che viene eseguito dal target. All'interno dello schema si nota la presenza della libreria dinamica che contiene al suo interno tre funzioni che vengono chiamate singolarmente in LabVIEW a seconda delle esigenze: una prima funzione permette l'inizializzazione del sistema di controllo, un'altra contiene al suo interno il cuore del controllo, ossia l'algoritmo e, infine, una funzione che provvede alla pulizia e all'eliminazione dei task. Lo schema è costituito principalmente da tre blocchi che riflettono le funzioni di cui è costituita la *DLL* creata, disposti secondo una struttura sequenziale<sup>2</sup>:

- Nel primo frame viene realizzata l'inizializzazione del sistema. In particolare il blocchetto arancione denominato con "Inizializzazione" effettua l'importazione

<sup>2</sup>In LabVIEW una struttura sequenziale è denotata dal comando mostrato in figura 3.12, che racchiude le operazioni che devono essere svolte in sequenza. A seconda dell'utilizzo è possibile creare differenti *frame* che vengono eseguiti in modo sequenziale da sinistra a destra. La partenza del frame successivo è legata al termine delle operazioni realizzate nel frame precedente.

della *DLL* creata e richiama la funzione di inizializzazione del sistema di controllo implementata in codice C. Successivamente, nello sviluppo degli algoritmi di controllo, verranno approfondite le funzioni contenute all'interno della *DLL* creata.

- Il secondo frame è costituito da due cicli *while* “Timed Loop”: il ciclo inferiore si occupa unicamente della comunicazione delle variabili tra il target real-time e l'host. Il ciclo superiore contiene il nocciolo del sistema di controllo rappresentato dal blocchetto arancione denotato con “Ciclo Lento”: questo blocco provvede all'importazione della *DLL* all'interno di LabView e richiama la funzione implementata nel codice C che realizza l'algoritmo di controllo. In questa funzione vengono implementati gli schemi a blocchi che verranno presentati in seguito durante la presentazione degli algoritmi (figure 3.49, 3.59, 3.68 e 3.76). È importante sottolineare che l'esecuzione dei due cicli appena descritti è demandata ai due differenti processori di cui è dotato il PXI: alla prima CPU del sistema Dual Core è affidata l'esecuzione del ciclo *while* per l'effettuazione del “Ciclo lento”, mentre il ciclo che si occupa della comunicazione tra il target e l'host è allocato sulla seconda CPU. L'utilizzo di entrambi i processori permette di tenere separati dei compiti che possiedono priorità differenti. La comunicazione tra target e host richiede infatti un intervallo temporale molto più ampio rispetto a quello per l'aggiornamento del ciclo *while* che contiene l'algoritmo di controllo.
- Nel terzo frame viene importata la *DLL* e richiamata la funzione per la pulizia dei task creati nel primo frame. Questa funzione di pulizia è necessaria al successivo riutilizzo del codice di controllo dopo un eventuale arresto dello stesso. La presenza di tale funzione assicura che il sistema operativo non subisca blocchi improvvisi, pulisce ed elimina i task predisponendo il sistema al suo prossimo utilizzo.

### 3.2.3 Progetto di un amplificatore

La scelta degli altoparlanti per la generazione del rumore e per il controllo dello stesso all'interno del condotto è stata effettuata esclusivamente guardando quelle che sono le caratteristiche degli attuatori in termini di gamma di frequenze sonore producibili e di funzione di trasferimento. Sapendo che tali attuatori devono essere comandati dal sistema di controllo rappresentato dal controllore PXI, ed in particolare dalla scheda

di output analogici NI 6733, è necessario progettare un circuito di amplificazione che disaccoppi idealmente i due circuiti elettrici.

Prima di procedere al dimensionamento dell'amplificatore si riportano le caratteristiche della scheda NI 6733 che comanda le casse acustiche:

$$V_{NIout,max} = \pm 10 \text{ V}, \quad (3.1)$$

$$I_{NIout,max} = 5 \text{ mA}. \quad (3.2)$$

Data la bassa corrente (massima) disponibile alle uscite della scheda, risulta necessaria l'interposizione di un circuito di amplificazione che fornisca adeguati valori di corrente per il pilotaggio del carico. Volendo progettare un amplificatore le cui caratteristiche siano adattabili a tutti gli altoparlanti scelti, si sceglie come carico su cui dimensionare l'amplificatore la cassa di controllo “*Visaton FRS 8 8 Ohm*” perchè è l'altoparlante in grado di assorbire la maggiore potenza. Le caratteristiche di tale altoparlante, rilevanti per la progettazione dell'amplificatore sono le seguenti:

$$P_{load,RMS} = 30 \text{ W}, \quad (3.3)$$

$$P_{load,max} = 50 \text{ W}, \quad (3.4)$$

$$Z_{load} = 8 \Omega. \quad (3.5)$$

Dove  $P_{load,RMS}$  è la potenza massima assorbibile dall'attuatore in condizioni di regime,  $P_{load,max}$  è la potenza di picco sopportabile dal dispositivo e  $Z_{load}$  è l'impedenza dell'altoparlante. Tutti gli altoparlanti sono stati scelti con la medesima impedenza, in modo tale che questi possano venire collegati al medesimo amplificatore senza particolari accorgimenti. Dai dati sull'altoparlante si ricavano i valori quadratici medi di tensione e corrente ( $V_{load,RMS}$ ,  $I_{load,RMS}$ ) e i valori di picco ( $V_{load,max}$ ,  $I_{load,max}$ ), sostenibili dall'apparecchiatura:

$$V_{load,RMS} = \sqrt{Z_{load}P_{load,RMS}} = 15.5 \text{ V}, \quad (3.6)$$

$$V_{load,max} = \sqrt{Z_{load}P_{load,max}} = 20.0 \text{ V}, \quad (3.7)$$

$$I_{load,RMS} = \sqrt{\frac{P_{load,RMS}}{Z_{load}}} = 1.9 \text{ A}, \quad (3.8)$$

$$I_{load,max} = \sqrt{\frac{P_{load,max}}{Z_{load}}} = 2.5 \text{ A}. \quad (3.9)$$

Per la realizzazione fisica dell'amplificatore, tra i componenti presenti sul mercato si utilizza l'amplificatore operazionale di potenza LM3875 prodotto da National Semiconductors, che consente di erogare una potenza continuativa di 56W e una potenza di picco pari a 100W. Le caratteristiche fisiche del componente sono riportate nel rispettivo datasheet [16]. La scelta di questo componente rispetto ad altri aventi le stesse caratteristiche elettriche è data dalla sua facile reperibilità presso distributori di materiale elettronico. Lo schema di amplificazione in cui si intende realizzare l'amplificatore è di tipo non invertente, nelle figure 3.13 e 3.14 si riportano lo schema elettrico ed il circuito equivalente.

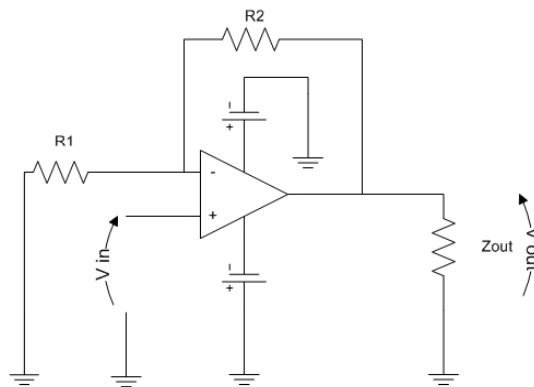


Figura 3.13: Schema elettrico di un amplificatore non invertente.

Il guadagno dell'amplificatore  $A_v$  viene definito come il rapporto tra la tensione di uscita e quella di ingresso:

$$A_v = \frac{V_{out}}{V_{in}}. \quad (3.10)$$

Se si applicano le leggi di Kirchhoff ai nodi e alle maglie del circuito equivalente si ottiene la seguente relazione:

$$\frac{V_{in}}{R_1} + \frac{V_{in} - V_{out}}{R_2} = 0, \quad (3.11)$$

dalla quale si può ricavare il guadagno dell'amplificatore:

$$V_{out} = \frac{R_1 + R_2}{R_1} V_{in}, \quad (3.12)$$

$$A_v = \frac{V_{out}}{V_{in}} = 1 + \frac{R_2}{R_1}. \quad (3.13)$$

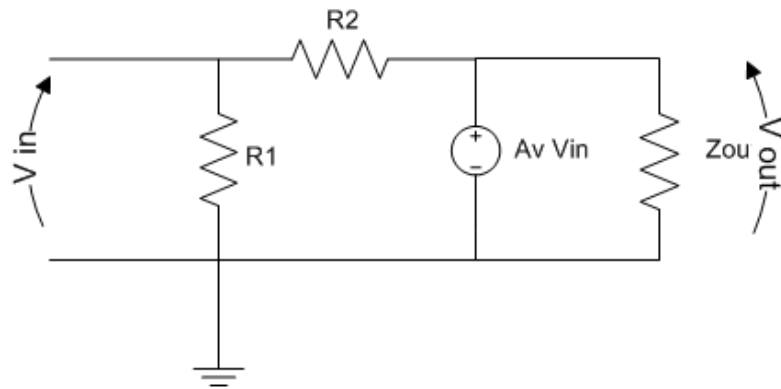


Figura 3.14: Circuito equivalente di un amplificatore non invertente.

Per la scelta del valore di guadagno da assegnare all'amplificatore, si cerca di poter sfruttare tutta la potenza erogabile dal dispositivo a semiconduttore, in modo tale da poter sfruttare il più possibile gli altoparlanti incrementando il livello di pressione sonora emessa da essi (SPL). Questo porterebbe alla scelta di un guadagno  $A_v$  elevato, il limite a questa scelta deriva dalla limitazione imposta sulla massima corrente erogabile in uscita dall'amplificatore. Con lo scopo di avere una corrente in uscita dall'amplificatore non superiore a  $I_{load,RMSmax} = 2 \text{ A}$ , si pone il guadagno dell'amplificatore pari a  $A_v = 2$ , e quindi il rapporto delle resistenze viene assunto unitario. I valori delle resistenze scelte sono pari a:

$$R_1 = R_2 = 1 \text{ k}\Omega. \quad (3.14)$$

In questo modo la corrente in uscita dal circuito amplificatore, che attraversa il carico è la seguente:

$$I_{load} = \frac{V_{out}}{Z_{load}} = \frac{A_v V_{in}}{Z_{load}}. \quad (3.15)$$

Dato che la massima tensione in ingresso al dispositivo è pari a quella massima erogabile dalla scheda NI 6733,  $V_{in,max} = V_{NIout,max} = 10 \text{ V}$ , si ha dunque una corrente massima pari a:

$$I_{load,max} = 2.5 \text{ A}, \quad (3.16)$$

e assumendo di pilotare l'altoparlante con un segnale sinusoidale, il valore quadratico medio della corrente risulta  $I_{load,RMSmax} = 1.768 \text{ A}$  entro il limite progettuale. I calcoli sono stati verificati con l'utilizzo del software di simulazione *NI Circuit Design Suite 10*, dal quale si ottengono i risultati riportati nella figura 3.15.



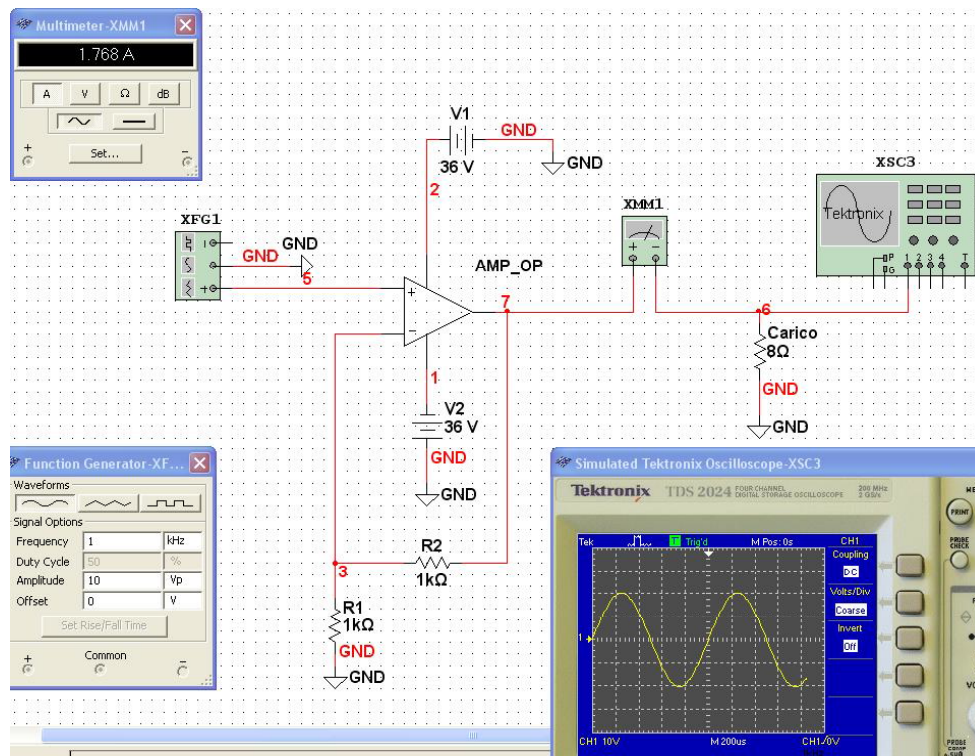


Figura 3.15: Simulazione dell'amplificatore mediante NI Circuit Design 10.

Accanto al dimensionamento di massima dell'amplificatore si è operato per la riduzione del rumore elettrico inserendo due condensatori che collegano gli ingressi dell'alimentazione dell'amplificatore operazionale con il riferimento di massa. L'inserimento di questi componenti (mostrati nello schema in figura 3.16) ha lo scopo di filtrare le possibili componenti in frequenza dovute ai disturbi sull'alimentazione. I condensatori scelti sono di tipo elettrolitico, con le seguenti capacità:

$$C_s = 100 \mu F. \quad (3.17)$$

Questi condensatori possono essere impiegati con tensioni fino a 50 V, ampiamente entro il limite di impiego del dispositivo. Infatti il segnale di ingresso può assumere valori massimi pari a 10 V, avendo un guadagno pari a  $A_v = 2$  la tensione di alimentazione può essere fissata pari a 20 V.

L'eventuale componente continua del segnale in ingresso, dannosa per gli attuatori, viene annullata via software mediante un'apposita programmazione del sistema di controllo.

Dall'impiego del circuito di amplificazione fin qui descritto, si è notata la presenza sul carico di disturbi con componenti ad elevata frequenza, pertanto è stato posto in serie all'uscita dell'amplificatore un filtro passa basso del primo ordine, costituito da un circuito RC il cui schema completo è riportato nella figura 3.16.

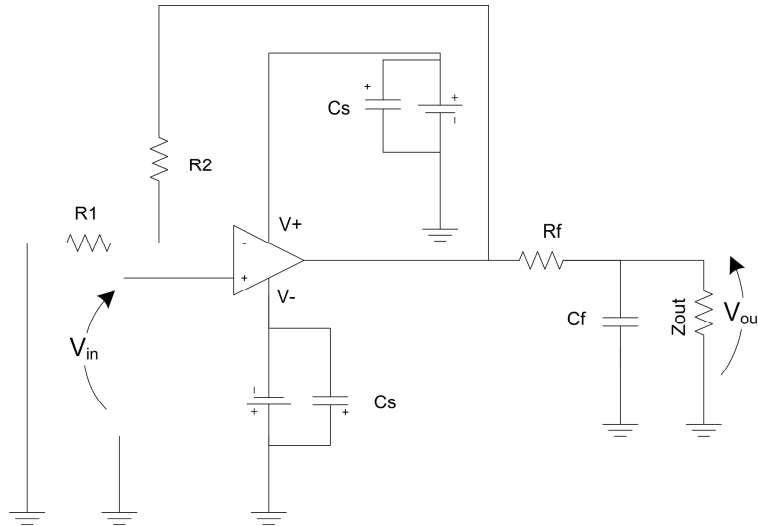


Figura 3.16: Schema elettrico dell'amplificatore audio.

L'impedenza del circuito costituito dalla resistenza  $R_f$  e dalla capacità  $C_f$  è pari a

$$Z(s) = R_f + \frac{1}{sC_f}. \quad (3.18)$$

La funzione  $Z(s)$  ha un polo per  $s = \frac{1}{\tau}$  dove

$$\tau = R_f C_f \quad (3.19)$$

è la costante di tempo caratteristica del circuito. Fissando la frequenza di taglio

$$f_{cut*} = 1000 \text{ Hz}, \quad (3.20)$$

alla quale si ha un'attenuazione del segnale in ingresso pari a 3 dB, si ottiene un valore della costante  $\tau$  pari a:

$$\tau = \frac{1}{2\pi f_{cut*}} = 1.59 \cdot 10^{-4} \text{ s}. \quad (3.21)$$

Scegliendo un valore di resistenza disponibile commercialmente pari a  $R_f = 4.7 \Omega$  ed un valore di capacità pari a  $C_f = 33 \mu F$  si ottiene un valore della frequenza di taglio pari a:

$$f_{cut} = \frac{1}{2\pi R_f C_f} = 1261 \text{ Hz}. \quad (3.22)$$

Nella figura 3.17 si riporta l'andamento teorico ( $FdT_{calc}$ ) e misurato ( $FdT_{ril}$ ) della funzione di trasferimento tra ingresso e uscita dell'amplificatore. L'andamento teorico della funzione di trasferimento è dato dalla funzione di trasferimento mostrata nell'equazione 3.23, di cui si riportano modulo e fase nelle equazioni 3.24 e 3.25:

$$FdT_{calc}(s) = \frac{2}{1 + 15.51s}; \quad (3.23)$$

$$|FdT_{calc}(f)| = \frac{2}{\sqrt{1 + 2\pi R_f C_f f}}; \quad (3.24)$$

$$\angle FdT_{calc}(f) = -\arctan(2\pi f R_f C_f). \quad (3.25)$$

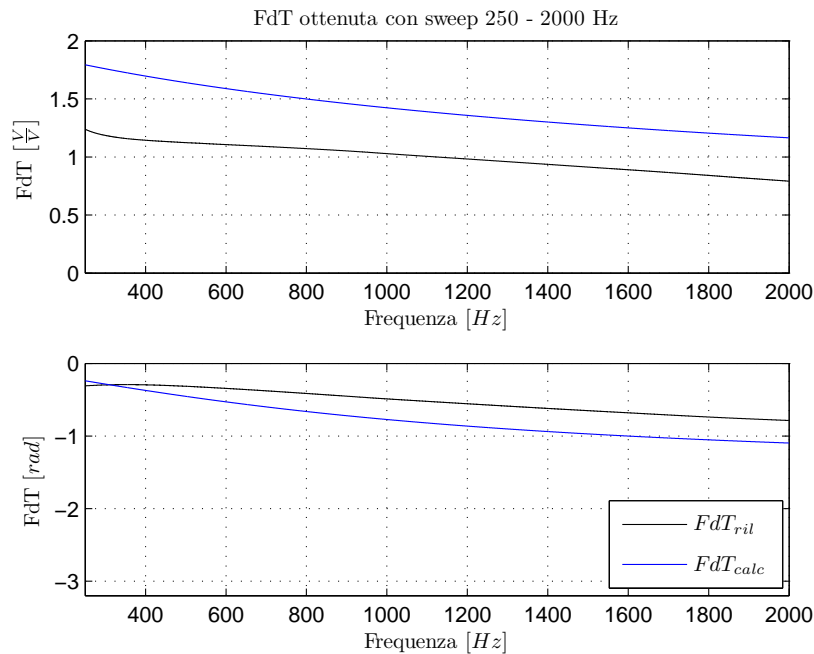


Figura 3.17: Funzione di trasferimento teorica e sperimentale tra ingresso e uscita dell'amplificatore.

Nella figura 3.18 si mostra la realizzazione del prototipo dell'amplificatore.

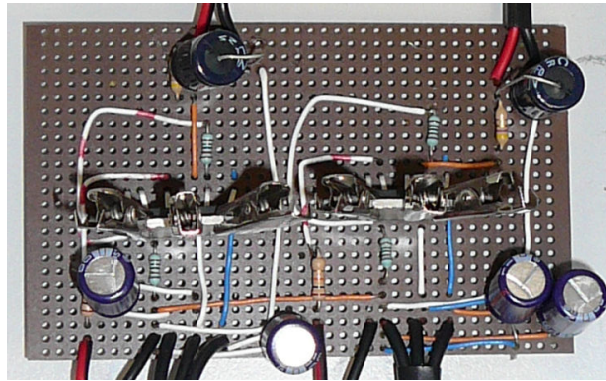


Figura 3.18: *Prototipo dell'amplificatore audio.*

### 3.2.4 Caratterizzazione sperimentale degli attuatori

La caratterizzazione degli altoparlanti è stata effettuata con lo scopo di utilizzare questi dati all'interno del sistema di controllo, consentendo di avere una catena di misura e attuazione a guadagno costante per tutte le frequenze di interesse. A partire dalla stima della funzione di trasferimento fra la tensione fornita all'altoparlante per mezzo dell'amplificatore e la risultante pressione sonora misurata attraverso un microfono opportunamente posizionato, mediante una opportuna fase di post processing dei dati ottenuti si è potuto stimare l'andamento della risposta dell'attuatore in funzione del segnale presente ai suoi ingressi. Lo scopo è quello di poter importare la funzione ottenuta, all'interno dell'algoritmo di controllo.

Sono state effettuate tre prove sperimentali, riassunte qui di seguito:

- Valutazione della funzione di trasferimento per punti, utilizzando segnali sinusoidali a specifiche frequenze e tenendo il microfono di misura ad una distanza fissa dalla cassa da caratterizzare.
- Valutazione della funzione di trasferimento utilizzando, come segnale di comando, uno sweep in frequenza. Anche in questa prova si è tenuto il microfono di misura ad una distanza fissa dalla cassa da caratterizzare.
- Valutazione dell'andamento dell'ampiezza e della fase della funzione di trasferimento in funzione della distanza tra la cassa da caratterizzare e il microfono di misura.

Lo schema della catena di misura è mostrato in figura 3.19 ed è composta dai seguenti elementi:

- Bruel & Kjaer PULSE 3560B.
- Amplificatore audio.
- Altoparlante da caratterizzare.
- Microfono di misura.

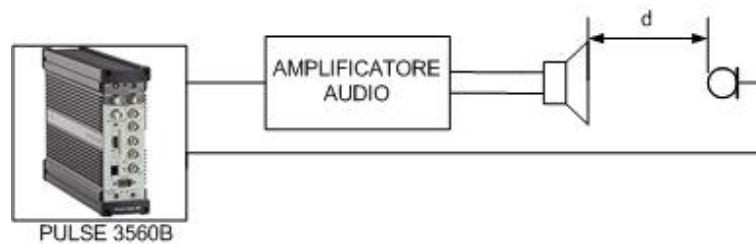


Figura 3.19: Schema della catena di misura per la caratterizzazione delle casse acustiche.

Le prove vengono eseguite sia senza l'amplificatore audio inserito nella linea, sia successivamente con l'amplificatore audio. Le prove vengono effettuate misurando la risposta dell'altoparlante in campo aperto, in modo da risentire il meno possibile della presenza di fenomeni di rifrazione e riflessione. Prima di procedere all'esecuzione delle prove è opportuno fissare l'intervallo in frequenza in cui devono essere effettuate le prove. Tale intervallo dipende dal tipo di altoparlante utilizzato e dall'impiego che ne verrà fatto, in particolare, per evitare il danneggiamento degli altoparlanti, gli estremi del range di frequenze vengono fissati in funzione dei dati riportati nelle schede tecniche degli stessi. Nel seguito vengono presentate le varie tecniche utilizzate per eseguire la caratterizzazione e i risultati ottenuti dalle prove sperimentali. L'obiettivo principale della caratterizzazione è quello di determinare l'andamento della funzione di trasferimento in termini di modulo e fase, e di verificare la dipendenza delle prove effettuate dai parametri sperimentali come ad esempio la distanza del microfono di misura dall'altoparlante da caratterizzare.

A seguito delle prove effettuate per la caratterizzazione degli altoparlanti, è stata anche eseguita una misura dell'influenza dell'amplificatore audio realizzato appositamente per questo sistema. Questa misura ha permesso di confermare la bontà realizzativa del circuito di amplificazione, quando esso è connesso al carico.

### Funzione di trasferimento delle casse acustiche

Il valore della funzione di trasferimento può essere valutato eseguendo differenti prove in cui il sistema viene eccitato con una forzante sinusoidale a frequenza fissa e viene valutata la risposta a regime. Il vantaggio principale legato a questa metodologia è essenzialmente la facilità nel controllo della forzante, che consente di misurare delle risposte in campo lineare e a regime. Questo risulta più complicato nel caso in cui la forzante utilizzata per la caratterizzazione è un segnale armonico a frequenza variabile (sweep), qui infatti può risultare più complicato garantire delle risposte a regime dell'attuatore. Lo svantaggio nel determinare la funzione di trasferimento per punti è legato invece all'elevato tempo di esecuzione della prova.

Per l'esecuzione della prova ci si avvale del software di analisi PULSE 7770 fornito da Bruel & Kjaer, tuttavia gli stessi risultati possono essere ottenuti mediante un qualsiasi software di calcolo, come ad esempio MatLab. Per effettuare l'analisi in frequenza occorre innanzitutto disporre di un segnale con cui eccitare il sistema, come ad esempio un segnale sinusoidale a frequenza fissa o uno sweep in frequenza. Di questi si devono stabilire i parametri principali per la sua generazione ovvero, l'ampiezza e la risoluzione del segnale. L'ampiezza del segnale è determinata in relazione alla potenza massima che può essere fornita dall'altoparlante. La risoluzione del segnale è invece determinata unicamente dalla risoluzione del convertitore DAC del sistema di generazione. All'aumentare del numero di bit del generatore aumenta la risoluzione del segnale generato secondo la nota legge:

$$\delta V = \frac{FS}{2^N} = 4.17 \cdot 10^{-7} V, \quad (3.26)$$

dove  $FS = 7 V$  è il fondo scala del generatore PULSE 3560B, e  $N = 24$  è il numero di bit del convertitore.

Una volta fissato il segnale da generare, è possibile configurare i parametri per effettuare l'analisi spettrale del segnale acquisito. In particolare sono stati impostati i seguenti parametri:

- banda di analisi;
- numero di linee;
- dominio in cui effettuare la media;
- tipologia di media;

- tipo di finestra di acquisizione;
- sovrapposizione dei record temporali.

La banda di analisi determina il range di frequenze in cui effettuare l'analisi spettrale. Ci sono due tipologie principali di banda di analisi: la banda base che permette di effettuare le analisi dalla frequenza di 0  $Hz$  alla frequenza superiore definita dall'utente, e l'analisi in banda che permette di fissare la frequenza centrale della banda e il range di frequenze di interesse della banda stessa. In quest'ultimo caso, tutte le frequenze vengono moltiplicate per  $e^{j2\pi f_b t}$  dove  $f_b$  è la frequenza centrale della banda e  $t$  rappresenta il tempo. Questa operazione trasla lo spettro di un valore pari a  $f_b$  (ad esempio una componente spettrale a  $f_b$  viene traslata a 0  $Hz$  [19]). I campioni sono trattati con un filtro passa basso opportunamente tarato. Questa metodologia di analisi è utile per studiare in dettaglio una parte di spettro delle frequenze o per ridurre il fenomeno del leakage nelle funzioni di risposta in frequenza. Poiché lo scopo delle prove è quello di ricavare l'andamento della funzione di trasferimento in un determinato intervallo di frequenze comprendente anche le basse frequenze, si è scelto di utilizzare l'analisi in banda base.

La scelta della banda di analisi determina anche la frequenza di campionamento del segnale misurato dal microfono. Tale frequenza è fissata in base al teorema del campionamento di Shannon ed è pari a:

$$f_c = 2f_{bh}, \quad (3.27)$$

dove  $f_{bh}$  è la frequenza più elevata che si vuole osservare. Tuttavia, data la presenza del filtro anti-aliasing, è necessario campionare ad una frequenza leggermente più elevata di quella calcolata con il teorema di Shannon:

$$f_c = 2.56f_{bh}. \quad (3.28)$$

La banda di analisi scelta è da 250  $Hz$  a 2500  $Hz$ , pertanto la si sceglie una banda di acquisizione maggiore e pari a 3200  $Hz$ ; la frequenza di acquisizione sarà  $f_c = 8192$   $Hz$ .

Il numero di linee ( $N_{linee}$ ) determina il numero di campioni acquisiti nel dominio del tempo su cui effettuare la trasformata discreta di Fourier (DFT). A parità di frequenza di campionamento  $f_c$ , aumentando il numero di campioni acquisiti aumenta la risoluzione spettrale  $\Delta f$ . Le grandezze citate sono legate tra loro dalle seguenti relazioni:

$$\Delta f = \frac{f_c}{N_{linee}}, \quad (3.29)$$

$$T_{acq} = \frac{1}{\Delta f} = \frac{N_{linee}}{f_c}. \quad (3.30)$$

Il numero di linee pregiudica la velocità di calcolo della trasformata di Fourier, infatti, se il numero di linee viene scelto pari ad una potenza di 2, per il calcolo della DFT viene utilizzato l'algoritmo di Cooley-Tukey che permette di ridurre notevolmente il numero delle operazioni per la determinazione dello spettro [17]. I fattori che limitano l'ampiezza della banda di frequenze da analizzare possono essere i seguenti:

- la massima frequenza analizzabile dal sistema di acquisizione utilizzato;
- la potenza di calcolo del sistema utilizzato.

Il numero di linee scelto è pari a 6400.

I record temporali acquisiti vengono mediati tra loro con lo scopo di ridurre l'apporto del rumore incorrelato e garantire una misurazione spettrale più corretta. È possibile eseguire la media dei segnali sia nel dominio del tempo che nel dominio delle frequenze. Effettuando la media dei segnali nel dominio delle frequenze, le funzioni di auto-spettro e cross-spettro sono funzioni mediate, mentre effettuando la media nel dominio del tempo i record temporali vengono mediati tra loro prima di effettuare l'analisi in frequenza. Per effettuare la funzione di trasferimento per punti è possibile adottare una media dei segnali nel dominio delle frequenze. Effettuare una media nel dominio delle frequenze significa mediare tra loro punto a punto i valori degli spettri dei segnali dei record temporali acquisiti, ad esempio secondo la seguente formula (media lineare):

$$\bar{X}(f_k) = \frac{1}{N_{record}} \sum_{i=1}^{N_{record}} X(f_k). \quad (3.31)$$

Considerando, come segnale che deve essere mediato, lo spettro in forma di numero complesso, risultato della trasformata di Fourier discreta, si ottiene una riduzione del rumore incorrelato; tuttavia occorre selezionare i record temporali in modo tale che essi siano acquisiti tutti con la stessa fase, per le stesse motivazioni espresse nella media nel dominio nel tempo. Per garantire la corretta acquisizione dei segnali occorre effettuare un'acquisizione a trigger, sincronizzato con il segnale da acquisire, oppure acquisire un unico record temporale e successivamente suddividerlo in sotto-record in base alla periodicità del segnale; se non si conosce la periodicità è possibile valutarla con la



funzione di autocorrelazione [18]. Con questo tipo di media il rapporto tra segnale e rumore migliora di una quantità pari a  $10\log(N_{record})$ .

È possibile effettuare tre tipologie di media:

- Esponenziale: questa media pone l'enfasi sull'ultimo spettro o sull'ultimo record temporale (a seconda del dominio utilizzato per effettuare la media). Il contributo di ciascuno spettro o record temporale è pesato utilizzando una funzione esponenziale. La formula su cui ci si basa è la seguente:

$$EMA_i(x_i, N) = SMA_i(x_i, N), \quad i = N, \quad (3.32)$$

$$EMA_i(x_i, N) = EMA_{i-1} + \frac{2}{N+1}(x(t_i) - EMA_{i-1}), \quad (3.33)$$

dove  $EMA_i$  è il risultato della media esponenziale tra il record temporale  $x_i$  di durata  $T$  e gli  $N$  record temporali precedenti, mentre  $SMA$  è il risultato della media mobile su  $N$  record; per la prima operazione di media si considera una media lineare. Se i record temporali sono stati acquisiti senza sovrapposizione, il prodotto  $\frac{N \cdot T}{2}$  è uguale ad una costante di tempo. Questo metodo è utilizzato principalmente per l'osservazione di fenomeni transitori.

- Lineare: viene eseguita la media senza pesare in maniera differente gli spettri o i record temporali. Questo metodo è utilizzato principalmente per analisi su segnali stazionari.
- Per picchi: con questo metodo vengono memorizzate, per ciascuna frequenza, le ampiezze più elevate dei vari spettri. Questa metodologia è utile quando si ha un segnale in transitorio e si vuole tenere traccia delle massime ampiezze raggiunte per ciascuna frequenza.

Per ciascuna metodologia presentata un parametro di fondamentale importanza è il numero di spettri o di record temporali su cui effettuare la media. Esso deve essere scelto in funzione del tipo di segnale da analizzare. Se nel segnale è presente molto rumore incorrelato, sarebbe opportuno effettuare il processo di media su un numero elevato di record temporali. Per segnali stazionari è possibile effettuare le medie solo su poche storie temporali, specialmente se un record temporale contiene molti periodi del segnale analizzato.

Per la valutazione della funzione di trasferimento per punti, si è scelto di adottare una mediazione di tipo esponenziale con un numero di record pari a 10, in modo da depurare il risultato dal transitorio iniziale causato dall'applicazione della forzante sinusoidale all'altoparlante.

L'operazione che consente il calcolo dello spettro da un segnale acquisito nel dominio del tempo, ossia la trasformata di Fourier discreta, si basa sull'ipotesi che la ripetizione circolare del record temporale acquisito possa ricostruire perfettamente il segnale. Questa ipotesi può essere soddisfatta solo se nel record temporale sono presenti un numero intero di periodi del segnale in esame. Nella realtà questa ipotesi difficilmente può essere soddisfatta, si presenta quindi un troncamento del segnale in corrispondenza della fine del record temporale, che causa una discontinuità fittizia nella ripetizione circolare adottata dalla trasformata di Fourier discreta. Questo fenomeno prende il nome di “*errore di leakage*” ed è causato dalla convoluzione nel dominio delle frequenze che, nel dominio del tempo, corrisponde alla moltiplicazione tra il segnale campionato e la finestra temporale utilizzata.

Per ridurre questo errore una prima tecnica consiste nell'aumentare il tempo totale di acquisizione del segnale, a parità di frequenza di campionamento. In questo modo si aumenta la risoluzione in frequenza ed il contenuto energetico del segnale viene suddiviso sempre nello stesso numero di linee, che però risultano più vicine tra loro, consentendo di identificare meglio le frequenze di interesse.

La seconda tecnica utilizzata consiste nell'abbattere la discontinuità agli estremi del record temporale, ovvero cambiare il tipo di finestra di acquisizione. La scelta della finestra più conveniente dovrà, pertanto, essere operata secondo un compromesso tra il valore dell'accuratezza delle misure di ampiezza ed il livello di selettività richiesto. Tra le varie finestre ponderali, viene utilizzata la finestra di Hanning la cui formulazione<sup>3</sup> è rappresentata dall'equazione 3.34:

$$W(n) = 0.5 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right). \quad (3.34)$$

Le finestre di Hanning, presenta una migliore selettività rispetto alla non applicazione di una finestra di ponderazione, un ripple contenuto ed un minor leakage.

Un altro parametro su cui è bene porre l'attenzione è la percentuale di sovrapposizione dei record temporali. Tale percentuale è determinata dal rapporto tra la lunghezza del record temporale ed il tempo necessario al processamento dei dati raccolti.

<sup>3</sup>La finestra è definita per un record di  $N$  campioni, con  $n = 0, 1, 2, 3, \dots, N-1$ .

I dati acquisiti vengono pesati con una finestrazione di Hanning per evitare i problemi di leakage, tuttavia l'utilizzo di questa finestra causa una perdita dei dati dovuta al ridotto peso con cui vengono moltiplicati i campioni del record acquisito agli estremi della finestra (infatti agli estremi la finestra assume valore nullo e va crescendo fino al massimo collocato nel mezzo della stessa); se si vuole ridurre la perdita dei dati tra un campione e il successivo è necessario adottare una sovrapposizione tra i campioni, la quale tanto più è elevata tanto minore è la quantità dei dati che vengono persi nell'analisi. Per evitare la perdita dei dati può essere utilizzata una finestra rettangolare e un overlap nullo tra le storie, tuttavia questo comporta i problemi di leakage a causa delle discontinuità presenti agli estremi della finestra. Si è pertanto scelto di utilizzare la massima sovrapposizione possibile e l'utilizzo di una finestrazione di Hanning, in modo da ridurre la perdita di dati durante l'elaborazione di un record temporale e ridurre i problemi di leakage.

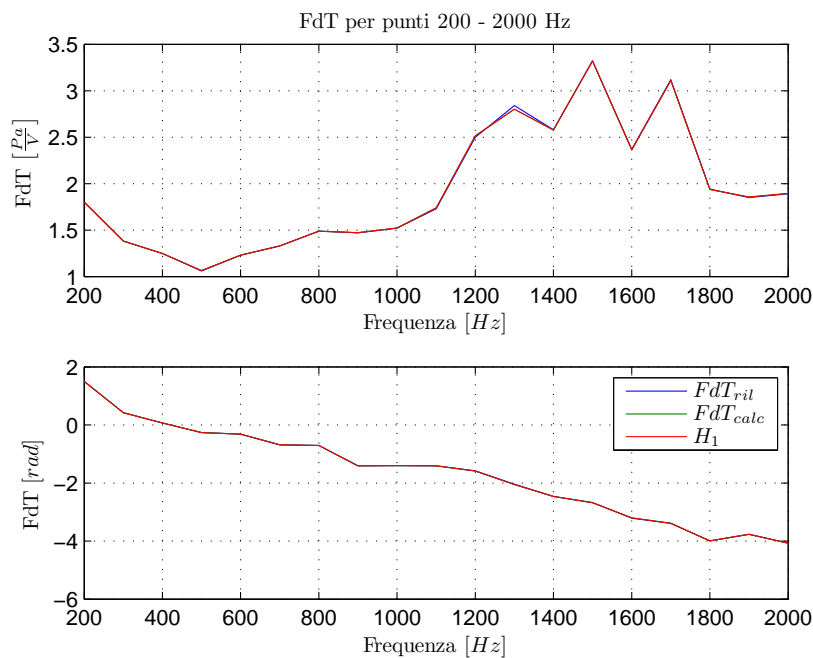


Figura 3.20: Funzione di trasferimento della cassa di controllo eseguita con prove a frequenza costante.

Nel caso in esame, dovendo stimare la funzione di trasferimento dell'altoparlante, il segnale in ingresso viene generato direttamente dall'analizzatore PULSE 3560B, che esegue anche la misura degli ingressi. Si può quindi affermare che il sistema non risen-

te di rumore sulla misura degli ingressi e che lo stimatore<sup>4</sup> più corretto risulta essere  $H_1(f)$ . In figura 3.20 si riportano i risultati delle analisi condotte sulla caratterizzazione dell'altoparlante di controllo; in particolare, con la curva denominata  $FDT_{ril}$  si mostrano i risultati ottenuti con il software di analisi PULSE e basati sulla stima ottenuta con  $H_1$ , con la curva  $FDT_{calc}$  i risultati ottenuti dall'analisi delle storie temporali e utilizzando la metodologia esposta nelle equazioni A.21 e A.22, e con la curva  $H_1$  i risultati ottenuti dall'analisi delle storie temporali e utilizzando lo stimatore  $H_1(f)$ . Come si può notare i risultati sono concordanti.

A scopo di verifica delle ipotesi effettuate prima dell'esecuzione della prova, sono state effettuate delle stime della funzione di trasferimento utilizzando anche gli altri stimatori, presentati nell'appendice A.4 nelle equazioni A.27 e A.29: si è notato che il rumore sul segnale di uscita è comunque minimo e praticamente non influenza la correttezza della misura.

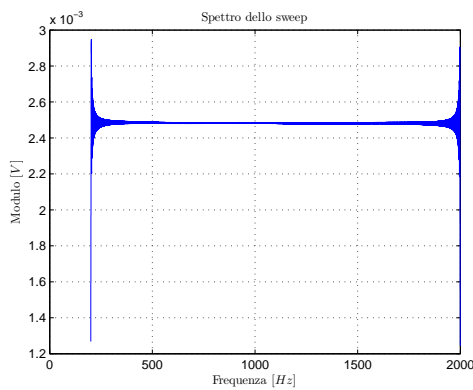


Figura 3.21: Spettro del segnale in ingresso, costituito da uno sweep in frequenza.

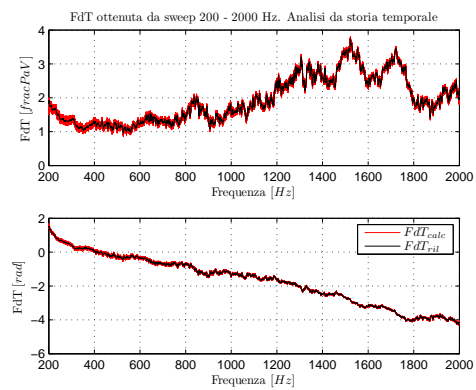


Figura 3.22: Funzione di trasferimento dell'altoparlante di controllo, ottenuta mediante prova con sweep in frequenza.

Per effettuare l'identificazione della funzione di trasferimento si può anche utilizzare una forzante a frequenza variabile con continuità da una frequenza minima ad una frequenza massima fissate. Le misure effettuate con questo metodo saranno dunque misure di risposta al transitorio. Occorre scegliere quindi una velocità di variazione della frequenza della forzante in modo da ottenere una risposta di tipo quasi stazionario. La velocità può essere lineare (espressa in  $[Hz/s]$ ) oppure logaritmica (espressa in  $[dec/s]$ ) e va scelta in funzione del sistema da analizzare e, in particolare, in funzione della

<sup>4</sup>Per una panoramica degli stimatori utilizzati si veda l'appendice A.4.

sua dinamica. Una bassa velocità di variazione consente di ottenere risposte quasi stazionarie, ma aumenta il tempo di esecuzione della campagna sperimentale.

Fornendo all'attuatore, come segnale di comando, uno sweep in frequenza, e calcolando lo spettro del segnale ottenuto nel dominio del tempo (risposta dell'attuatore alla forzante in ingresso), è possibile verificare che vi è effettivamente lo stesso contributo, in termini di energia fornita, per tutte le frequenze. Nella figura 3.21 viene mostrato lo spettro di uno sweep in frequenza da  $200 \text{ Hz}$  a  $2000 \text{ Hz}$ . Come si può notare l'energia fornita è pressoché la medesima per tutte le frequenze, tranne per le frequenze vicine agli estremi dello sweep. Per ridurre questo effetto di ripple, è possibile effettuare più sweep concatenati, effettuando salita e discesa in modo alternato, così da ridurre le discontinuità.

Per la prova di caratterizzazione si è scelta una velocità di variazione della frequenza della forzante pari a  $10 \frac{\text{Hz}}{\text{s}}$  in modo da ridurre gli errori di stima della funzione dovuti al transitorio. Poiché lo spettro del segnale di uscita varia continuamente, per stimare correttamente la funzione di trasferimento si scelgono i seguenti parametri:

- Banda di analisi: pari a quella utilizzata per l'esecuzione della funzione di trasferimento per punti e contenente il campo di frequenze generate dallo sweep in frequenza.
- Numero di linee: per avere una buona risoluzione in frequenza si è scelto un numero elevato di linee, pari a 6400. Poiché il segnale è in continuo transitorio, la sua variazione deve essere minima all'interno del record temporale acquisito.
- Dominio in cui effettuare la media: si effettua la media nel dominio delle frequenze per gli stessi motivi esposti precedentemente.
- Tipo di media: viene effettuata una media per picchi, che consente di individuare per ogni frequenza il valore di picco degli spettri. Questo consentirà, durante il calcolo della funzione di trasferimento, di considerare solo le righe spettrali che di volta in volta sono soggette all'eccitazione.
- Tipo di finestra di acquisizione: Viene selezionata una finestra di ponderazione di tipo Hanning, dalle considerazioni esposte precedentemente.
- Sovrapposizione dei record temporali: si effettuano delle misurazioni con la massima sovrapposizione dei record temporali, in modo da evitare la perdita di campioni acquisiti.

Per effettuare il calcolo della funzione di trasferimento si utilizza anche in questo caso lo stimatore  $H_1(f)$ . Nella figura 3.22 si riportano i risultati ottenuti dalle analisi svolte sul sistema applicando una forzante di tipo sweep: con la curva identificata con  $FdT_{ril}$  si riportano i risultati ottenuti dal software di analisi PULSE, con la curva identificata con  $FdT_{calc}$  si riportano i risultati ottenuti analizzando l'intera storia temporale dei segnali ed utilizzando la metodologia esposta nell'appendice A.4 nelle equazioni A.21 e A.22. Nella figura 3.23 vengono confrontati i risultati ottenuti con la misurazione effettuata per punti e la misurazione effettuata mediante lo sweep in frequenza. Dall'osservazione di detta figura si nota come i risultati siano ampiamente confrontabili, a riprova del fatto di avere effettuato un'accurata scelta dei parametri con cui viene eseguita la misura della funzione di trasferimento mediante sweep in frequenza.

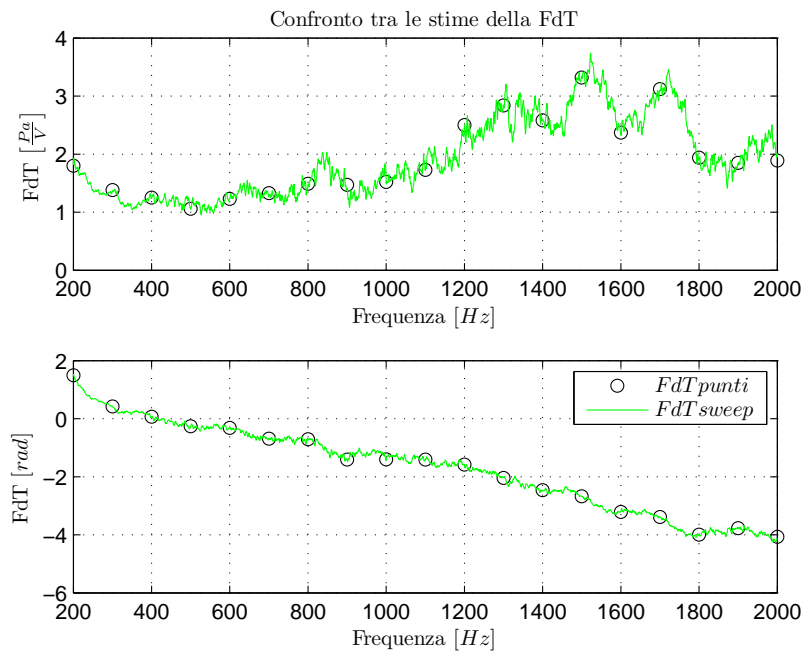


Figura 3.23: Confronto tra i risultati ottenuti dalle due metodologie di misura.

### Influenza dei parametri sulla determinazione della FdT

Con lo scopo di verificare l'influenza nella stima della funzione di trasferimento della distanza del microfono dalla cassa da caratterizzare, si è proceduto ad effettuare una serie di prove sperimentali determinando la funzione di trasferimento a frequenza fissa

ma variando la distanza del microfono dall'altoparlante. Le prove sono state effettuate fornendo, come segnale di ingresso, una sinusoida a frequenza fissa e pari a  $f = 300 \text{ Hz}$ . La lunghezza d'onda del segnale è pari a  $\lambda = \frac{c}{f} \cong 1.14 \text{ m}$ , dove  $c \cong 343 \frac{\text{m}}{\text{s}}$  è la velocità di propagazione dell'onda di pressione<sup>5</sup>.

Il calcolo della funzione di trasferimento è stato effettuato con il software PULSE, con le relazioni mostrate nell'appendice A.4 nelle equazioni A.21 e A.22 e con lo stimatore  $H_1(f)$ . I risultati delle prove effettuate sono riportati in figura 3.24.

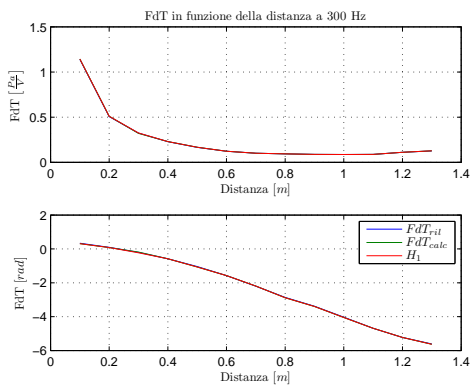


Figura 3.24: Funzione di trasferimento dell'altoparlante di controllo a  $f = 300 \text{ Hz}$  in funzione della distanza del microfono.

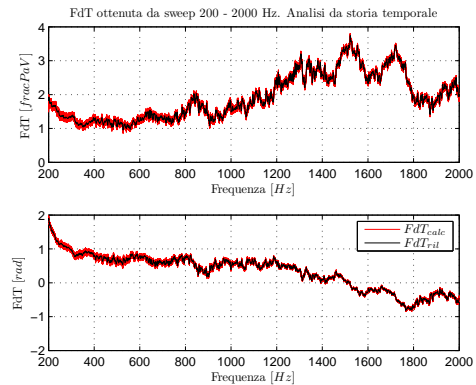


Figura 3.25: Funzione di trasferimento dell'altoparlante di controllo ottenuta con sweep in frequenza e correzione di fase.

Si nota come l'andamento della fase della funzione di trasferimento sia pressoché lineare ed assuma gli stessi valori ad una distanza relativa pari alla lunghezza d'onda  $\lambda$ . Risulta infatti verificata la seguente proporzione:

$$\frac{d}{\lambda} = \frac{\phi}{2\pi}, \quad (3.35)$$

dove  $d$  è la distanza del microfono dall'altoparlante e  $\phi$  è la fase misurata. La relazione 3.35 è stata inoltre verificata per diversi valori di frequenza, notando che la fase continua ad avere un andamento pressoché lineare con la distanza  $d$ . Attraverso quest'ultima relazione è possibile correggere la stima della fase della funzione di trasferimento calcolata precedentemente. La fase della funzione di trasferimento è stata dunque corretta aggiungendo un termine lineare con la distanza del microfono dalla cassa, secondo la relazione:

<sup>5</sup>La velocità del suono è stata calcolata considerando una temperatura dell'ambiente sonoro pari a  $20^\circ \text{ C}$ .

$$\phi_{FDT_{cor}} = \phi_{FDT_{mis}} + \frac{2\pi d}{\lambda} = \phi_{FDT_{mis}} + 2\pi d \frac{f}{c}. \quad (3.36)$$

La stima della funzione di trasferimento assume dunque i valori mostrati nelle figura 3.25.

Per quanto concerne la determinazione della funzione di trasferimento, si è osservato, nella caratterizzazione del sistema costituito dall'amplificatore e dall'altoparlante, che esso risulta essere più sensibile al rumore incorrelato presente nella misurazione dell'uscita del sistema (pressione sonora) rispetto al rumore presente all'ingresso. La determinazione della funzione di trasferimento, effettuata mediante diverse tecniche e diversi stimatori, ha permesso di validare le misure ottenute in relazione alle diverse problematiche di misura citate precedentemente.

Le prove effettuate per identificare la dipendenza delle misure dai parametri sperimentali, come la distanza del microfono dall'altoparlante, ha permesso di apportare delle correzioni alle misure effettuate. Le prove finora presentate sono state eseguite senza l'inserimento dell'amplificatore audio; in seguito ai risultati ottenuti e dopo aver verificato la loro correttezza si è provveduto ad eseguire l'inserimento dell'amplificatore descritto nel paragrafo 3.2.3.

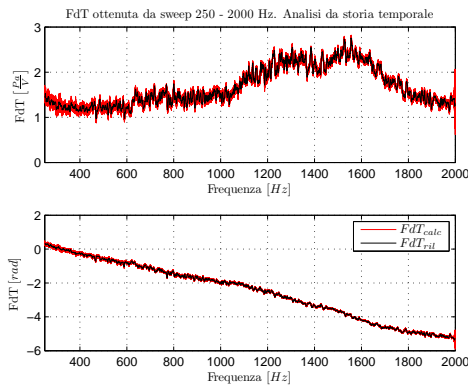


Figura 3.26: *Funzione di trasferimento dell'altoparlante di controllo ottenuta con l'inserimento dell'amplificatore.*

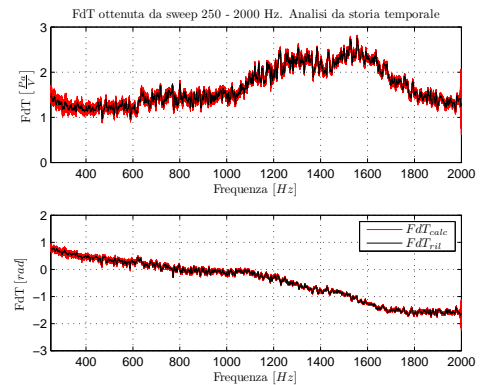


Figura 3.27: *Funzione di trasferimento dell'altoparlante di controllo ottenuta con l'inserimento dell'amplificatore e la correzione di fase.*

Con la nuova configurazione del sistema sono state eseguite solo le prove mediante sweep in frequenza, per ridurre i tempi di esecuzione dell'analisi sperimentale. Nella figura 3.26 si mostra la funzione di trasferimento ottenuta con l'inserimento dell'ampli-



ficatore. Nella figura sono mostrati, indicati con  $FdT_{ril}$  i risultati ottenuti dal software di analisi, mentre con  $FdT_{calc}$  i risultati ottenuti analizzando in post processing le storie temporali dei segnali acquisiti. Nella figura 3.26 sono presentati i risultati senza la correzione di fase, che invece è stata introdotta nella figura 3.27. In seguito si è proceduto anche alla rilevazione della funzione di trasferimento dell'altoparlante preposto alla generazione di rumore, mostrata in figura 3.28 senza la correzione di fase ed in figura 3.29 con la correzione di fase. In seguito alla rilevazione delle funzioni di trasferimento del sistema costituito da cassa-amplificatore si è proceduto alla rilevazione della funzione di trasferimento del sistema costituito solo dalla cassa acustica, acquisendo come segnale in ingresso al sistema quello presente all'uscita dell'amplificatore. Si riporta in figura 3.30 lo spettro del segnale di sweep rilevato all'uscita dell'amplificatore.

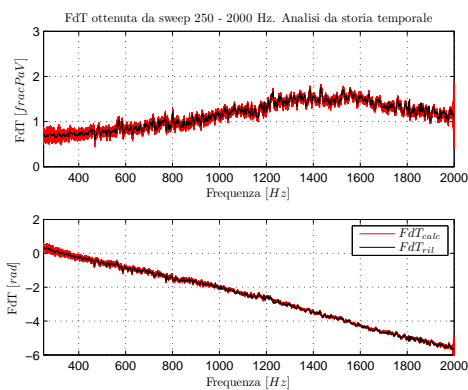


Figura 3.28: *Funzione di trasferimento dell'altoparlante per la generazione del rumore ottenuta con l'inserimento dell'amplificatore.*

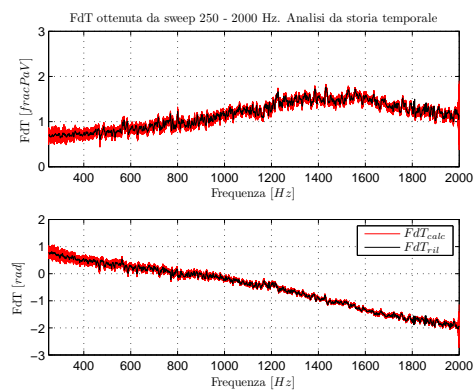


Figura 3.29: *Funzione di trasferimento dell'altoparlante per la generazione del rumore ottenuta con l'inserimento dell'amplificatore e la correzione di fase.*

Si nota la differenza rispetto allo spettro mostrato in figura 3.21 dovuta alla presenza a monte del segnale dell'amplificatore. In figura 3.31 è mostrato il confronto tra le funzioni di trasferimento del sistema cassa-amplificatore, indicata con  $FdT_{ril}$  ed il sistema costituito dalla sola cassa acustica di controllo, indicata con  $FdT_{ril}$  NO AMPLI. Vengono riportati per maggiore chiarezza della rappresentazione solo i risultati ottenuti con l'analizzatore, fermo restando che anche in questo caso si è riscontrata una buona corrispondenza con i risultati ottenuti dall'analisi in post processing delle storie temporali. Le funzioni di trasferimento vengono riportate già con la correzione di fase.

Nella figura 3.32 si riportano i risultati ottenuti eseguendo le medesime misurazioni sulla cassa acustica preposta alla generazione di rumore.

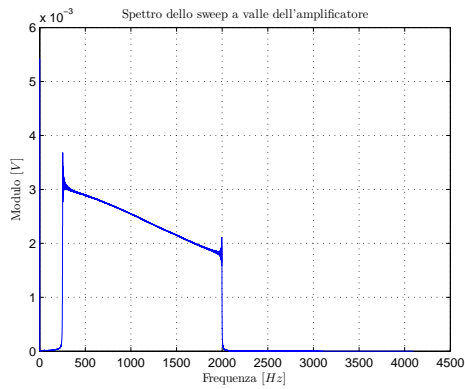


Figura 3.30: Spettro del segnale in ingresso alla cassa acustica, rilevato a valle dell'amplificatore.

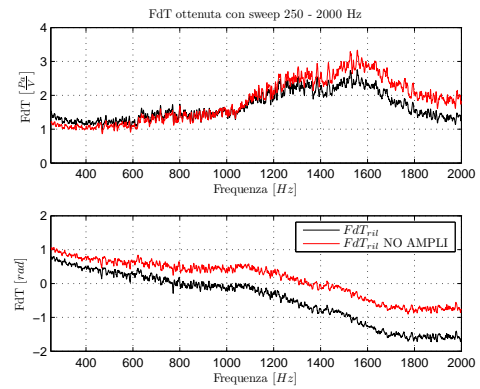


Figura 3.31: Confronto tra le funzioni di trasferimento della cassa di controllo, rilevate a monte e a valle dell'amplificatore.

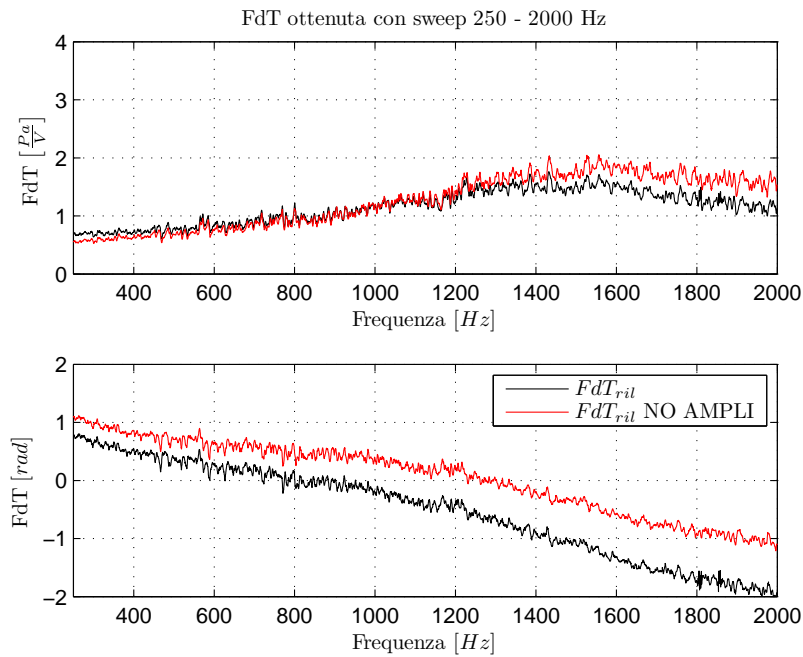


Figura 3.32: Confronto tra le funzioni di trasferimento della cassa di per la generazione del rumore, rilevate a monte e a valle dell'amplificatore.

Si nota come le funzioni di trasferimento risentano della presenza dell'amplificatore, infatti si può riconoscere nella differenza tra le funzioni di trasferimento rilevate a monte (fig. 3.28) e a valle (fig. 3.31) la funzione di trasferimento dell'amplificatore, mostrata in figura 3.17.

### 3.3 Caratterizzazione sperimentale dell'impianto di test

A seguito delle prove effettuate per la caratterizzazione degli altoparlanti, si sono eseguite altre misure per la caratterizzazione del sistema fisico oggetto di studio. In particolare sono state effettuate le seguenti misurazioni:

- Misura dell'influenza dell'amplificatore audio realizzato appositamente per questo sistema.
- Misura della funzione di trasferimento tra la cassa acustica di rumore ed il microfono di misura del sistema di controllo, il cui schema è mostrato in figura 3.33.
- Misura della funzione di trasferimento tra la cassa acustica di controllo ed il microfono di misura del sistema di controllo, il cui schema è mostrato in figura 3.34.

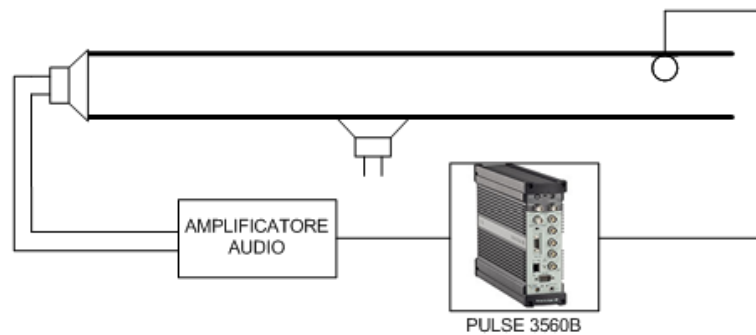


Figura 3.33: Schema della catena di misura della funzione di trasferimento tra la cassa acustica di rumore ed il microfono di misura del sistema di controllo sistema.

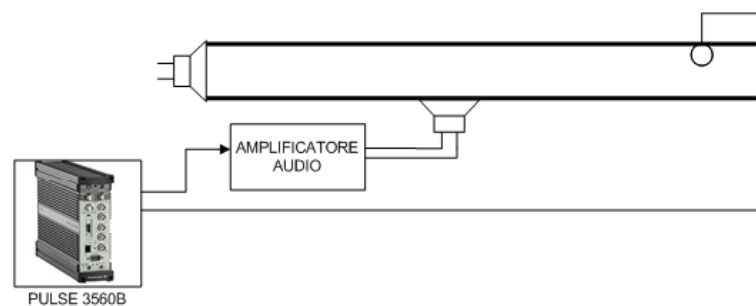


Figura 3.34: Schema della catena di misura della funzione di trasferimento tra la cassa acustica di controllo ed il microfono di misura del sistema di controllo sistema.

Queste funzioni di trasferimento, se non direttamente utilizzate per lo sviluppo del sistema di controllo, aiutano a comprendere meglio la natura del sistema e a risolvere eventuali problemi che possono presentarsi durante lo sviluppo del sistema di controllo.

Con lo scopo di poter meglio comprendere i fenomeni fisici che interessano l'ambiente sonoro oggetto di studio quando è eccitato da onde di pressione, si è proceduto alla rilevazione della funzione di trasferimento tra la cassa acustica per la generazione del rumore ed il microfono posto all'interno del condotto e utilizzato dall'algoritmo di controllo, e della funzione di trasferimento tra la cassa acustica di controllo ed il medesimo microfono. Lo schema della catena di misura è rispettivamente quello mostrato nelle figure 3.33 e 3.34.

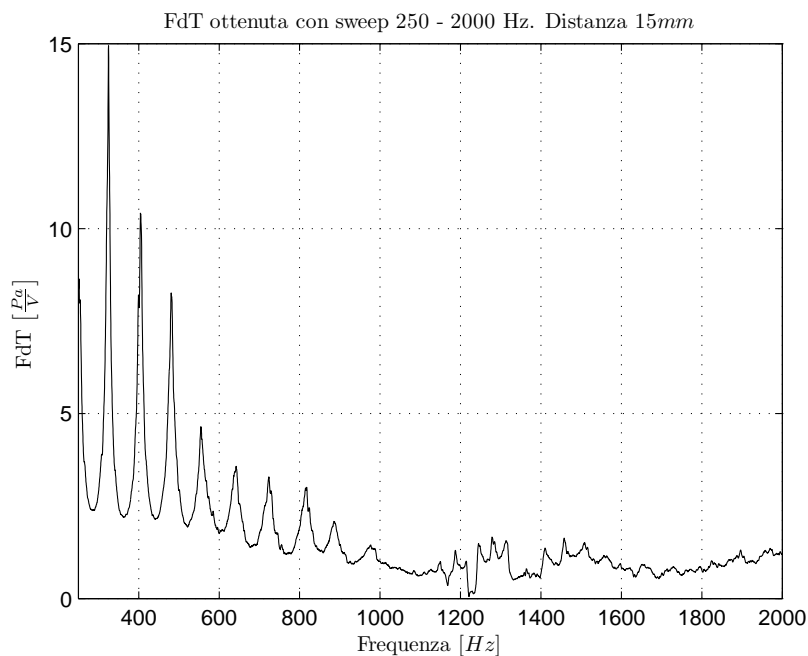


Figura 3.35: Funzione di trasferimento tra la cassa preposta alla generazione del rumore ed il microfono posto all'interno del condotto con distanza della cassa acustica pari a 10 mm.

Nella figura 3.35 viene mostrata la funzione di trasferimento tra la cassa preposta alla generazione del rumore ed il microfono posto all'interno del condotto. Essa è stata misurata fornendo come segnale di ingresso al sistema uno sweep in frequenza da 250 a 2000 Hz e fissando la distanza tra la cassa acustica preposta alla generazione del rumore ed il condotto pari a 15 mm. I parametri dell'analizzatore sono stati mantenuti pari a quelli utilizzati per la misura delle funzioni di trasferimento delle casse acustiche. Nella figura 3.35 sono evidenti, soprattutto nella gamma di frequenza compresa tra 200

e  $1000\text{ Hz}$ , le frequenze proprie del sistema, caratterizzate dai picchi della funzione di trasferimento. Nella figura è mostrato il risultato ottenuto dal software di analisi, tuttavia anche in questo caso è stata eseguita un'analisi in post processing sulle storie temporali dei dati campionati verificando la corrispondenza dei risultati con quelli ottenuti dal software di analisi. Da un'analisi più attenta della funzione di trasferimento, mostrata in figura 3.35, si nota come le frequenze proprie del sistema non siano corrispondenti ad alcuna delle situazioni ideali in cui può essere definito un modello analitico del sistema, che equivalgono alle condizioni al contorno corrispondenti ad un condotto aperto alle estremità ed un condotto chiuso ad una estremità, le cui analisi sono riportate nella sezione 2.1.

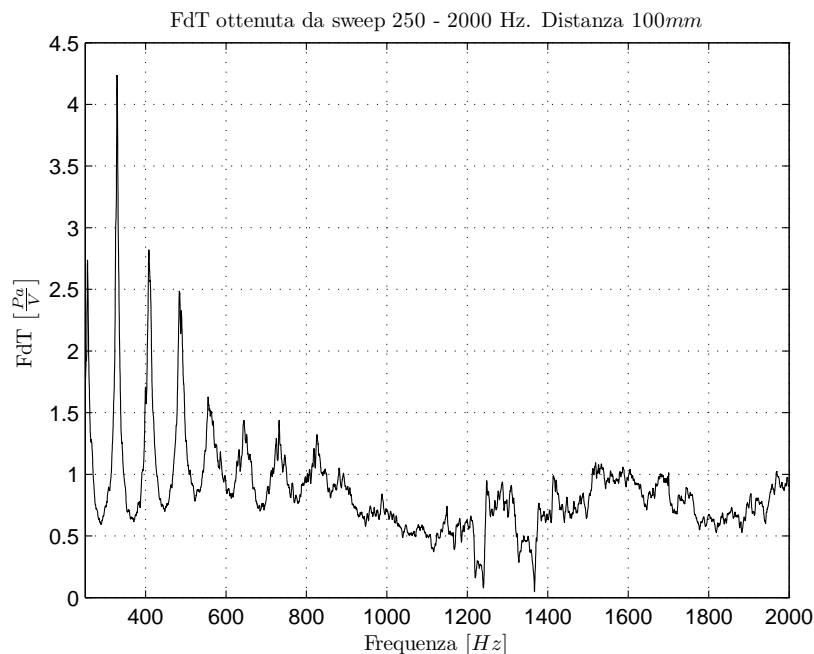


Figura 3.36: *Funzione di trasferimento tra la cassa preposta alla generazione del rumore ed il microfono posto all'interno del condotto con distanza della cassa acustica pari a 100 mm.*

Le frequenze proprie del sistema sembrano essere comprese tra queste due situazioni limite. Allo scopo di verificare tale affermazione si è effettuata un'ulteriore misura della funzione di trasferimento ponendo la cassa acustica ad una distanza pari a  $100\text{ mm}$ . Il risultato di questa misura è mostrato in figura 3.36. Si nota come le frequenze proprie si spostino leggermente, migrando verso quelle ipotetiche calcolate nella condizione di tubo aperto. L'andamento delle frequenze proprie comprese tra  $200$  e  $1000\text{ Hz}$  è riportato nella tabella 3.1 dove è possibile notare maggiormente questo effetto.

<b>Tubo aperto</b> [Hz]	<b>Cassa<sub>rum</sub></b> <b>100 mm</b> [Hz]	<b>Cassa<sub>rum</sub></b> <b>15 mm</b> [Hz]	<b>Tubo chiuso</b> [Hz]
343.0	329.1	325.0	300.1
428.8	408.4	404.4	343.0
514.5	484.1	480.9	385.9
600.3	555.6	554.7	428.8
686.0	645.3	642.2	471.6
771.8	732.2	723.4	514.5
857.5	825.9	817.2	557.4

Tabella 3.1: Confronto tra le frequenze proprie dell'ambiente sonoro oggetto di studio, con diverse configurazioni delle condizioni al contorno.

Questo fenomeno è stato indagato solo per via sperimentale, in quanto la sua modellazione dal punto di vista analitico è molto complessa, mentre dal punto di vista numerico avrebbe richiesto una campagna di simulazioni che si è preferito evitare data la durata delle simulazioni in relazione alla rilevanza del risultato ottenuto rispetto allo studio che si intende effettuare.

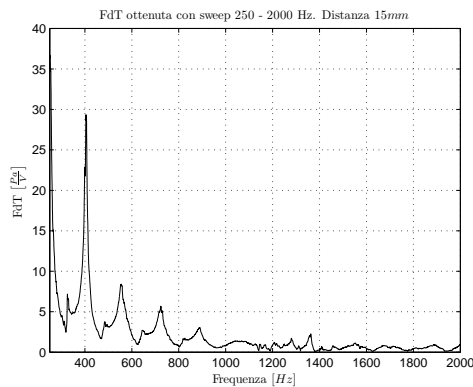


Figura 3.37: Funzione di trasferimento tra la cassa acustica di controllo ed il microfono posto all'interno del condotto con distanza della cassa acustica di rumore pari a 15mm.

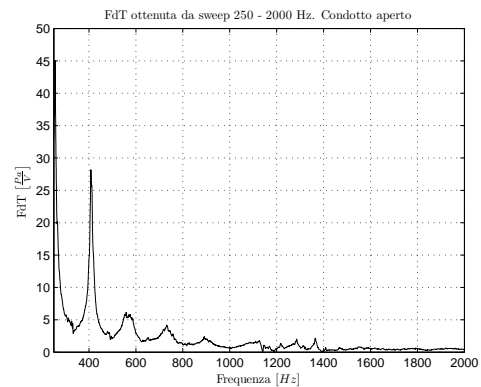


Figura 3.38: Funzione di trasferimento tra la cassa acustica di controllo ed il microfono posto all'interno del condotto aperto ad entrambe le estremità.

Dall'osservazione della funzione di trasferimento della cassa acustica di controllo rispetto al microfono posizionato all'interno del condotto (figura 3.37), si nota come

il suo andamento sia più disturbato da fenomeni di riflessione, data la posizione e l'orientazione della cassa acustica di controllo all'interno del condotto. Tuttavia, la misura di questa funzione di trasferimento può risultare utile nelle fasi di sviluppo, implementazione e verifica dell'algoritmo di controllo. La misura è stata eseguita con due configurazioni del sistema, la prima è mostrata in figura 3.37 e prevede una distanza tra l'imboccatura del condotto e la cassa acustica preposta alla generazione del rumore pari a  $15\text{ mm}$ , la seconda è mostrata in figura 3.38 ed è stata eseguita in configurazione di condotto aperto ad entrambe le estremità. Anche in questo caso si nota come la misura porti ad una agevole identificazione delle risonanze del sistema nell'intervallo di frequenze pari a  $250 \div 1000\text{ Hz}$ , e che le risonanze identificate non corrispondano a quelle ideali del modello analitico.

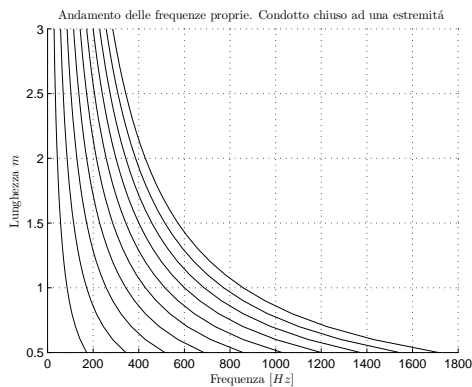


Figura 3.39: *Andamento delle frequenze proprie dell'ambiente sonoro in configurazione di condotto chiuso ad una estremità.*

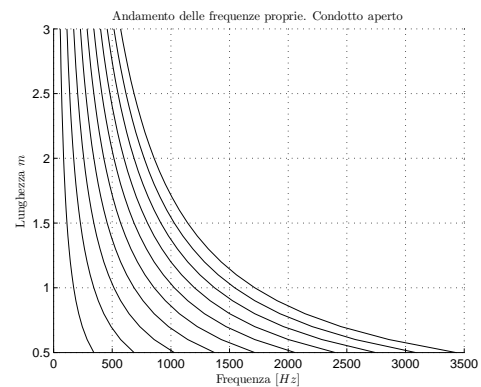


Figura 3.40: *Andamento delle frequenze proprie dell'ambiente sonoro in configurazione di condotto aperto.*

Con lo scopo di mostrare l'influenza dei parametri di progetto del condotto sulle frequenze proprie dell'ambiente sonoro, si mostrano nelle figure 3.39 e 3.40 due grafici che riportano l'andamento delle frequenze proprie in funzione della lunghezza del condotto, nelle due configurazioni ideali di condotto aperto ad entrambe le estremità e condotto chiuso ad una estremità. Nei grafici sono visualizzati i primi dieci modi dell'ambiente sonoro. Si nota come la lunghezza del condotto, influenza notevolmente le frequenze proprie del sistema soprattutto per lunghezze del condotto ridotte. Da non sottovalutare è anche il rapporto tra le dimensioni del condotto, che deve sempre privilegiare una direzione particolare allo scopo di avere un ambiente sonoro che possa essere trattato come un problema monodimensionale.

Nella figura 3.41 viene mostrata la sovrapposizione dei risultati ottenuti dalle diverse analisi effettuate. Sono riportate le frequenze di risonanza calcolate analiticamente nella sezione 2.1 e mediante modellazione ad elementi finiti, descritta nel paragrafo 2.1.1, oltre all'andamento sperimentale della funzione di trasferimento mostrata in figura 3.36. Dal confronto si nota come l'andamento delle frequenze di risonanza non concorde tra il calcolo analitico e quello numerico, ma entrambi i modelli non rispecchiano il comportamento reale del sistema. Le frequenze di risonanza reali sono infatti comprese tra le frequenze di risonanza calcolate in configurazione di condotto aperto ad entrambe le estremità e quelle di condotto chiuso ad una estremità, sintomo del fatto che la presenza della cassa acustica per la generazione di rumore influisce sulle frequenze proprie del condotto. Una sintesi del confronto tra frequenze è mostrata in figura 3.42.

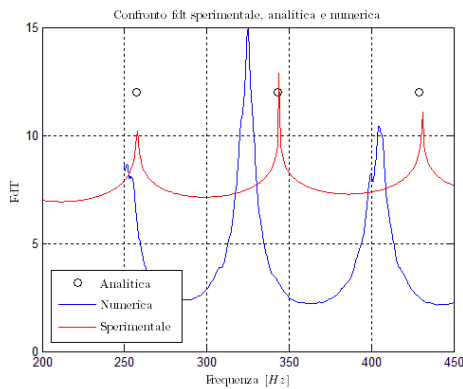


Figura 3.41: *Andamento delle frequenze proprie dell'ambiente sonoro in configurazione di condotto aperto. Confronto fra i risultati ottenuti.*

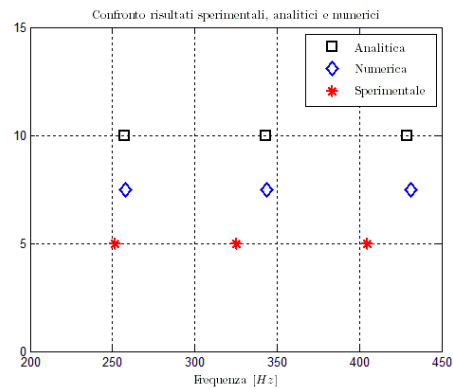


Figura 3.42: *Andamento delle frequenze proprie dell'ambiente sonoro in configurazione di condotto aperto. Sintesi dei risultati ottenuti*

Oltre all'analisi sperimentale delle frequenze di risonanza del condotto, necessaria per avere una maggiore conoscenza del sistema fisico, è utile eseguire un confronto sperimentale con i risultati delle tecniche di controllo attivo del rumore, prima di procedere all'implementazione vera e propria dell'algoritmo di controllo.

### Mappatura dell'RMS di pressione sonora

In seguito alle considerazioni effettuate nel paragrafo 2.3 si è ritenuto opportuno mappare sia analiticamente sia sperimentalmente il dominio entro cui l'algoritmo di controllo si troverà ad operare.



La mappatura è stata effettuata considerando, per il segnale di rumore, una sinusoide a parametri costanti, mentre per il segnale di controllo un segnale sinusoidale con parametri variabili in determinati range. I valori dei parametri assegnati ai segnali di rumore e di controllo sono i seguenti:

- Segnale di rumore: sinusoide con ampiezza pari a  $A_{rum} = 1 Pa$ , fase pari a  $60^\circ$  e frequenza pari a  $f_{rum} = 1000 Hz$ .
- Segnale di controllo: sinusoide la cui ampiezza viene fatta variare all'interno del range  $-1.5 \leq A_{cont} \leq 1.5 Pa$  e con fase variabile nel range  $0^\circ \leq A_{cont} \leq 360^\circ$ .

Se si vuole effettuare un confronto tra il calcolo analitico e i risultati sperimentali, occorre considerare che sperimentalmente la mappatura viene eseguita scrivendo un software opportuno per il comando del sistema di controllo, utilizzato in questa fase come analizzatore di segnale. Mediante la scheda per la generazione delle uscite del controllore vengono generati il segnale di rumore, ovvero una sinusoide con ampiezza pari ad  $A_{rum}$  e frequenza pari a  $f_{rum}$ , ed il segnale di controllo, la cui ampiezza è espressa in Pascal e pertanto viene trasformata in valore di tensione che possono essere generati dalla scheda di output del controllore.

Per effettuare un confronto tra i risultati analitici e quelli sperimentali, occorre anche considerare che sperimentalmente i segnali generati dalla scheda di output vengono inviati ad un amplificatore e successivamente alle casse acustiche che li trasformano in onde di pressione rilevate poi dal microfono posto all'interno del tubo. Per questo motivo occorre effettuare un'ulteriore trasformazione delle grandezze analitiche considerando la funzione di trasferimento tra le uscite del controllore ed il segnale letto dal microfono di misura. Inoltre occorre considerare che il segnale viene acquisito per un periodo di tempo limitato, pari a  $100 ms$ , e campionato ad una frequenza di  $50 kHz$ .

Nelle figure 3.43 e 3.44 viene riportato l'andamento della funzione RMS ottenuta sperimentalmente. Si può notare come anche sperimentalmente la funzione presenti due minimi e quindi due soluzioni al problema di minimizzazione. Il confronto tra i calcoli analitici (fig. 2.16) e i risultati sperimentali è mostrato nella figura 3.45. È evidente la forte somiglianza tra i dati ottenuti dalla sperimentazione e quelli ottenuti analiticamente, e mostrati nella figura 2.16. La corretta scalatura di entrambe le mappature fa sì che esse possano essere rappresentate nello stesso spazio cartesiano, mostrando la buona aderenza tra il modello analitico e il sistema fisico. Dall'osservazione dei grafici appena riportati è evidente la presenza di un punto a derivata nulla che non è un mi-

nimo della funzione, pertanto, l'algoritmo di ricerca del minimo, dovrà prevedere delle soglie accettabili sul valore dell'RMS in modo da consentire allo stesso di eseguire in modo affidabile il controllo del rumore.

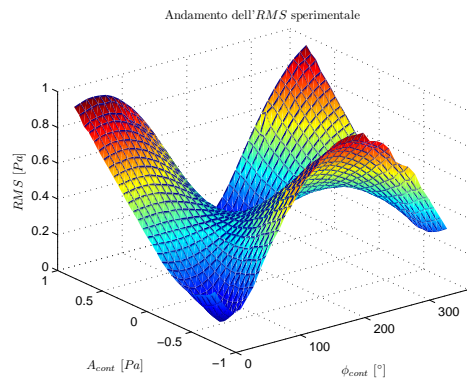


Figura 3.43: Mappatura sperimentale dell'RMS - 3D.

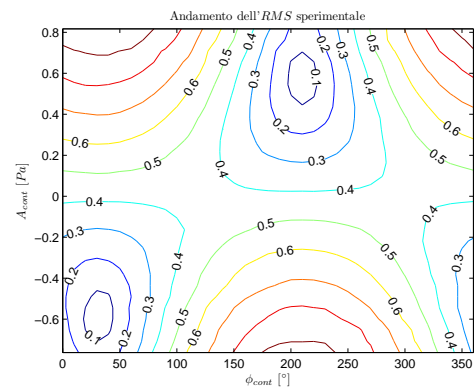


Figura 3.44: Mappatura sperimentale dell'RMS - isolinee.

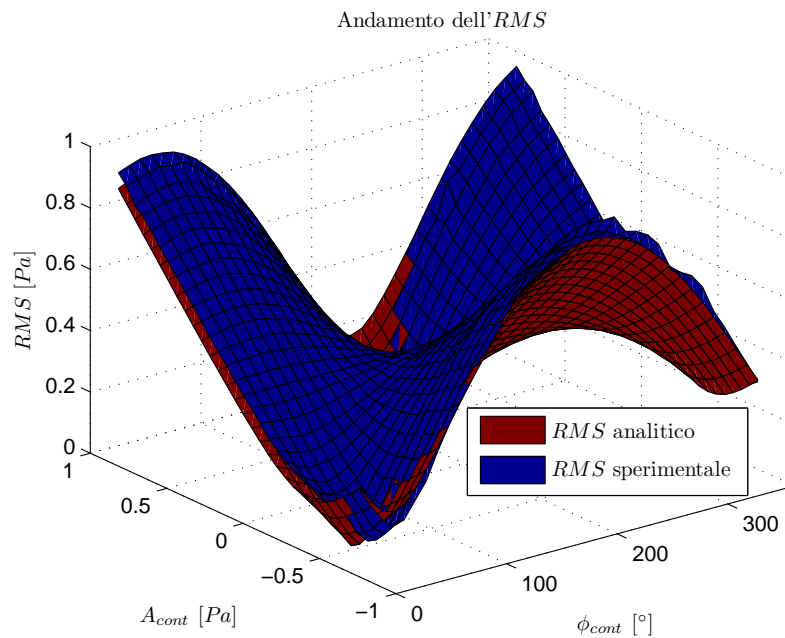


Figura 3.45: Confronto tra la mappatura analitica e sperimentale dell'andamento dell'RMS.

### Mappatura dell'ampiezza spettrale della pressione sonora

Il processo per determinare il segnale da fornire alla cassa acustica di controllo che minimizza l'ampiezza del picco dello spettro del rumore è lo stesso utilizzato nella minimizzazione dell'RMS (si veda il paragrafo 2.3.1), a patto di sostituire il gradiente della funzione RMS con il gradiente dell'ampiezza spettrale dell'armonica considerata. Per questo motivo si è ritenuto opportuno eseguire una nuova mappatura del dominio di interesse, considerando l'andamento dell'ampiezza spettrale alla frequenza  $f_{rum} = f_{cont}$  in funzione delle grandezze del segnale di controllo. Per poter effettuare un confronto tra la mappatura analitica (fig. 2.22) e quella sperimentale dell'andamento dell'ampiezza spettrale sono state effettuate le stesse scalature descritte per la mappatura dell'RMS. I parametri del segnale di rumore sono gli stessi utilizzati per determinare l'andamento della funzione RMS mostrato in figura 3.43. Nelle figure 3.46 e 3.47 si riporta l'andamento del valore del picco spettrale ottenuto sperimentalmente, e nella figura 3.48 si riporta il confronto tra i risultati analitici e sperimentali.

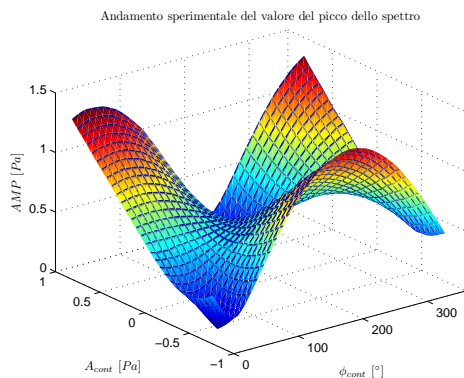


Figura 3.46: *Mappatura sperimentale dell'ampiezza spettrale - 3D.*

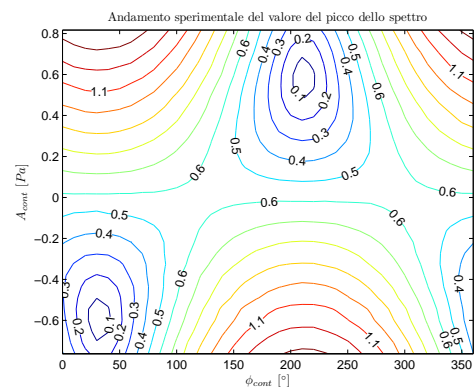


Figura 3.47: *Mappatura dell'ampiezza spettrale - isolinee.*

Anche in questo caso è evidente la forte somiglianza delle funzioni ottenute dalla mappatura sperimentale e dall'analisi matematica. Per le differenze tra i due grafici valgono le considerazioni espresse nel paragrafo relativo alla mappatura dell'RMS. Dall'osservazione dei grafici appena riportati è evidente la presenza di un punto a derivata nulla che non è un minimo della funzione, pertanto, l'algoritmo di ricerca del minimo, dovrà prevedere delle soglie accettabili sul valore dell'ampiezza spettrale in modo da consentire allo stesso algoritmo di giungere al termine in un numero accettabile di iterazioni.

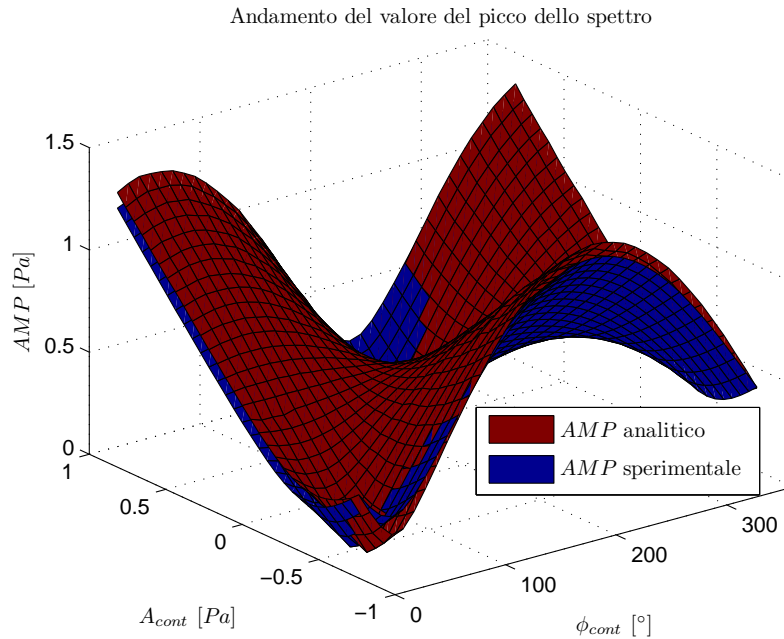


Figura 3.48: Confronto tra la mappatura analitica e sperimentale dell'andamento dell'ampiezza spettrale della pressione sonora.

## 3.4 Risultati sperimentali

In questo paragrafo si riportano i risultati sperimentali ottenuti implementando gli algoritmi di controllo (par. 2.3) nell'hardware National Instruments presentato precedentemente. Per ciascun algoritmo viene presentato inizialmente lo schema a blocchi riportante le funzioni che vengono implementate nell'esecuzione del ciclo di controllo e successivamente una breve descrizione delle funzioni realizzate nel codice sorgente con tutte le scelte che sono state adottate durante l'implementazione sul controllore.

### 3.4.1 Tecnica di controllo ARMS

Dalla verifica sperimentale della funzione obiettivo a seguito della mappatura sperimentale della funzione RMS (fig. 3.45) è possibile definire la sequenza delle operazioni da eseguire per ricercarne il minimo. Data la forte somiglianza tra il comportamento analitico e quello sperimentale è possibile procedere con lo sviluppo e l'implementazione del codice di controllo secondo lo schema a blocchi riportato in figura 3.49. Questo schema mostra il principio di funzionamento dell'algoritmo di ricerca del minimo secondo la metodologia del calcolo dell'RMS riportata nel paragrafo 2.3.1 che permette di

abbattere un rumore stazionario costituito da un tono puro che si propaga all'interno del condotto ove la propagazione si ritiene monodimensionale.

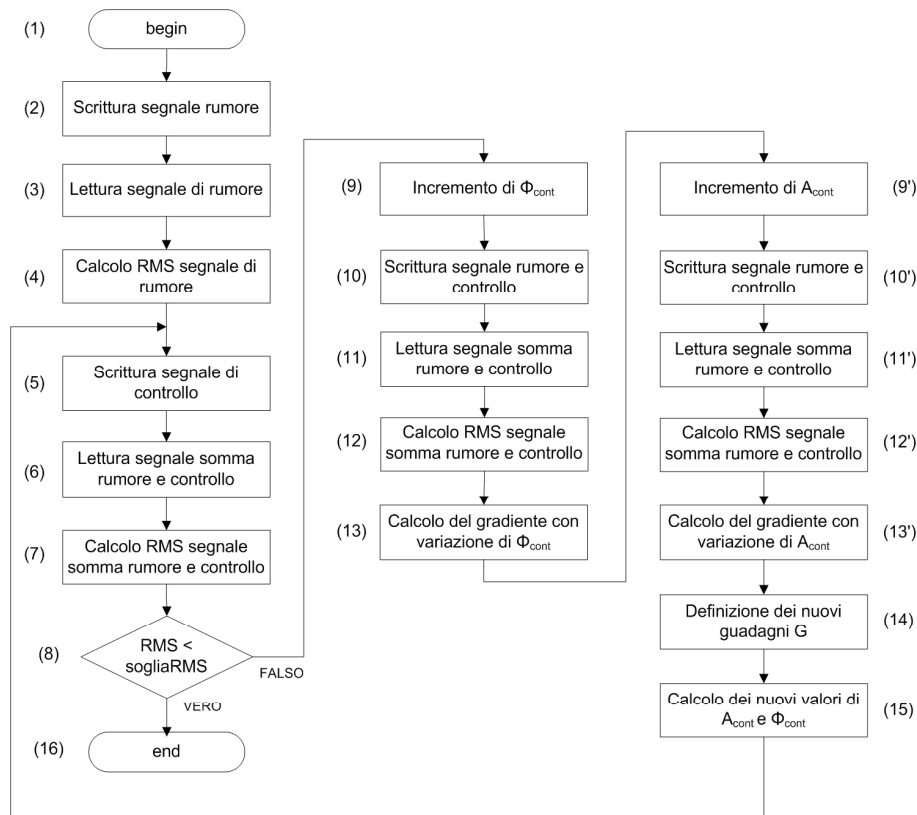


Figura 3.49: Schema a blocchi dell'algoritmo di ricerca del minimo mediante la valutazione del valore quadratico medio della pressione sonora.

Di seguito vengono commentati i singoli passi dell'algoritmo secondo quanto riportato nello schema a blocchi:

1. Inizializzazione delle variabili e delle costanti necessarie al primo passo di iterazione. In questa fase risulta di particolare importanza la scelta del valore della soglia dell'RMS, ossia del valore limite che deve essere raggiunto affinché l'algoritmo di iterazione possa terminare e quindi mantenere costanti i parametri di controllo determinati durante la minimizzazione del funzionale. Un altro parametro che viene impostato in questo step riguarda il valore del guadagno iniziale  $G_1^{k=0}$  che viene tarato diversamente in funzione della frequenza del rumore che si intende abbattere. Sperimentalmente è emersa infatti la necessità di procedere ad un affinamento del parametro  $G_1^{k=0}$  in funzione della frequenza del rumore da abbattere al fine di garantire la stabilità e la convergenza del metodo di minimizzazione

in tutto il range di frequenze di interesse. Per quanto riguarda il valore iniziale del guadagno  $G_1^{k=0}$  sperimentalmente risulta possibile assegnare valori maggiori rispetto alle basse frequenze, poiché valori troppo elevati causano delle instabilità nella ricerca del minimo.

Il segnale di comando delle casse acustiche viene scritto all'interno di un buffer, e successivamente viene generato ciclicamente e giunge agli attuatori. Nel momento in cui l'algoritmo di controllo assegna un nuovo segnale, viene inviato un segnale di interrupt alla scheda di output del controllore che provvede a fermare la generazione del segnale precedente (indipendentemente dal valore di tensione presente sulle uscite), viene caricato nel buffer il nuovo segnale e la generazione ciclica riprende dal primo valore presente nel buffer. Questo tipo di funzionamento causa una discontinuità nel segnale presente all'uscita della scheda di output, che si ripercuote sulla cassa di controllo e di conseguenza sulla pressione sonora generata. Poiché si tratta di un fenomeno transitorio e sperimentalmente si è visto dipendere dalla frequenza del segnale generato, si è interposto un tempo di attesa  $T_{attesa}$  tra la scrittura del segnale sulle casse acustiche e la lettura della pressione sonora sul microfono, in modo da acquisire ed analizzare il segnale a regime. In questo step vengono inoltre fissate le soglie sulla variazione consentita delle grandezze di controllo  $A_{SAT}$  e  $\phi_{SAT}$ , queste risultano necessarie per limitare l'elevata variazione delle grandezze che potrebbe risultare dal calcolo del gradiente. Qualora la variazione imposta dal gradiente ( $G_1^k \nabla RMS$ ) superasse il valore consentito di soglia, verrebbe nuovamente calcolato il guadagno  $G_1$  definendo un nuovo guadagno  $G_2^k$  in modo tale che la variazione imposta alla grandezza che ha superato il limite sia la massima consentita. In tal modo è possibile conservare la corretta direzione di ricerca del minimo lungo le direzioni dell'ampiezza e della fase, riducendo il passo effettuato per garantire la convergenza del metodo.

Di fondamentale importanza risulta anche la scelta degli incrementi delle variabili di controllo  $\Delta A_{cont}$  e  $\Delta \phi_{cont}$  da fornire al segnale della cassa di controllo. Per l'ampiezza si è scelto un incremento  $\Delta A_{cont}$  pari ad 1/100 del fondoscala della scheda di output (fissato a  $\pm 3 V$  per consentire di comandare correttamente le casse evitando sovratensioni), mentre per la fase si è scelta una variazione  $\Delta \phi_{cont}$  pari ad 1/100 di un angolo giro.

2. Questo step viene eseguito solo in fase di messa a punto dell'algoritmo al fine di generare il segnale da fornire alla cassa acustica di rumore. Nelle reali applicazioni

il rumore sarà quello generato da una qualsiasi sorgente. Prevedendo quindi di generare il rumore per mezzo di un altoparlante, noti i valori di ampiezza e fase dello stesso, in questo passo dell'algoritmo viene definito il buffer di scrittura, quindi aggiornate le uscite della scheda di output e infine viene generato in modo continuativo il segnale di rumore.

3. Utilizzando il microfono di errore viene letto il segnale di rumore presente all'interno del condotto, e grazie alle schede di acquisizione i dati vengono memorizzati in un buffer e i e disponibili attraverso un vettore. In questa sede la frequenza del segnale di rumore risulta già nota, tuttavia qualora non fosse nota sarà sufficiente effettuare un'analisi spettrale sui dati acquisiti e determinare quindi l'ampiezza e la frequenza del segnale di rumore in ingresso.
4. Viene calcolato il valore quadratico medio del solo segnale di rumore secondo la relazione 2.43. Il segnale di rumore che viene utilizzato nel calcolo è quello presente all'interno del buffer, infatti una volta noto questo vettore ( $a_{mic}$ ) il calcolo dell'RMS può essere effettuato secondo la seguente funzione:

$$RMS^k = \sqrt{\frac{\sum_{j=1}^N a_{mic}^k(j)^2}{N}}, \quad (3.37)$$

dove  $N$  è il numero di punti acquisiti del segnale del microfono, pari alla dimensione del buffer.

5. Viene attivato il segnale di controllo con le dovute attenzioni in riferimento ai valori iniziali  $A_{cont}^{k=0}$  e  $\phi_{cont}^{k=0}$ . La scelta di questi valori deve essere fatta considerando la presenza nella funzione RMS di punti a gradiente nullo ma non di minimo. Da ora nel sistema sarà presente sia il segnale di rumore che quello di controllo pertanto il segnale letto dal microfono di errore sarà costituito dalla somma dei segnali presenti, nonché da eventuale rumore esterno presente nell'ambiente.
6. Mediante il microfono e le schede degli ingressi viene letto e acquisito il segnale dato dalla somma dei segnali di rumore e di controllo.
7. Come visto al punto 4) viene calcolato il valore quadratico medio del segnale acquisito al passo 6).

8. In questo passo viene confrontato il valore di  $RMS^k$  raggiunto con il valore di soglia scelto al punto 1), qualora questa condizione risulta verificata si procede al punto 16) altrimenti si procede al punto successivo.
9. In questa fase viene fornito un incremento alle variabili di controllo. Si ricorda infatti, secondo le relazioni precedenti che, per la valutazione del gradiente, è necessario calcolare l'RMS del segnale dopo aver fornito una variazione ai parametri di controllo. Per i valori scelti per gli incrementi delle variabili di controllo  $A_{cont}$  e  $\phi_{cont}$  si veda il punto 1).
10. Viene scritto sul buffer della scheda di generazione delle uscite il segnale di controllo dato dalla nuova variazione dei parametri, le cui caratteristiche sono:

$$A_{cont}^{temp} = A_{cont}^k + \Delta A_{cont}, \quad (3.38)$$

$$\phi_{cont}^{temp} = \phi_{cont}^k + \Delta \phi_{cont}. \quad (3.39)$$

L'apice *temp* indica che questi valori sono dei valori temporanei infatti questi non devono essere confusi con i valori dei parametri di controllo che vengono determinati con le relazioni 2.47 e 2.48.

11. Mediante il microfono e le schede degli ingressi viene letto e acquisito il nuovo segnale.
12. Viene calcolato il valore quadratico medio del segnale acquisito al passo 11).
13. Secondo le relazioni esposte nel paragrafo 2.3.1 viene calcolato il gradiente nella direzione della fase. I punti successivi da 9') a 13') svolgono sull'ampiezza le stesse operazioni eseguite sulla fase. Viene fornito un incremento sulla variabile  $A_{cont}$  e viene valutato il valore quadratico medio della pressione sonora in modo tale da calcolare il gradiente nella direzione dell'ampiezza.
14. In questo step viene rivalutato il valore del guadagno qualora la variazione delle grandezze di controllo imposte dal gradiente,  $G_1^k \nabla RMS_A^k$  e  $G_1^k \nabla RMS_\phi^k$ , superino i limiti  $A_{SAT}$  e  $\phi_{SAT}$  consentiti. Si possono presentare quattro casi:

- Viene superato il limite sulla fase ( $G_1^k \nabla RMS_\phi^k > \phi_{SAT}$ ). Si ha:

$$G_2^k = G_{2\phi}^k = \frac{\phi_{SAT}}{|\nabla RMS_\phi^k|}. \quad (3.40)$$



- Viene superato il limite sull'ampiezza ( $G_1^k \nabla RMS_A^k > A_{SAT}$ ). In questo caso si ha:

$$G_2^k = G_{2A}^k = \frac{A_{SAT}}{|\nabla RMS_A^k|}. \quad (3.41)$$

- Viene superato sia il limite di ampiezza che il limite di fase. In questo caso si determina il valore di  $G_2^k$  secondo la soglia più restrittiva, prendendo il valore di guadagno più piccolo tra quelli mostrati nei punti precedenti.

$$G_2^k = \min(G_{2A}^k, G_{2\phi}^k). \quad (3.42)$$

- Non viene superato nessuno dei limiti imposti. In questo caso il valore del guadagno è assunto pari a quello del passo precedente:

$$G_2^k = G_1^k. \quad (3.43)$$

Dopo la rivalutazione del guadagno  $G_2^k$  a seguito del superamento di uno dei limiti consentiti sui parametri di controllo, viene valutato l'eventuale oscillazione attorno al punto di minimo della funzione RMS. Questo controllo viene effettuato controllando se il valore dei gradienti nelle due direzioni cambia di segno entro un numero prefissato di iterazioni. Qualora si verifichi tale circostanza si procede a dimezzare il valore del guadagno  $G_1^k$ . Tuttavia per evitare una riduzione eccessiva del guadagno  $G_1^k$  che potrebbe causare una lentezza ingiustificata nella ricerca del minimo viene fissata anche una soglia inferiore  $G_{1MIN}^k$ :

$$G_1^k \geq G_{1MIN}^k. \quad (3.44)$$

15. Vengono calcolati i nuovi valori da assegnare ai parametri di controllo secondo le seguenti relazioni:

$$A_{cont}^{k+1} = A_{cont}^k - G_2^k \nabla RMS_A^k, \quad (3.45)$$

$$\phi_{cont}^{k+1} = \phi_{cont}^k - G_2^k \nabla RMS_\phi^k. \quad (3.46)$$

16. L'algoritmo termina e viene mantenuta la soluzione di minimo raggiunta.

Nell'appendice D.1 viene riportato il codice sorgente che esegue quanto riportato nello schema a blocchi mostrato in figura 3.49 e vengono descritte in dettaglio tutte le funzioni che permettono di eseguire l'implementazione del codice sul controllore NI PXI.

### Risultati sperimentali

Attraverso il monitoraggio di alcune importanti grandezze del codice è stato possibile analizzare il comportamento dell'algoritmo alle diverse frequenze del segnale di rumore e apportare al codice di controllo una serie di modifiche sulle variabili dell'algoritmo, garantendo allo stesso un migliore comportamento in termini di convergenza e di tempo di raggiungimento della soglia di rumore desiderata. Le grandezze, monitorate in funzione del passo di iterazione, sono le seguenti:

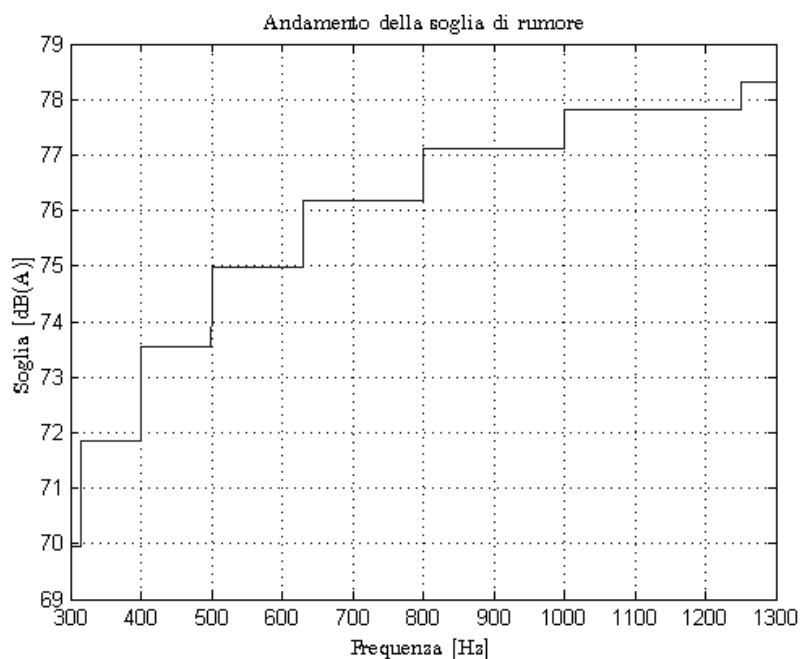


Figura 3.50: Soglia di rumore impostata nell'algoritmo di controllo attivo del rumore.

- Valore quadratico medio  $RMS_1$  del segnale costituito dalla somma del segnale di rumore e di controllo. Nella visualizzazione di questo andamento (figura 3.51) vengono inoltre mostrati il valore quadratico medio del solo segnale di rumore  $RMS_0$  presente nel condotto e i valori di soglia di rumore desiderati. Secondo quanto mostrato in figura 3.50, il valore della soglia di rumore desiderata (1<sup>a</sup> so-

*glia*) è stata fissata in funzione della frequenza al fine di tenere in considerazione l'andamento della curva di ponderazione A (tab. 1.3) tuttavia, qualora il numero di iterazioni per il raggiungimento del minimo risulti troppo elevato ed in particolare superiore a 15 iterazioni, è sufficiente il soddisfacimento di una soglia meno stringente e pari al 20 % del valore  $RMS_0$  ( $2^a$  soglia): in questo caso l'algoritmo termina la ricerca del minimo quando il rumore viene ridotto di circa 14 dB.

- Gradiente  $\nabla RMS_A$  dell'ampiezza del segnale di controllo.
- Gradiente  $\nabla RMS_\phi$  della fase del segnale di controllo.
- Valori quadratici medi  $RMS_2$  e  $RMS_3$  associati alle variazioni dei parametri delle variabili del segnale di controllo. In particolare  $RMS_2$  è il valore quadratico medio associato all'incremento  $\Delta\phi_{cont}$  della fase del segnale di controllo, mentre  $RMS_3$  è legato all'incremento  $\Delta A_{cont}$  dell'ampiezza dello stesso segnale di controllo.
- Ampiezza  $A_{cont}$  del segnale di controllo.
- Fase  $\phi_{cont}$  del segnale di controllo.
- Guadagno  $G_2^k$  associato ai nuovi valori delle variabili  $A_{cont}$  e  $\phi_{cont}$ .

A titolo di esempio si riportano i risultati sperimentali ottenuti abbattendo un segnale di rumore alla frequenza  $f_{rum} = 800 Hz$ . Nella figura 3.51 è riportato l'andamento del valore quadratico medio  $RMS_1$  in funzione del passo di iterazione: si nota che questo valore decresce, rispetto al valore iniziale  $RMS_0$ , all'aumentare del passo di iterazione e raggiunge la soglia desiderata di rumore in 24 iterazioni. In questo caso viene raggiunta la seconda soglia di rumore, ossia la soglia dettata dal limite di iterazioni impostato nell'algoritmo. L'andamento monotono decrescente di  $RMS_1$  è sintomo di una adeguata scelta dei parametri del metodo di minimizzazione.

Nelle figure 3.52 e 3.54 viene riportato l'andamento dei gradienti dei parametri di controllo  $A_{cont}$  e  $\phi_{cont}$ . È possibile notare un andamento non molto regolare di queste funzioni, infatti la funzione  $RMS$  è molto suscettibile all'ambiente in cui viene effettuata la misura; per quanto riguarda il gradiente  $\nabla RMS_\phi$  della fase del segnale di controllo è possibile notare come il valore di questa funzione, a partire dall'iterazione 14, oscilla attorno allo zero, a significare il raggiungimento del valore di fase per cui la funzione RMS ha un minimo. Questo è ancor più evidente nella figura 3.55 che mostra l'andamento della fase  $\phi_{cont}$  del segnale di controllo. L'ampiezza dopo 24 iterazioni non

ha ancora raggiunto una condizione di stabilità (figure 3.52 e 3.53), tuttavia il valore di ampiezza raggiunto è tale da soddisfare la seconda soglia sul valore di RMS desiderato.

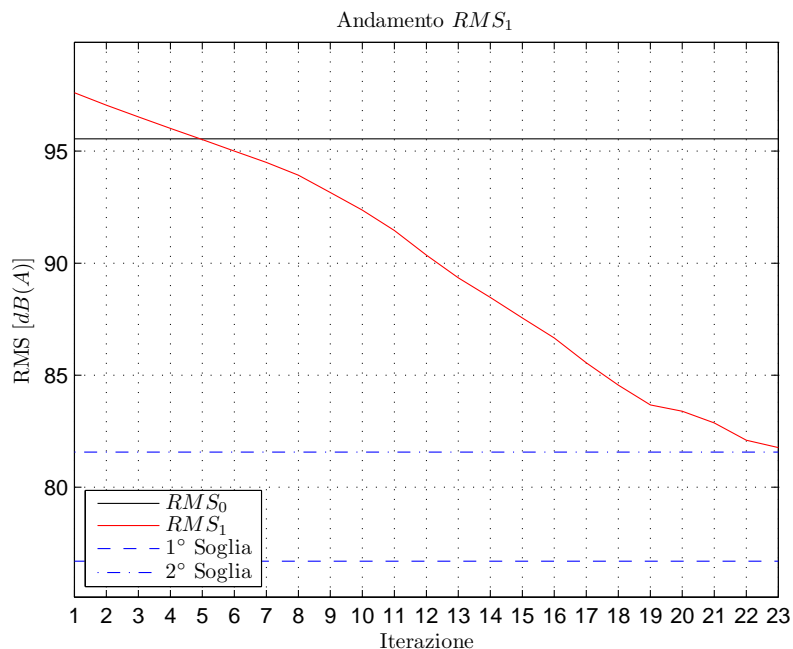


Figura 3.51: Andamento del valore quadratico medio  $RMS_1$ .

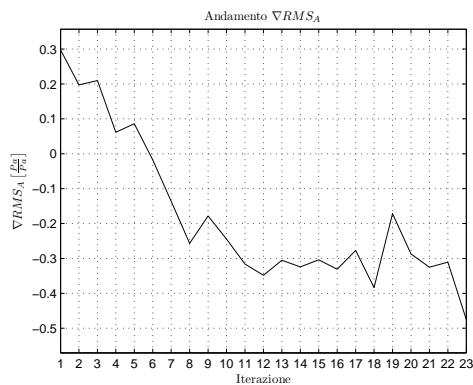


Figura 3.52: Andamento del gradiente  $\nabla RMS_A$  dell'ampiezza del segnale di controllo.

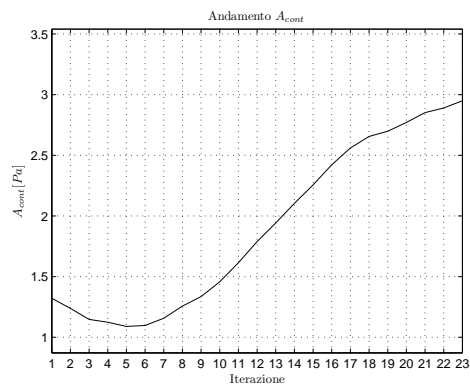


Figura 3.53: Andamento dell'ampiezza  $A_{cont}$  del segnale di controllo.

Nella figura 3.56 è possibile osservare l'andamento dei valori quadratici medi associati alla variazione dei parametri del segnale di controllo, in particolare si nota come gli incrementi  $\Delta A_{cont}$  e  $\Delta \phi_{cont}$  non causino grandi variazioni delle grandezze  $RMS_2$

e  $RMS_3$  rispetto a  $RMS_1$ , sintomo di una scelta dei valori di incremento abbastanza piccoli, che consente di poter approssimare correttamente il calcolo delle derivate.

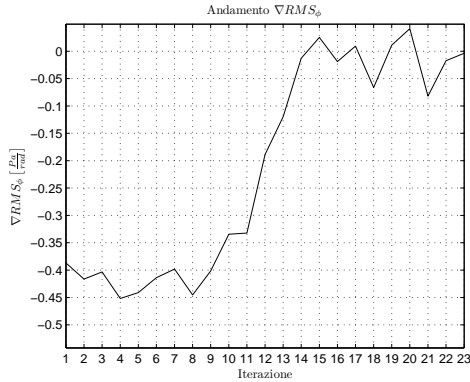


Figura 3.54: *Andamento del gradiente  $\nabla RMS_\phi$  della fase del segnale di controllo.*

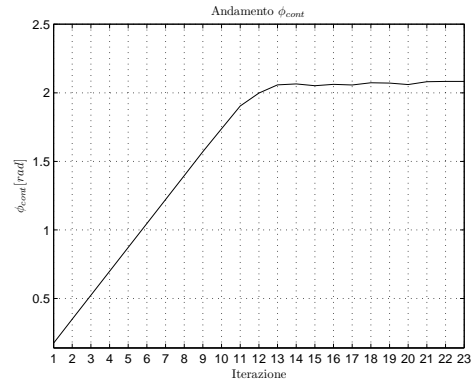


Figura 3.55: *Andamento della fase  $\phi_{cont}$  del segnale di controllo.*

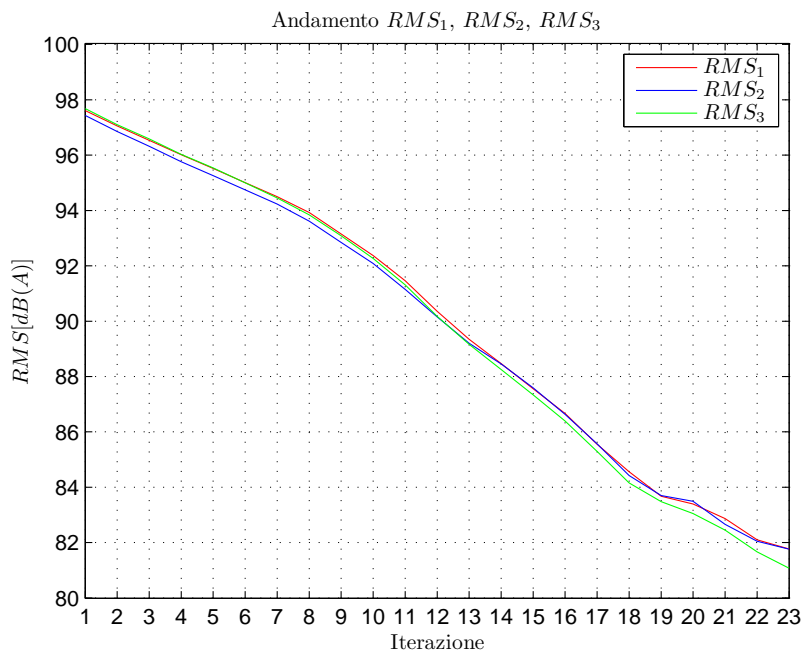


Figura 3.56: *Andamento dei valori quadratici medi  $RMS_1$ ,  $RMS_2$  e  $RMS_3$ .*

Nella figura 3.57 viene riportato l'andamento del parametro  $G_2^k$ : a partire dall'iterazione 17 questo parametro assume valori via via decrescenti all'avvicinarsi dei parametri di controllo alla posizione in cui è minimo il valore quadratico medio  $RMS_1$ . Per la

sceita dei valori di  $G_2^k$  si faccia riferimento all'appendice D.1.2. È inoltre possibile notare che, dall'iterazione 10 all'iterazione 17 e dalla 18 alla 21, il passo  $G_2^k$  si mantiene costante, pertanto non vengono superati i limiti sulla variazione di fase e/o ampiezza e quindi non vi è la necessità di variare il passo  $G_2^k$ .

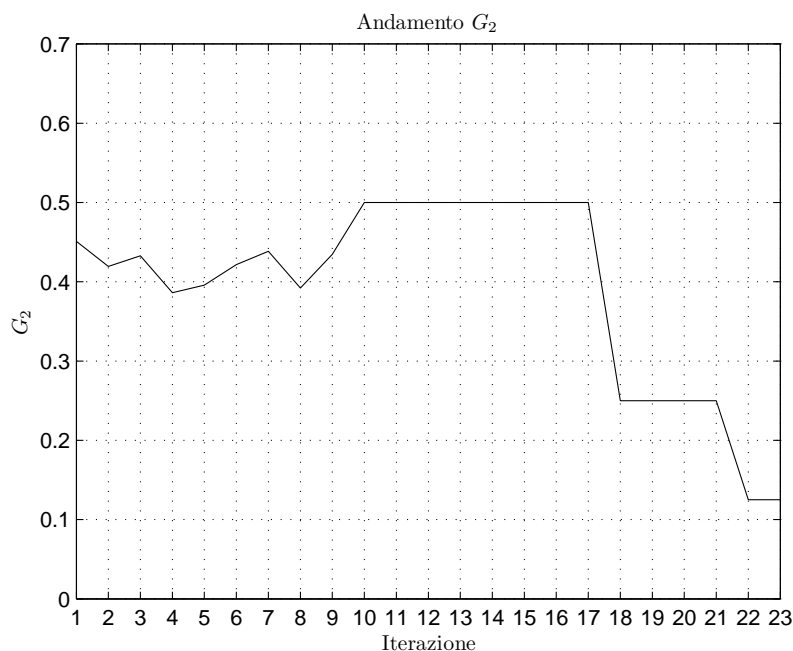


Figura 3.57: *Andamento del guadagno  $G_2^k$ .*

In figura 3.58 viene riportato il grafico riepilogativo dei risultati ottenuti alle diverse frequenze del segnale di rumore, in particolare si può notare che questo metodo risulta efficace in un range di frequenze da 300 a 1200  $Hz$ .

La tabella 3.2 riporta l'elenco delle prove effettuate con gli indici di valutazione delle performance di abbattimento in corrispondenza delle diverse frequenze testate, evidenziando in modo particolare il fattore di abbattimento  $\beta^{RMS}$  calcolato secondo la seguente relazione:

$$\beta^{RMS} = RMS_{OFF} - RMS_{ON}, \quad (3.47)$$

dove  $RMS_{ON}$  e  $RMS_{OFF}$  sono rispettivamente il valore quadratico medio con il controllo acceso ed il valore quadratico medio con il controllo spento, espressi in dB(A). Le grandezze  $RMS_{in}$  e  $RMS_{out}$  riportate in tabella indicano rispettivamente la misurazione all'interno del condotto in corrispondenza della zona in cui si intende abbattere

il rumore e la misurazione esterna al condotto effettuata per mezzo di un microfono aggiuntivo. È possibile notare come il fattore di abbattimento medio all'interno del condotto sia superiore rispetto a quello esterno, questo è dovuto principalmente alla presenza di un rumore continuativo nell'ambiente in cui si sono effettuate le prove e ad un fattore di smorzamento dell'aria non lineare. Infatti all'aumentare della distanza della cassa di rumore dal microfono è facilmente rilevabile la minore potenza richiesta alla cassa di controllo, poiché il segnale della cassa di rumore giunge al microfono con una maggiore attenuazione.

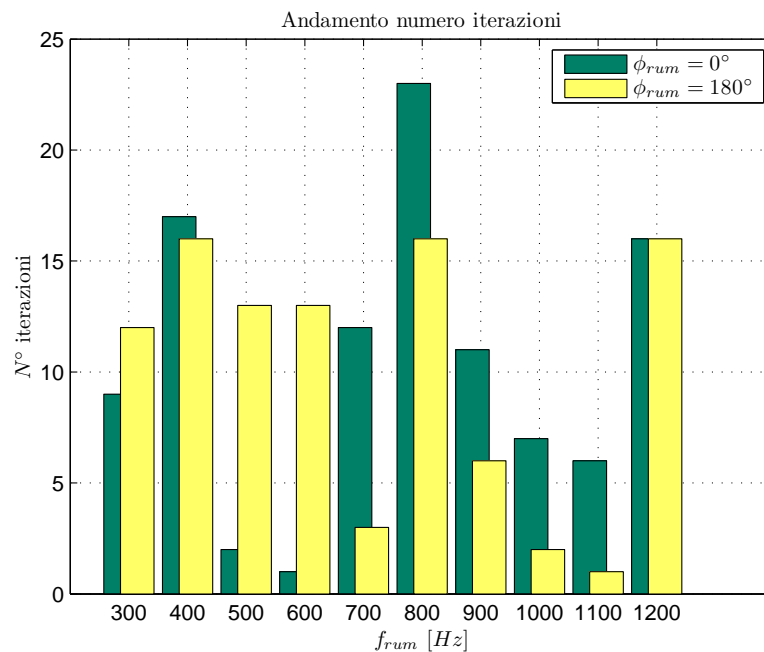


Figura 3.58: Andamento del numero di iterazioni con l'algoritmo basato sul RMS.

L'abbattimento medio complessivo del rumore generato dalla cassa di rumore, con il metodo ARMS, è dell'ordine di 20 dB all'interno del condotto dove è posto il microfono di errore, e di circa 6 dB all'esterno. La differenza dei fattori di abbattimento rilevati all'interno del condotto con quelli rilevati all'esterno è da attribuirsi, come esposto precedentemente, al maggiore smorzamento dell'ambiente sonoro esterno al condotto: l'onda sonora propagandosi in campo aperto diminuisce progressivamente di intensità, da cui derivano i valori minori di  $RMS_{out}$  rispetto a  $RMS_{in}$ .

N° prova	Rumore		Controllo OFF		Controllo ON		Abbattimento	
	$f_{rum}$ [Hz]	$\Phi_{rum}$ [°]	RMS <sub>in</sub> [dB(A)]	RMS <sub>out</sub> [dB(A)]	RMS <sub>in</sub> [dB(A)]	RMS <sub>out</sub> [dB(A)]	$\beta_{in}^{RMS}$ [dB(A)]	$\beta_{out}^{RMS}$ [dB(A)]
1	300	0	94	67	69	62	25	5
2	300	180	94	67	77	62	17	5
3	400	0	106	68	90	62	16	6
4	400	180	106	68	89	61	17	8
5	500	0	97	68	69	65	28	3
6	500	180	97	68	68	64	29	4
7	600	0	95	68	74	63	22	5
8	600	180	95	70	74	66	20	5
9	700	0	96	74	71	64	25	10
10	700	180	96	74	75	67	22	8
11	800	0	96	73	81	65	15	8
12	800	180	96	73	79	69	17	4
13	900	0	95	75	74	65	21	10
14	900	180	95	75	72	67	23	9
15	1000	0	93	76	78	68	15	9
16	1000	180	93	77	77	66	15	11
17	1100	0	88	67	74	64	15	2
18	1100	180	88	67	73	66	16	1
19	1200	0	97	75	70	71	27	4
20	1200	180	97	75	74	70	23	5

Tabella 3.2: *Riepilogo risultati con algoritmo basato sul valore quadratico medio.*

### 3.4.2 Tecnica di controllo AS

Come visto precedentemente per l'algoritmo ARMS, anche in questo caso si è verificato sperimentalmente l'andamento della funzione obiettivo da minimizzare eseguendo una mappatura sperimentale del comportamento di tale funzione (fig. 3.48). Data la forte somiglianza tra il comportamento analitico e quello sperimentale è possibile procedere con lo sviluppo e l'implementazione del codice di controllo secondo lo schema a blocchi riportato in figura 3.59. Questo schema mostra il principio di funzionamento dell'algoritmo di ricerca del minimo secondo la metodologia di minimizzazione del rumore basata sulla misurazione dell'ampiezza del picco rilevato dallo spettro del segnale misurato dal microfono di errore.

Per gli step dell'algoritmo coincidenti a quelli della tecnica esposta precedentemente si faccia riferimento al paragrafo 2.3.2, mentre di seguito vengono presentati solo gli step propri del metodo di minimizzazione basato sull'analisi spettrale del segnale rilevato dal microfono di errore:

4. Ipotizzando note le caratteristiche del rumore in ingresso al sistema, ossia ipotizzando nota la frequenza del segnale di rumore, si determina l'ampiezza del picco dello spettro in corrispondenza della frequenza  $f_{rum}$  secondo quanto mostrato nel paragrafo 2.3.2. Una volta noto l'array di punti  $a_{MIC}$  acquisito, viene effettuato su questo vettore l'analisi spettrale mediante l'algoritmo di Cooley-Tukey. Noto



lo spettro viene effettuata la ricerca del massimo attorno all'indice del vettore corrispondente alla frequenza  $f_{rum}$ . Il medesimo procedimento viene affrontato nei passi 7) e 11) dello schema a blocchi in figura 3.59.

8. In questo passo viene confrontato il valore di ampiezza attuale misurato dallo spettro con il valore dell'ampiezza di soglia di rumore desiderato. Qualora questa condizione risultasse verificata l'algoritmo procede al passo successivo, altrimenti riprende dal passo 5) (par. 3.4.1).

14. Vengono calcolati i nuovi valori da assegnare ai parametri di controllo secondo le relazioni 2.66 e 2.67.

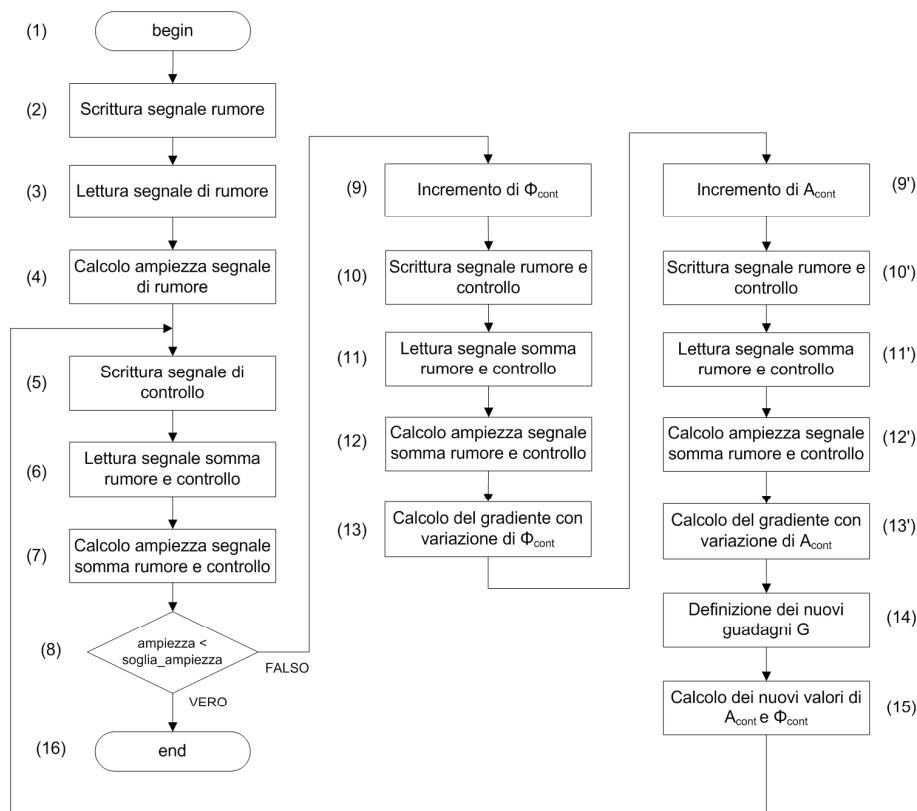


Figura 3.59: Schema a blocchi dell'algoritmo di ricerca del minimo mediante l'analisi spettrale su una frequenza di interesse.

Il numero di parametri da tarare con questa tecnica è lo stesso del metodo basato sulla minimizzazione del valore quadratico medio infatti, rispetto a quest'ultimo, cambia solamente la funzione che viene valutata mentre il processo di minimizzazione è lo stesso. Da un'analisi dell'andamento dell'RMS (fig. 2.16) e dell'ampiezza del picco

dello spettro (fig. 2.22) al variare dell'ampiezza e della fase del segnale di controllo, si nota che i minimi si trovano in corrispondenza dei medesimi valori di ampiezza e fase del segnale di controllo, ma il gradiente di queste funzioni risulta differente poiché la pendenza della funzione da minimizzare con il metodo AS è più elevata. Da queste considerazioni risulta necessario ritrarre i parametri del metodo al fine di garantire un'adeguata stabilità e convergenza.

Nell'appendice D.2 viene riportato il codice sorgente che esegue quanto riportato nello schema a blocchi 3.59 e vengono descritte in dettaglio le funzioni che permettono di eseguire l'implementazione del codice sul controllore NI PXI.

### Risultati sperimentali

Come già visto nel caso dell'algoritmo ARMS, attraverso il monitoraggio di alcune importanti grandezze del codice è possibile analizzare il comportamento dell'algoritmo alle diverse frequenze del segnale di rumore e apportare al codice di controllo una serie di modifiche sulle variabili dell'algoritmo, garantendo allo stesso un migliore comportamento in termini di convergenza e di tempo di raggiungimento della soglia di rumore desiderata. Le grandezze, monitorate in funzione del passo di iterazione, sono le seguenti:

- Valore del picco  $Pf_1$  dello spettro del segnale del microfono di errore costituito dalla somma del segnale di rumore e di controllo. Si mette in evidenza l'andamento di questa grandezza rispetto al valore del picco spettrale  $Pf_0$ , misurato quando nel sistema vi è la presenza del solo segnale di rumore. Il valore di soglia relativo all'ampiezza del picco desiderata, per ritenere il raggiungimento da parte dell'algoritmo di un buon abbattimento del rumore, è stata fissata in funzione della frequenza, come mostrato precedente nell'algoritmo ARMS (fig. 3.50). Qualora il numero di iterazioni risulti troppo elevato si ritiene sufficiente il raggiungimento di una soglia di ampiezza meno restrittiva e pari al 20 % del valore di ampiezza iniziale  $Pf_0$ .
- Gradiente  $\nabla Pf_A$  dell'ampiezza del segnale di controllo.
- Gradiente  $\nabla Pf_\phi$  della fase del segnale di controllo.
- Valori delle ampiezze dei picchi associate alle variazioni dei parametri delle variabili del segnale di controllo. In particolare  $Pf_2$  è il valore quadratico medio

associato all'incremento  $\Delta\phi_{cont}$  della fase del segnale di controllo, mentre  $Pf_3$  è legato all'incremento  $\Delta A_{cont}$  dell'ampiezza dello stesso segnale di controllo.

- Ampiezza  $A_{cont}$  del segnale di controllo.
- Fase  $\phi_{cont}$  del segnale di controllo.
- Guadagno  $G_2^k$  associato ai nuovi valori delle variabili  $A_{cont}$  e  $\phi_{cont}$ .

Si riportano i risultati sperimentali ottenuti abbattendo un segnale di rumore alla frequenza  $f_{rum} = 800 \text{ Hz}$ . Nella figura 3.60 si mette in evidenza l'andamento dell'ampiezza del picco  $Pf_1$  in funzione del passo di iterazione: si nota che questo valore decresce rispetto al valore iniziale  $Pf_0$ , all'aumentare del passo di iterazione e raggiunge la soglia desiderata di rumore in 17 iterazioni. A partire dalla settima iterazione il valore di  $Pf_1$  ha un andamento oscillatorio, la cui analisi verrà proposta in seguito analizzando gli andamenti dei gradienti dei parametri di controllo.

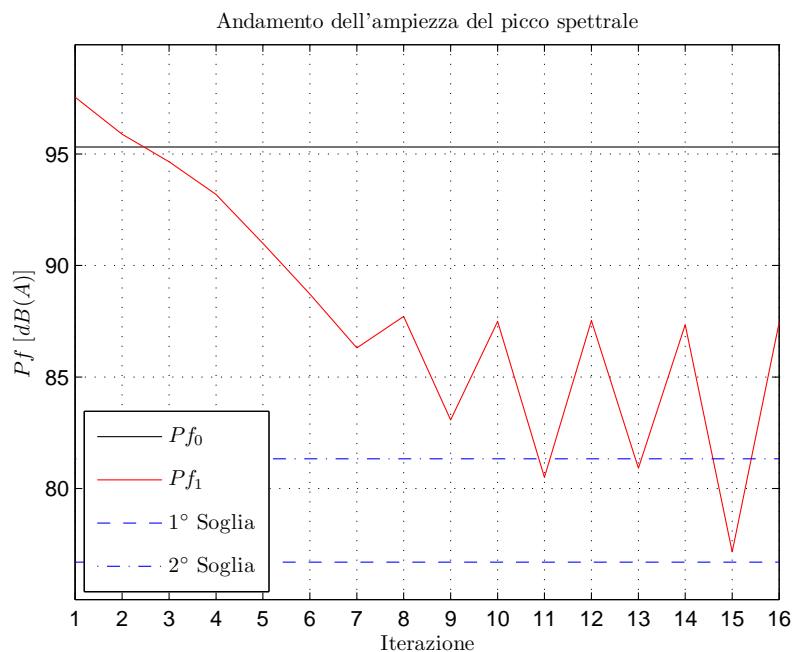


Figura 3.60: *Andamento dell'ampiezza del picco spettrale con algoritmo basato sull'analisi spettrale.*

Nelle figure 3.61 e 3.63 viene riportato l'andamento dei gradienti dei parametri di controllo  $A_{cont}$  e  $\phi_{cont}$ . È possibile notare che già a partire dalla settima iterazione l'algoritmo si assesta attorno alla posizione del minimo infatti i valori dei gradienti

iniziano ad oscillare attorno allo zero. Questo risulta ancor più evidente nelle figure 3.62 e 3.64 che mostrano l'andamento della fase  $\phi_{cont}$  e dell'ampiezza  $A_{cont}$  del segnale di controllo. Nella figura 3.65 viene riportato l'andamento dei valori dei picchi spettrali associato alla variazione dei parametri del segnale di controllo, in particolare si nota che la scelta iniziale degli incrementi  $\Delta A_{cont}$  e  $\Delta \phi_{cont}$  risulta adeguata fino alla settima iterazione in quanto vengono rispettate le seguenti condizioni:  $|Pf_1 - Pf_2| \ll |Pf_1|$  e  $|Pf_1 - Pf_3| \ll |Pf_1|$ . Tuttavia, a partire dalla settima iterazione queste condizioni, richieste per una corretta approssimazione della derivata non vengono più rispettate. Il sistema diviene così più sensibile alle variazioni di entrambi i parametri di controllo e questo causa, seppur modeste, delle variazioni rispetto all'andamento generale di decadimento dell'ampiezza del picco spettrale  $Pf_1$ . La taratura del metodo è stata effettuata, come mostrato nell'appendice D.1.2, su determinati range di frequenze pertanto può capitare che i parametri del metodo di minimizzazione non siano ovunque ottimali.

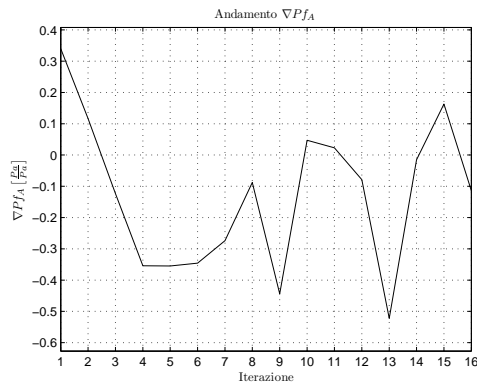


Figura 3.61: *Andamento del gradiente  $\nabla Pf_A$  dell'ampiezza del segnale di controllo con algoritmo basato sull'analisi spettrale.*

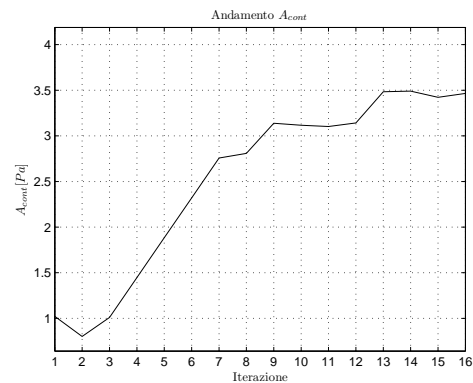


Figura 3.62: *Andamento dell'ampiezza  $A_{cont}$  del segnale di controllo con algoritmo basato sull'analisi spettrale.*

Nella figura 3.66 viene riportato l'andamento del guadagno  $G_2^k$ ; questo parametro assume valori via via decrescenti all'avvicinarsi dei parametri di controllo alla posizione in cui è minimo il valore del picco  $Pf_1$  in modo da garantire un'adeguata stabilità all'algoritmo. La diminuzione del parametro  $G_2^k$ , per assicurare la convergenza del metodo di minimizzazione, risulta molto evidente a partire dalla settima iterazione dove i gradienti dell'ampiezza del picco rispetto ai parametri di controllo iniziano ad oscillare attorno allo zero. Per la scelta dei valori di  $G_2^k$  si faccia riferimento all'appendice D.2.2.

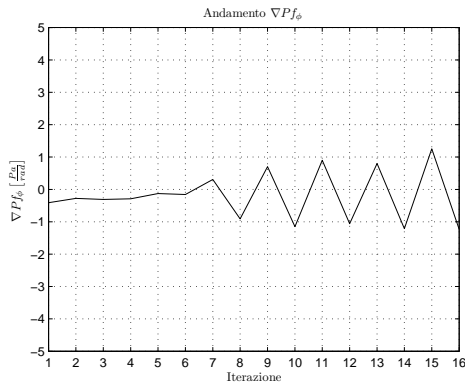


Figura 3.63: *Andamento del gradiente  $\nabla Pf_\phi$  della fase del segnale di controllo con algoritmo basato sull'analisi spettrale.*

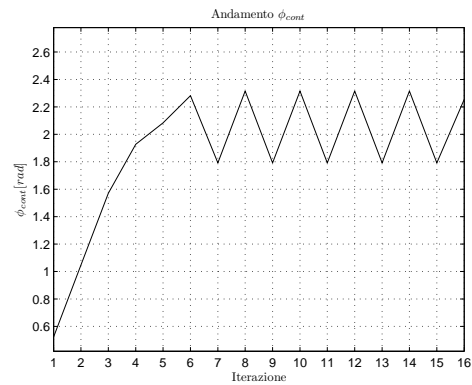


Figura 3.64: *Andamento della fase  $\phi_{cont}$  del segnale di controllo con algoritmo basato sull'analisi spettrale.*

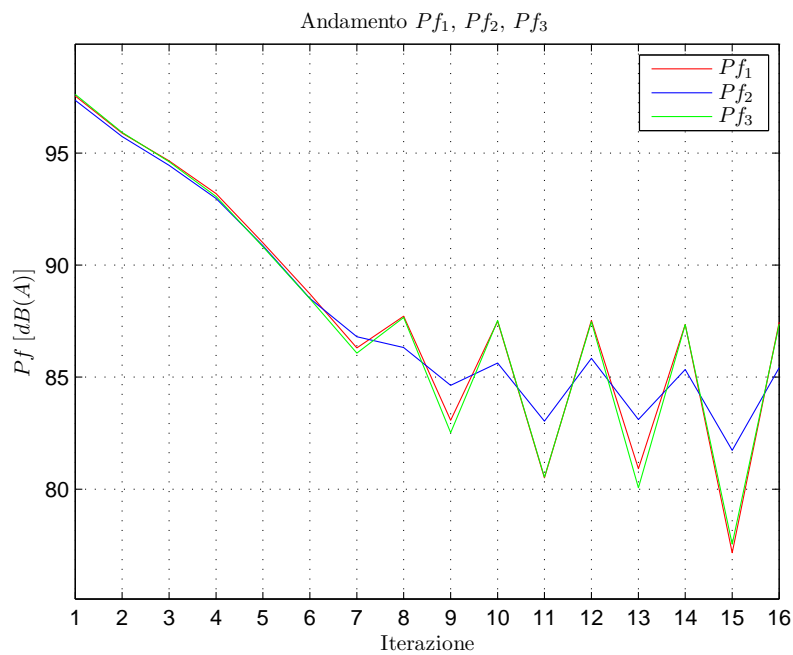


Figura 3.65: *Andamento di  $Pf_1$ ,  $Pf_2$  e  $Pf_3$  con algoritmo basato sull'analisi spettrale*

La figura 3.67 mostra un grafico riepilogativo dei risultati ottenuti alle diverse frequenze del segnale di rumore, da cui si evidenzia l'efficace funzionamento in un range di frequenze da 300 a 1200  $Hz$ . Da un confronto delle figure 3.67 e 3.58 si osserva che il metodo basato sull'analisi spettrale richiede un numero di iterazioni, per il raggiungimento del minimo, pressoché costante alle diverse frequenze, questo è legato al fatto

che questo metodo è meno sensibile ai rumori presenti nell'ambiente.

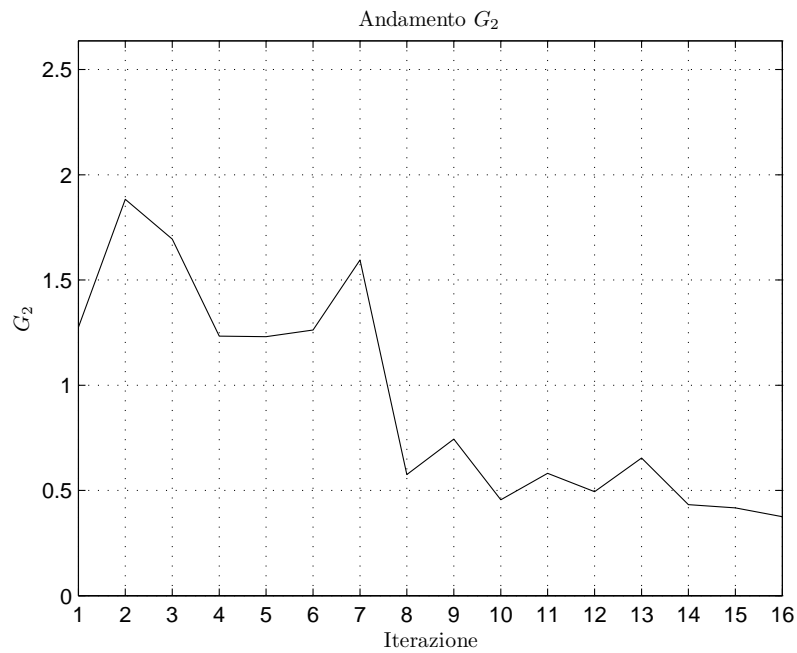


Figura 3.66: Andamento del guadagno  $G_2^k$  con algoritmo basato sull'analisi spettrale.

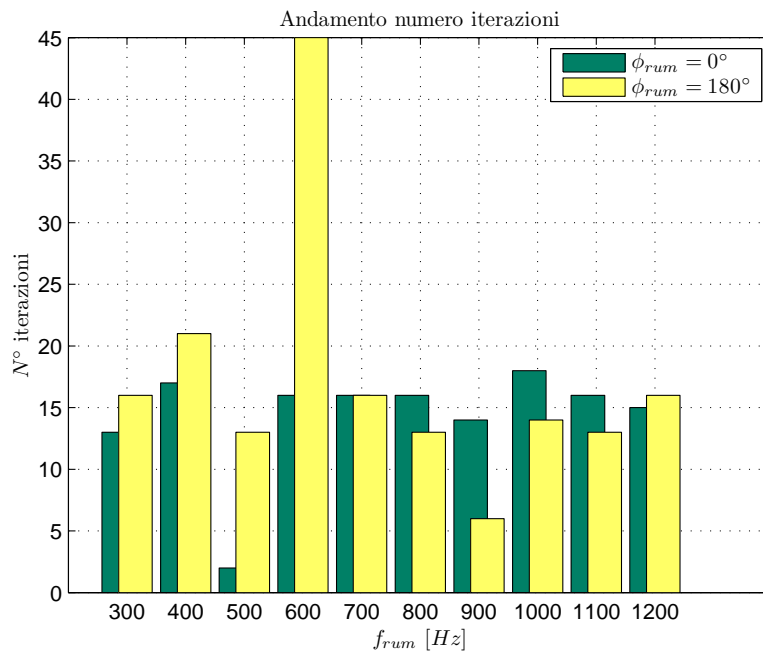


Figura 3.67: Andamento del numero di iterazioni con algoritmo basato sull'analisi spettrale.

Nella tabella 3.3 vengono mostrati i risultati riepilogativi delle prove effettuate. Rispetto al metodo di minimizzazione del valore quadratico medio, dove si pone attenzione al fattore di abbattimento medio, in questo caso viene considerato un fattore di abbattimento  $\beta^{Pf}$  correlato all'ampiezza del picco in corrispondenza della frequenza che si intende abbattere, secondo la seguente relazione:

$$\beta^{Pf} = Pf_{OFF} - Pf_{ON}, \quad (3.48)$$

dove  $Pf_{OFF}$  e  $Pf_{ON}$  sono rispettivamente i valori delle ampiezze dei picchi spettrali, espressi in  $dB(A)$ , in assenza (OFF) e in presenza (ON) del segnale della cassa di controllo. In termini di fattore di abbattimento sull' $RMS$  si ottengono mediamente gli stessi risultati del metodo precedente. L'abbattimento medio complessivo rilevato sul valore RMS è di circa 21  $dB$ , mentre l'attenuazione del picco spettrale è mediamente pari a 11  $dB$ . I maggiori vantaggi del metodo sono legati alla sua quasi totale insensibilità ai rumori presenti nell'ambiente esterno.

N° prova	Rumore		Controllo OFF			Controllo ON			Abbattimento		
	$f_{rum}$ [Hz]	$\Phi_{rum}$ [°]	$RMS_{in}$ [dB(A)]	$RMS_{out}$ [dB(A)]	$Pf_{out}$ [dB(A)]	$RMS_{in}$ [dB(A)]	$RMS_{out}$ [dB(A)]	$Pf_{out}$ [dB(A)]	$\beta_{in}^{RMS}$ [dB(A)]	$\beta_{out}^{RMS}$ [dB(A)]	$\beta_{out}^{Pf}$ [dB(A)]
1	300	0	93	65	64	66	61	57	27	4	8
2	300	180	93	66	64	77	62	59	15	4	5
3	400	0	104	66	65	86	61	43	18	5	22
4	400	180	104	66	65	84	61	51	20	5	14
5	500	0	97	68	67	70	64	59	27	3	7
6	500	180	97	68	67	69	64	59	27	4	8
7	600	0	102	75	75	87	70	70	15	6	6
8	600	180	102	76	75	88	70	68	14	6	7
9	700	0	104	81	81	87	71	70	17	10	12
10	700	180	104	81	81	77	69	65	27	13	16
11	800	0	96	75	75	73	72	72	23	3	3
12	800	180	96	75	75	77	72	71	19	3	3
13	900	0	95	75	74	72	67	62	23	8	12
14	900	180	95	74	74	74	66	63	21	9	12
15	1000	0	98	81	82	82	70	69	16	11	13
16	1000	180	98	81	82	75	70	67	23	11	15
17	1100	0	94	70	69	74	64	58	20	6	11
18	1100	180	94	70	69	74	68	58	20	2	11
19	1200	0	96	74	74	78	66	59	18	8	14
20	1200	180	96	74	74	71	68	63	25	6	11

Tabella 3.3: Riepilogo risultati con algoritmo basato sulla minimizzazione del picco spettrale.

### 3.4.3 Tecnica di controllo MFS

Secondo quanto mostrato nel paragrafo 2.3.3 è possibile utilizzare la tecnica precedente in modo ricorsivo al fine di minimizzare rumori multi-armonici. La tecnica è del tutto simile a quella dell'algorithm AS vista nel paragrafo 3.4.2 dove lo schema a blocchi dell'algorithm di controllo viene modificato secondo quanto mostrato in figura 3.68.

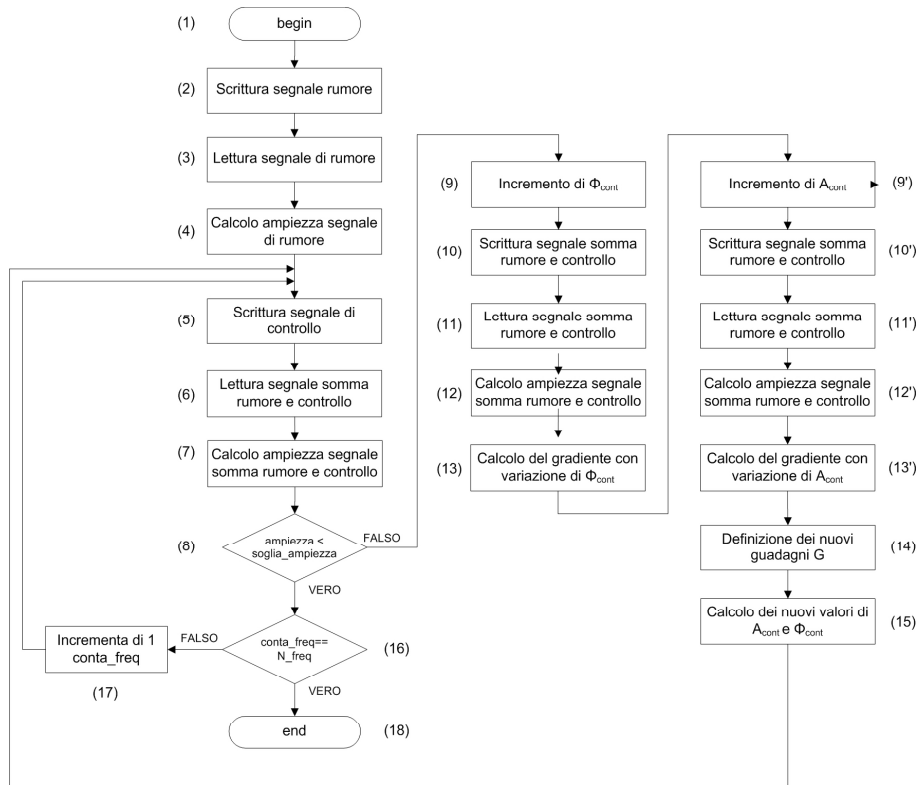


Figura 3.68: Schema a blocchi dell'algorithm di ricerca del minimo nel caso di rumore con più armoniche.

Per quanto riguarda i blocchi da 1) a 15) (fig. 3.68) si faccia riferimento a quanto riportato nel paragrafo 3.4.2: a differenza dello schema a blocchi dell'algorithm AS, in questo caso compaiono i blocchi 16) e 17) che permettono di richiamare ricorsivamente il metodo per ciascuna frequenza contenuta nel segnale di rumore. In particolare si ha:

16. In questo step dell'algorithm viene confrontato il numero di frequenze che sono state analizzate con il numero totale delle frequenze del rumore che si desidera abbattere. Se il numero di frequenze analizzato è inferiore al numero di frequenze del segnale di rumore, l'algorithm procede con lo step 17) dove viene incrementato il numero di frequenze analizzate e chiamato lo step 5), in caso contrario, se il



rumore è stato minimizzato a tutte le frequenze di interesse, l'algoritmo procede al termine mantenendo costanti tutti i parametri di controllo determinati in precedenza.

17. Viene incrementata una variabile contatore che permette di considerare di volta in volta le variabili relative ad una particolare frequenza tra quelle in analisi e per le quali deve essere ancora effettuato l'abbattimento della pressione sonora.

Nell'appendice D.3 viene riportato il codice sorgente che esegue quanto riportato nello schema a blocchi 3.68 e vengono descritte tutte le funzioni che permettono di eseguire l'implementazione del codice sul controllore NI PXI.

### Risultati sperimentali

Come già visto nell'analisi degli algoritmi precedenti, attraverso il monitoraggio di alcune importanti grandezze del codice, viene analizzato il comportamento dell'algoritmo alle diverse frequenze del segnale di rumore e vengono apportate al codice di controllo una serie di modifiche sulle variabili dell'algoritmo, al fine di garantire allo stesso un migliore comportamento in termini di convergenza e di tempo di raggiungimento della soglia di rumore desiderata. Le grandezze analizzate dal codice di monitoraggio sono le stesse del metodo AS riportate nel paragrafo 3.4.2. Di seguito si riportano i risultati sperimentali ottenuti abbattendo un segnale di rumore composto da tre differenti armoniche date da tre segnali sinusoidali con frequenze pari a 400, 800 e 900  $Hz$  e fasi pari rispettivamente a  $90^\circ$ ,  $180^\circ$  e  $250^\circ$ .

Nella figura 3.69 si riporta l'andamento delle ampiezze dei picchi  $Pf_1$  in funzione del passo di iterazione per le diverse frequenze del segnale di rumore: si nota che questi valori decrescono rispetto ai loro valori iniziali  $Pf_0$  all'aumentare del passo di iterazione e raggiungono la soglia desiderata dopo un certo numero di iterazioni. È evidente che il tempo complessivo (o il numero complessivo di iterazioni) per il raggiungimento dei minimi alle diverse frequenze è dato dalla somma dei tempi (o delle iterazioni) sulle singole frequenze:

$$N^\circ \text{iterazioni}_{400 \text{ Hz}} = 19; \quad (3.49)$$

$$N^\circ \text{iterazioni}_{800 \text{ Hz}} = 9; \quad (3.50)$$

$$N^\circ \text{iterazioni}_{900 \text{ Hz}} = 10; \quad (3.51)$$

quindi, in questo caso è possibile determinare il segnale di controllo dopo un totale di 38 iterazioni.

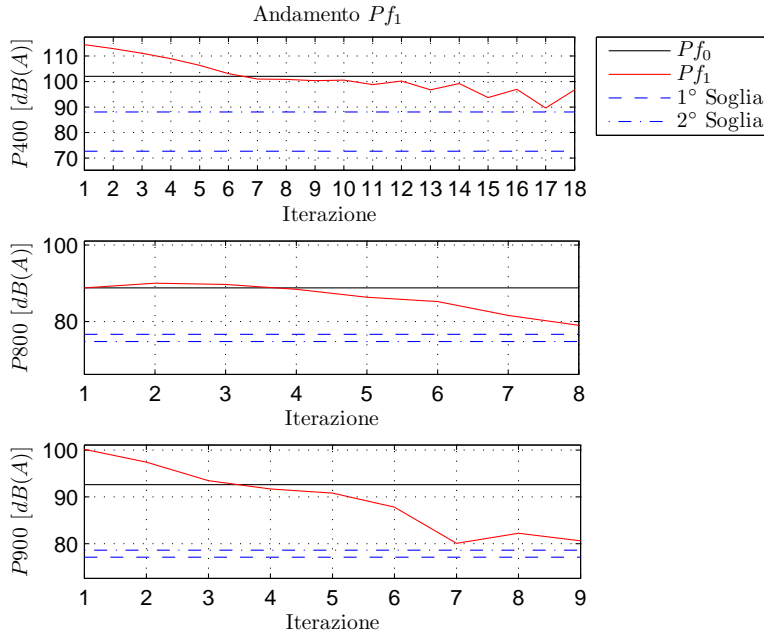


Figura 3.69: *Andamento delle ampiezze dei picchi spettrali con algoritmo ricorsivo basato sull'analisi spettrale.*

Nelle figure 3.70 e 3.72 viene riportato l'andamento dei gradienti dei parametri di controllo  $A_{cont}$  e  $\phi_{cont}$  alle diverse frequenze del segnale di rumore. Per quanto riguarda i gradienti dei valori dei picchi rispetto alle ampiezze del segnale di controllo è possibile notare che alla frequenza di 400 Hz il parametro di controllo inizia ad oscillare attorno alla soluzione dopo 8 iterazioni, alla frequenza di 800 Hz oscilla attorno alla soluzione già dalla prima iterazione, mentre alla frequenza di 900 Hz oscilla a partire dalla terza iterazione. Questa situazione potrebbe essere un caso abbastanza fortuito in cui il segnale di controllo iniziale è già vicino alla soluzione tale da minimizzare il rumore secondo le soglie desiderate.

Osservando i gradienti dei valori dei picchi rispetto alle fasi del segnale di controllo risulta necessario un adeguato numero di iterazioni prima che l'algoritmo si assesti attorno alle soluzioni del problema di minimizzazione. Alla frequenza di 400 Hz la fase del segnale di controllo non raggiunge una posizione di minimo della funzione, ma è comunque tale da soddisfare la soglia di rumore desiderata a quella frequenza. Infatti, osservando la figura 3.72 alla frequenza di 400 Hz, il gradiente continua a rimanere

positivo ed è tale da assegnare un andamento monotono decrescente al valore della fase del segnale di controllo a quella frequenza.

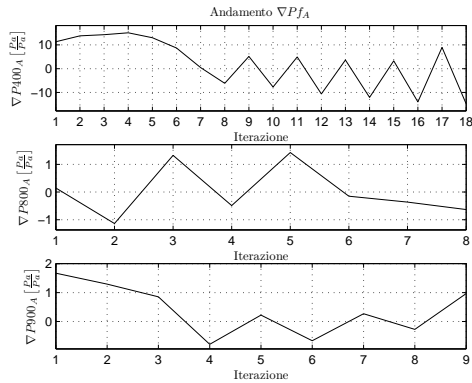


Figura 3.70: *Andamento del gradiente  $\nabla P f_A$  delle ampiezze del segnale di controllo con algoritmo ricorsivo basato sull'analisi spettrale.*

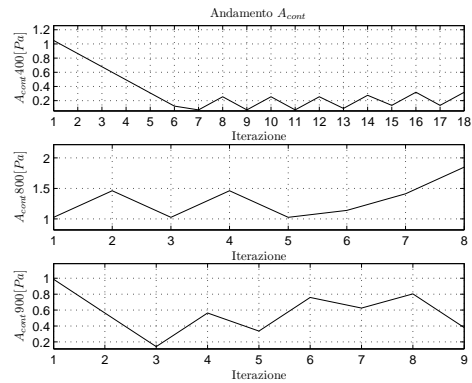


Figura 3.71: *Andamento delle ampiezze  $A_{cont}$  del segnale di controllo con algoritmo ricorsivo basato sull'analisi spettrale.*

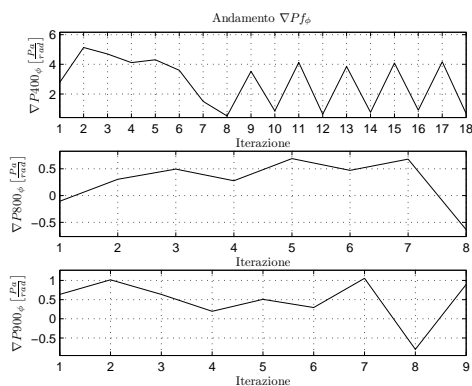


Figura 3.72: *Andamento del gradiente  $\nabla P f_\phi$  delle fasi del segnale di controllo con algoritmo ricorsivo basato sull'analisi spettrale.*

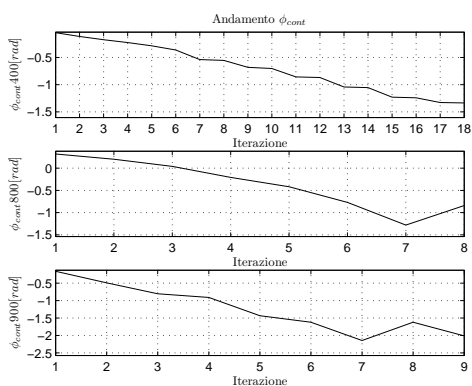


Figura 3.73: *Andamento delle fasi  $\phi_{cont}$  del segnale di controllo con algoritmo ricorsivo basato sull'analisi spettrale.*

Questo andamento è sintomo di una scelta corretta dei parametri dell'algoritmo di minimizzazione. Alle frequenze di 800 e 900 Hz, l'algoritmo è in grado di localizzare la posizione dei minimi dopo sole 7 iterazioni infatti le fasi del segnale di controllo si stabilizzano attorno alla soluzione dove sono minimi i valori dei picchi spettrali a quelle frequenze (fig. 3.73).

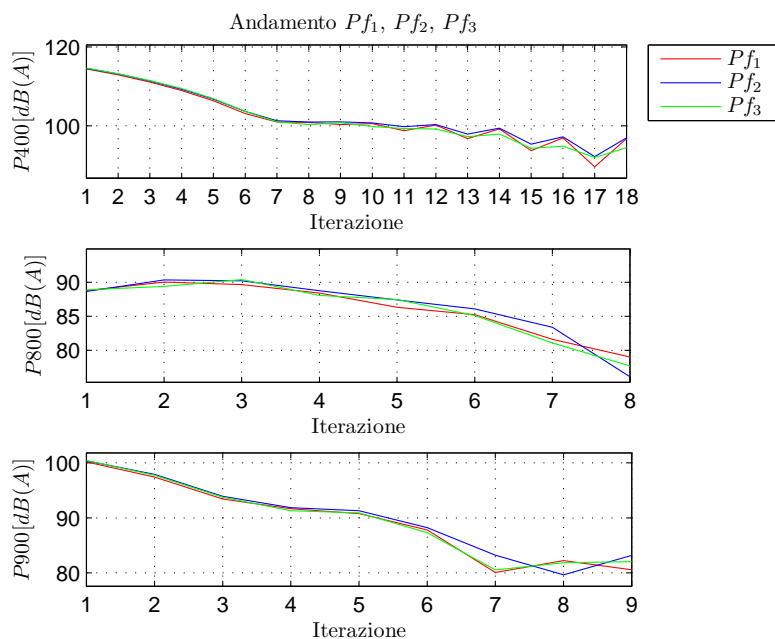


Figura 3.74: Andamento delle ampiezze dei picchi spettrali  $P_{f_1}$ ,  $P_{f_2}$  e  $P_{f_3}$  con algoritmo ricorsivo basato sull'analisi spettrale.

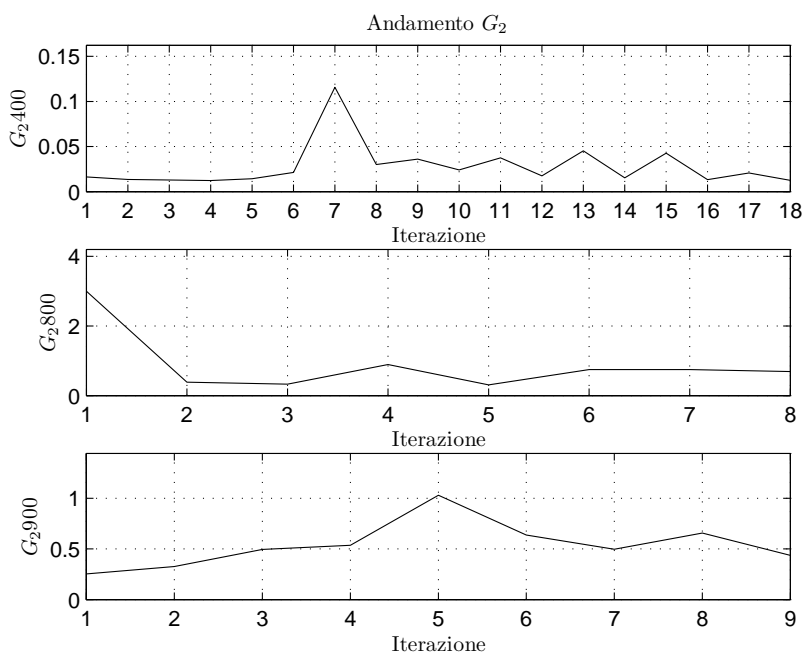


Figura 3.75: Andamento dei guadagni  $G_2^k$  con algoritmo ricorsivo basato sull'analisi spettrale.

La figura 3.74 mostra l'andamento dei valori dei picchi spettrali associati alla variazione dei parametri del segnale di controllo in corrispondenza delle frequenze di cui è composto il segnale di rumore: si nota che gli incrementi scelti per i parametri di controllo  $\Delta A_{cont}$  e  $\Delta \phi_{cont}$  siano tali da non causare notevoli variazioni rispetto all'andamento generale di decadimento delle ampiezze dei picchi spettrali  $Pf_1$ .

La figura 3.75 riporta gli andamenti dei guadagni  $G_2^k$ : i valori tendono a stabilizzarsi dopo un discreto numero di passi, sintomo che il loro valore assicura una buona convergenza e stabilità al metodo. Per la scelta dei valori di  $G_2^k$  si faccia riferimento all'appendice D.2.2.

N° prova	Rumore		Controllo OFF	Controllo ON	Abbattimento
	$f_{rum}$ [Hz]	$\Phi_{rum}$ [°]	$Pf_{out}$ [dB(A)]	$Pf_{out}$ [dB(A)]	$\beta_{out}^{Pf}$ [dB(A)]
1	1000	0	59	46	13
2	300	0	62	49	13
3	300	0	62	51	11
	800	0	65	53	12
4	1200	0	67	40	27
	700	0	64	54	10
	600	0	57	37	20
5	400	90	63	51	13
	800	180	66	64	2
	900	250	72	49	23
6	300	0	56	48	8
	500	30	49	43	6
	700	80	48	40	8
	1100	120	60	50	10
7	300	0	55	50	5
	400	180	58	48	10
	600	0	60	47	13
	1000	180	56	46	10

Tabella 3.4: *Riepilogo risultati con algoritmo basato sulla minimizzazione del picco spettrale con minimizzazione seriale.*

Poiché il metodo attualmente analizzato è basato sulla stessa logica di funzionamento dell'algoritmo AS, considerando singolarmente le frequenze controllate ci si aspetterebbe di ottenere i medesimi risultati in termini di fattore di abbattimento e numero di iterazioni tra i due metodi. Tuttavia, col metodo MFS sembrerebbe necessario un numero inferiore di iterazioni sulle singole frequenze per il raggiungimento delle condizioni di minimo della funzione. Questo aspetto può essere dovuto ad una migliore taratura dell'algoritmo nel suo insieme, ovvero la taratura dei guadagni e delle soglie.

Nella tabella 3.4 vengono mostrati i risultati riepilogativi delle prove effettuate. L'abbattimento medio complessivo, rilevato sull'ampiezza del picco spettrale, è pari a circa 12 dB rispetto al sistema con controllore non attivato. Uno dei maggiori vantaggi di questo metodo, oltre al fatto di minimizzare più componenti del segnale di rumore,

è quello di essere completamente insensibile ai rumori presenti nell'ambiente esterno, a meno di rumori esterni con frequenze esattamente pari alle armoniche presenti nel segnale di rumore.

### 3.4.4 Tecnica di controllo MFP

L'algoritmo per la ricerca dei minimi delle ampiezze dei picchi rilevati dallo spettro, nonché degli stessi spettri, è basato sullo schema a blocchi riportato in figura 3.76.

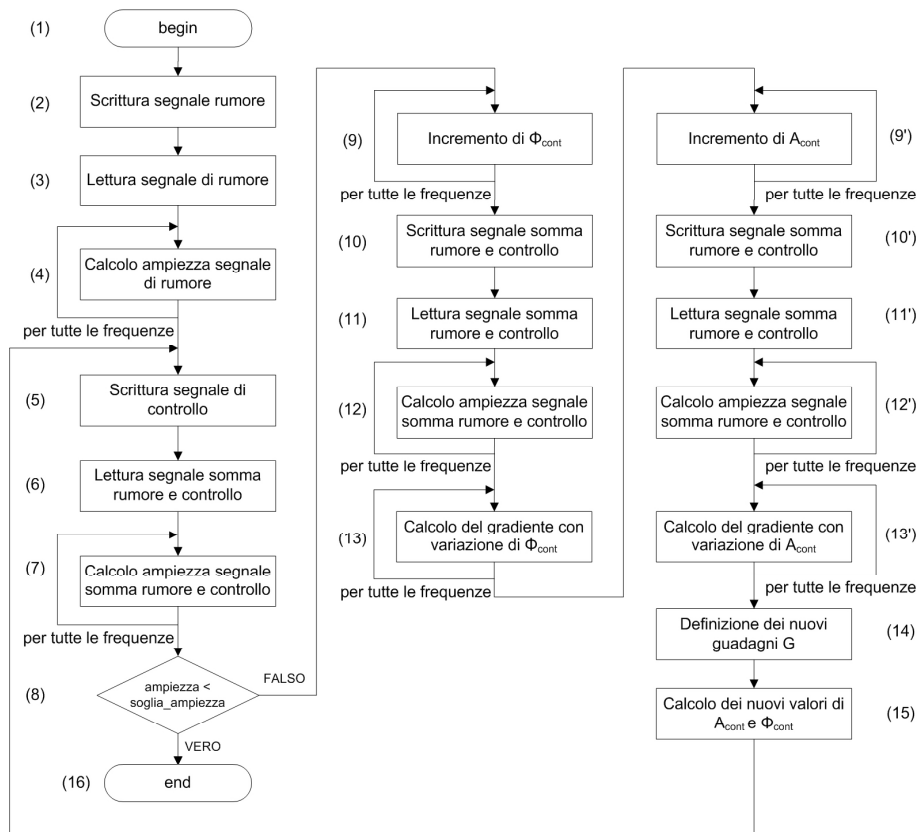


Figura 3.76: Schema a blocchi dell'algoritmo di ricerca del minimo basato sull'analisi spettrale in parallelo.

Diversi step dell'algoritmo coincidono a quelli visti negli a blocchi precedenti, in particolare avendo mantenuto gli stessi riferimenti tra gli schemi, vengono di seguito discussi solo i passi propri del metodo di minimizzazione multi-frequenza con abbattimento parallelo:

4. Ipotizzando note le caratteristiche del rumore in ingresso al sistema si determinano le ampiezze dei picchi dello spettro in corrispondenza delle  $N$  frequenze

$f_{rum}^i, i = 1, \dots, N$  secondo quanto mostrato nel paragrafo 2.3.4. Una volta noto l'array di punti  $a_{MIC}$  viene effettuato su questo vettore il calcolo dello spettro sfruttando l'algoritmo di Cooley-Tukey e successivamente, secondo un criterio ricorsivo, viene effettuata la ricerca del massimo attorno all'indice del vettore corrispondente alla frequenza  $f_{rum}^i$ . Il medesimo procedimento viene affrontato nei passi 7) e 11) dello schema a blocchi.

8. In questo passo vengono confrontati i valori delle nuove ampiezze dei picchi con il valore di soglia fissata al punto 1), secondo la relazione:

$$P_{f_i} \leq P_{soglia}, \quad i = 1, \dots, N. \quad (3.52)$$

Qualora la disuguaglianza 3.52 risultasse verificata, ad esempio in corrispondenza della frequenza  $f_{\bar{i}}$ , viene aggiornato un *flag* che indica il raggiungimento della condizione di minimo a quella specifica frequenza, di conseguenza l'algoritmo termina la procedura di minimizzazione relativa alla frequenza  $f_{\bar{i}}$ . Se la disuguaglianza è verificata per tutte le frequenze di interesse, l'algoritmo di minimizzazione procede al termine e si porta al passo 16), altrimenti continua dal passo 9) per le sole frequenze che devono ancora essere minimizzate.

9. In questa fase vengono forniti gli incrementi alle variabili di controllo mediante un approccio iterativo. Per la valutazione dei gradienti è infatti necessario, secondo le relazioni 2.62 e 2.63, calcolare le ampiezze dei picchi dello spettro dopo aver fornito una variazione ai parametri. I valori scelti per gli incrementi  $\Delta A_{cont}$  e  $\Delta \phi_{cont}$  delle variabili di controllo sono i medesimi per tutte le frequenze di interesse e vengono impostati nella prima fase dell'algoritmo. Dalla fase 9) e 9') risulta evidente la peculiarità di questo metodo, infatti vengono forniti gli incrementi alle variabili di controllo in corrispondenza di tutte le frequenze che devono ancora essere abbattute; questo risulta fondamentale per snellire il carico computazionale richiesto al processore. Come in tutti i metodi visti precedentemente, vengono dapprima forniti gli incrementi delle fasi del segnale di controllo e calcolati i gradienti sulle variazioni di fase e successivamente, a partire dalle stesse condizioni precedenti, vengono effettuati gli incrementi delle ampiezze del segnale di controllo e calcolati i gradienti sulle variazioni di ampiezza.
10. Viene definito e quindi generato mediante la cassa di controllo il segnale dato dalle variazioni dei parametri di controllo le cui caratteristiche sono:

$$A_{cont}^{i temp} = A_{cont}^{k,i} + \Delta A_{cont}, \quad i = 1, \dots, N; \quad (3.53)$$

$$\phi_{cont}^{i temp} = \phi_{cont}^{k,i} + \Delta \phi_{cont}, \quad i = 1, \dots, N. \quad (3.54)$$

L'apice *temp* indica che questi valori sono dei valori temporanei infatti questi non devono essere confusi con i valori dei parametri di controllo che vengono determinati con le relazioni 2.62 e 2.63.

13. Con una tecnica ricorsiva vengono calcolati i gradienti in corrispondenza di tutte le frequenze di interesse in cui deve ancora essere minimizzato il rumore secondo le relazioni 2.62 e 2.63.
14. In questo step vengono rivalutati i valori dei guadagni  $G^{k,i}$  qualora la variazione delle grandezze di controllo imposte dal gradiente,  $G_1^{k,i} \nabla P_{f_i}^A$  e  $G_1^{k,i} \nabla P_{f_i}^\phi$ , superino i limiti  $A_{SAT}^i$  e  $\phi_{SAT}^i$  consentiti. I guadagni vengono rivalutati secondo i criteri esposti nell'appendice D.2.2. Dopo la rivalutazione del guadagno  $G_2^{k,i}$  a seguito del superamento di uno dei limiti consentiti sui parametri di controllo, si procede a valutare l'eventuale oscillazione della soluzione attorno al punto di minimo della funzione secondo i medesimi criteri esposti nell'appendice D.2.2.
15. Vengono calcolati i nuovi valori da assegnare ai parametri di controllo secondo le seguenti relazioni:

$$A_{cont}^{(k+1),i} = A_{cont}^{k,i} - G_2^{k,i} \nabla P_{f_i}^A, \quad i = 1, \dots, 2; \quad (3.55)$$

$$\phi_{cont}^{(k+1),i} = \phi_{cont}^{k,i} - G_2^{k,i} \nabla P_{f_i}^\phi, \quad i = 1, \dots, 2. \quad (3.56)$$

Nell'appendice D.4 viene riportato il codice sorgente che esegue quanto riportato nello schema a blocchi 3.76 e vengono descritte in dettaglio tutte le funzioni che permettono di eseguire l'implementazione del codice sul controllore NI PXI.

### Risultati sperimentali

Come già visto nell'analisi degli algoritmi precedenti, attraverso il monitoraggio di alcune importanti grandezze del codice, viene analizzato il comportamento dell'algoritmo alle diverse frequenze del segnale di rumore e vengono apportate al codice di controllo una serie di modifiche sulle variabili dell'algoritmo, al fine di garantire allo stesso un migliore comportamento in termini di convergenza e di tempo di raggiungimento della



soglia di rumore desiderata. La minimizzazione dei picchi avviene in questo caso in modo simultaneo alle diverse frequenze di cui è composto il segnale di rumore. Le grandezze analizzate dal codice di monitoraggio sono le stesse riportate nei due metodi precedenti (paragrafi 3.4.2 e 3.4.3).

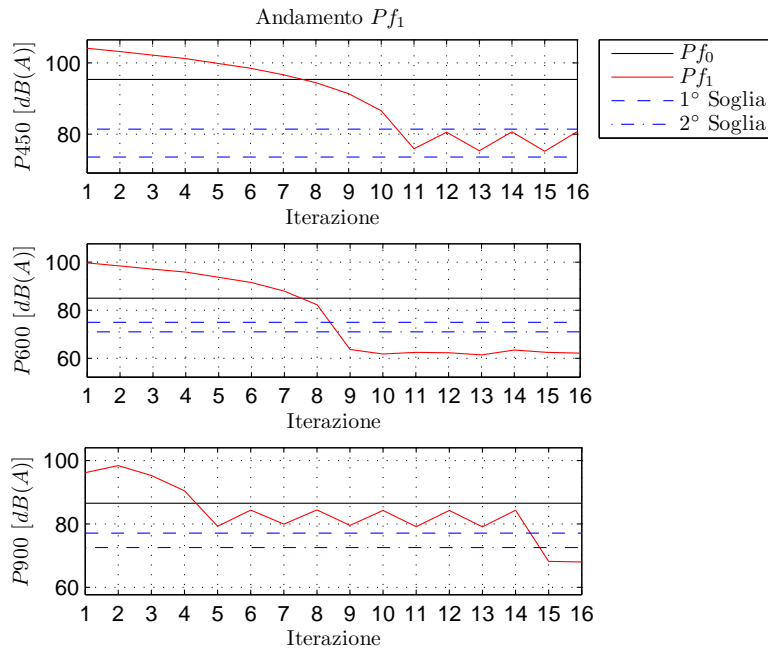


Figura 3.77: Andamento delle ampiezze dei picchi spettrali con algoritmo simultaneo basato sull'analisi spettrale.

Di seguito vengono riportati i risultati sperimentali ottenuti abbattendo un segnale di rumore composto dalla somma di tre segnali sinusoidali con frequenze pari a 450, 600 e 900  $Hz$  e con fasi pari rispettivamente a  $0^\circ$ ,  $180^\circ$  e  $0^\circ$ . Nella figura 3.77 è riportato l'andamento delle ampiezze dei picchi  $Pf_1$  in funzione del passo di iterazione per le diverse frequenze del segnale di rumore: si nota che questi valori decrescono rispetto ai loro valori iniziali  $Pf_0$  all'aumentare del passo di iterazione e raggiungono la soglia desiderata dopo un certo numero di iterazioni: il numero complessivo di iterazioni per l'abbattimento del rumore è pari al maggior numero di iterazioni richieste per l'abbattimento di una delle armoniche di cui è composto il segnale di rumore. La frequenza di 450  $Hz$  è quella che richiede il maggior numero di iterazioni (fig. 3.77), in particolare sono necessarie 17 iterazioni. A differenza di quanto visto nel metodo MFS, in questo metodo è necessario un numero di iterazioni e quindi un tempo di risoluzione dell'algoritmo inferiore dato dalla relazione 2.69.

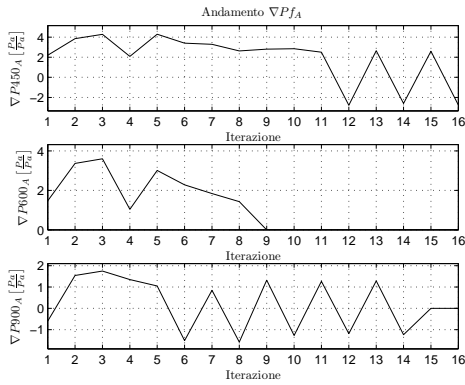


Figura 3.78: *Andamento del gradiente  $\nabla P f_A$  delle ampiezze del segnale di controllo con algoritmo simultaneo basato sull'analisi spettrale.*

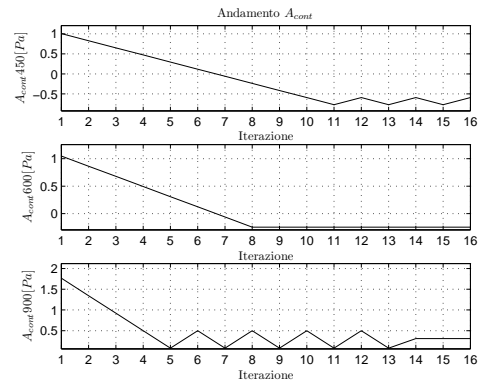


Figura 3.79: *Andamento delle ampiezze  $A_{cont}$  del segnale di controllo con algoritmo simultaneo basato sull'analisi spettrale.*

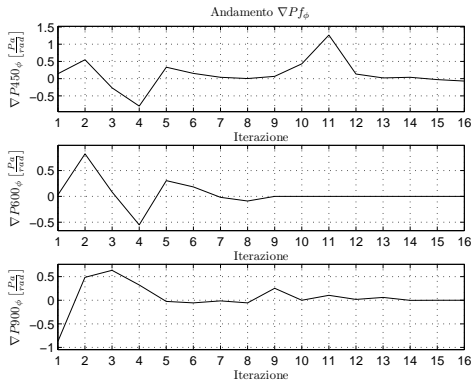


Figura 3.80: *Andamento del gradiente  $\nabla P f_\phi$  delle fasi del segnale di controllo con algoritmo simultaneo basato sull'analisi spettrale.*

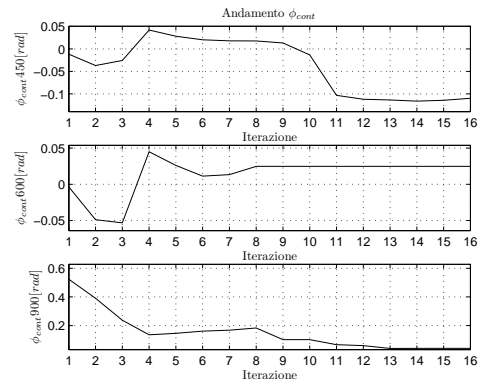


Figura 3.81: *Andamento delle fasi  $\phi_{cont}$  del segnale di controllo con algoritmo simultaneo basato sull'analisi spettrale.*

Le figure 3.78 e 3.80 riportano l'andamento dei gradienti dei parametri di controllo  $A_{cont}$  e  $\phi_{cont}$  alle diverse frequenze del segnale di rumore. Una volta raggiunta la condizione di minimo ad una determinata frequenza, di questa vengono mantenuti costanti i valori dei parametri di controllo. Come risulta evidente dal grafico la frequenza di 450  $Hz$  richiede 17 iterazioni, la frequenza di 600  $Hz$  richiede 8 iterazioni ed infine la frequenza di 900  $Hz$  richiede 14 iterazioni. Osservando le figure 3.78 e 3.79 l'algoritmo è in grado di stabilizzarsi attorno alla posizione di minimo, per quanto riguarda l'am-

piezza del segnale di controllo, a partire dall'undicesima iterazione per la frequenza di 450  $Hz$ , dall'iterazione 9 per la frequenza di 600  $Hz$  e dalla quinta iterazione per la frequenza di 900  $Hz$ , mentre la fase si assesta attorno alla posizione in cui è minimo il valore del picco spettrale in 12 iterazioni per la frequenza di 450  $Hz$ , in 7 iterazioni per la frequenza di 600  $Hz$  e in 10 iterazioni per la frequenza di 900  $Hz$  (figg. 3.80 e 3.81).

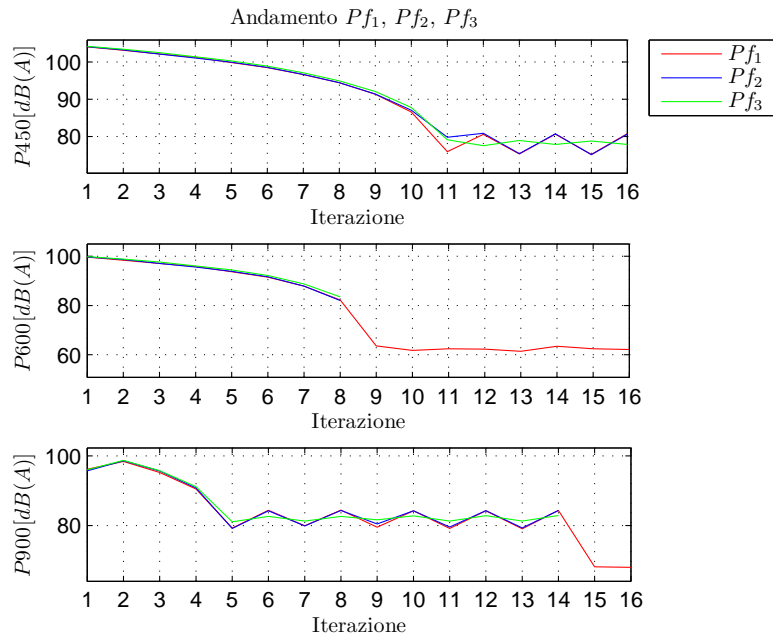


Figura 3.82: *Andamento delle ampiezze dei picchi spettrali  $Pf_1, Pf_2$  e  $Pf_3$  con algoritmo simultaneo basato sull'analisi spettrale.*

La figura 3.82 mostra l'andamento dei valori dei picchi associati alla variazione dei parametri del segnale di controllo in corrispondenza delle frequenze di cui è composto il segnale di rumore. L'andamento monotono decrescente delle grandezze riportate nel grafico e, in particolare, la bassa variabilità di  $Pf_2$  e  $Pf_3$  attorno al valore di  $Pf_1$  sono sintomo di una scelta adeguata dei parametri per la convergenza dell'algoritmo. La figura 3.83 mostra invece l'andamento del valore del guadagno  $G_2^k$  in funzione del passo di iterazione. Le variazioni di segno dei gradienti che avvengono per le armoniche a 450 e 900  $Hz$  causano un dimezzamento del valore dei guadagni  $G_1$  che non influenzano, tuttavia, la definizione dei guadagni  $G_2$ . Una volta raggiunta la convergenza in una determinata armonica il valore del relativo guadagno viene imposto a zero in modo da preservare i valori dei parametri di controllo determinati con l'algoritmo.

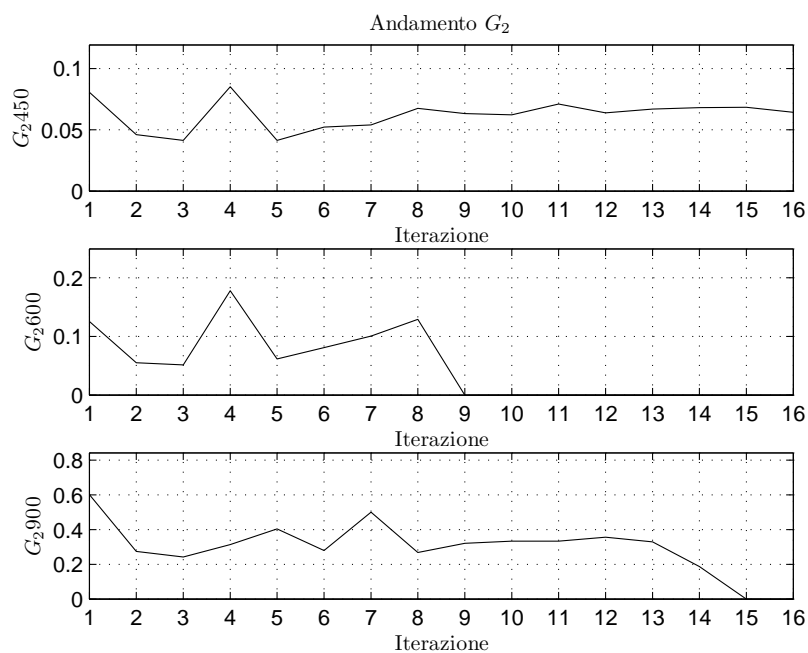


Figura 3.83: *Andamento dei guadagni  $G_2^k$  con algoritmo simultaneo basato sull'analisi spettrale.*

N° prova	Rumore		Controllo OFF	Controllo ON	Abbattimento
	$f_{rum}$ [Hz]	$\Phi_{rum}$ [°]	$P_{fout}$ [dB(A)]	$P_{fout}$ [dB(A)]	$\beta_{out}^{PF}$ [dB(A)]
1	1200	0	68	59	9
	700	0	63	54	9
	600	0	60	41	19
2	300	0	57	49	8
	400	180	59	43	16
3	450	0	60	51	9
	600	180	54	43	10
	900	0	64	50	14
4	300	0	58	51	7
	500	30	48	37	11
	700	80	60	55	5
	1100	120	63	59	4
5	300	0	58	51	7
	400	180	58	48	10
	600	0	60	45	15
	1000	180	60	46	14
6	320	0	67	57	10
	250	0	62	45	17
	390	0	63	48	15
7	430	0	60	50	10
	500	0	65	54	11
	570	0	68	51	17
8	1000	0	70	48	22
	1100	0	66	54	11
	900	0	74	48	26

Figura 3.84: *Riepilogo risultati con algoritmo basato sulla minimizzazione del picco spettrale con minimizzazione parallela.*

Nella tabella 3.84 vengono mostrati i risultati riepilogativi delle prove effettuate. L'abbattimento medio complessivo, valutato sull'ampiezza dei singoli picchi spettrali, è pari a circa  $12\text{ dB}$  rispetto al sistema di controllo non attivato. Uno dei maggiori vantaggi di questo metodo, oltre al fatto di minimizzare contemporaneamente più componenti del segnale di rumore, quindi con un tempo ridotto per la ricerca del segnale di controllo, è quello di essere completamente insensibile ai rumori presenti nell'ambiente esterno.

### 3.5 Conclusioni

Le tecniche per il controllo attivo del rumore rivestono notevole importanza e col passare del tempo queste vengono sempre più applicate a livello industriale in prodotti commerciali di largo consumo, si pensi ad esempio alla loro applicazione nelle cuffie per l'ascolto della musica (par. 1.4.2) o ai cellulari moderni dove con questa tecnologia è possibile migliorare la qualità acustica delle chiamate anche in ambienti rumorosi. Nelle analisi effettuate con l'apparato sperimentale sviluppato in questo lavoro di tesi è stato possibile valutare le performance di abbattimento, nonché i pregi e i difetti dei diversi algoritmi per il controllo del rumore implementati. In questo paragrafo conclusivo si vogliono riassumere tali risultati. Considerando le quattro tecniche di controllo implementate è possibile affermare quanto segue:

- Tecnica di controllo ARMS.
  - Abbattimento di rumori mono-armonici.
  - Convergenza dell'algoritmo fortemente influenzata dal rumore presente nell'ambiente esterno.
  - Fattore di abbattimento  $\beta$  ridotto a causa della possibile presenza di rumore nell'ambiente esterno.
  - Velocità di convergenza, in termini di numero di iterazioni, influenzata dall'ambiente esterno.
  - Impossibilità di realizzare abbattimenti di rumore multi-armonici.
  - Abbattimento ottenuto, valutato sul valore dell'ampiezza del picco in frequenza del segnale di rumore, pari ad una media  $6\text{ dB}$  all'uscita del condotto.
  - Frequenze di rumore ridotte nel range  $300 \div 1200\text{ Hz}$ .

- Tecnica di controllo AS.
  - Abbattimento di rumori mono-armonici.
  - Convergenza dell'algoritmo non influenzata dal rumore presente nell'ambiente esterno.
  - Velocità di convergenza elevata in termini di numero di iterazioni, poiché l'abbattimento è concentrato attorno alla sola frequenza del rumore di interesse.
  - Con un ulteriore sviluppo della tecnica vi è la possibilità di realizzare abbattimenti di rumore multi-armonici.
  - Abbattimento ottenuto, valutato sul valore dell'ampiezza del picco in frequenza del segnale di rumore, pari ad una media di 11 *dB* all'uscita del condotto.
  - Frequenze di rumore ridotte nel range 300 ÷ 1200 *Hz*.
  
- Tecnica di controllo MFS.
  - Abbattimento di rumori multi-armonici.
  - Convergenza dell'algoritmo non influenzata dal rumore presente nell'ambiente esterno.
  - Velocità di convergenza ridotta in termini di numero di iterazioni, poiché il tempo necessario all'algoritmo per giungere alla convergenza globale è dato dalla somma dei tempi necessari per giungere a convergenza sulle singole frequenze.
  - Abbattimento ottenuto, valutato sul valore delle ampiezze dei picchi in frequenza del segnale di rumore, pari ad una media di 12 *dB* all'uscita del condotto. Questo valore indica l'abbattimento medio sulla singola frequenze delle armoniche costituenti il segnale di rumore. In termini di abbattimento sul segnale complessivo, questo valore risulta molto più elevato rispetto a quello ottenuto nelle tecniche di abbattimento per rumori mono-armonici.
  - Frequenze di rumore ridotte nel range 300 ÷ 1200 *Hz*.
  
- Tecnica di controllo MFP.
  - Abbattimento di rumori multi-armonici.

- Convergenza dell'algoritmo non influenzata dal rumore presente nell'ambiente esterno.
- Velocità di convergenza, in termini di numero di iterazioni, elevata e dettata dalla frequenza del segnale di rumore che necessita di un numero di iterazioni superiori per il raggiungimento del punto di minimo.
- Abbattimento ottenuto, valutato sul valore dell'ampiezza del picco in frequenza del segnale di rumore, pari ad una media di  $12 \text{ dB}$  all'uscita del condotto. Questo valore indica l'abbattimento medio sulla singola frequenza delle armoniche costituenti il segnale di rumore. In termini di abbattimento sul segnale complessivo, questo valore risulta molto più elevato rispetto a quello ottenuto nelle tecniche di abbattimento per rumori mono-armonici.
- Frequenze di rumore ridotte nel range  $250 \div 1200 \text{ Hz}$ .

Dai risultati ottenuti è possibile affermare che la migliore tecnica di abbattimento del rumore tra quelle analizzate, è la tecnica MPF che consente di abbattere rumori multi-armonici con abbattimento medio sulla singola frequenza dell'armonica del segnale di rumore pari a  $12 \text{ dB}$ .

Le tecniche di controllo attivo del rumore presentate mostrano una limitazione sull'abbattimento di oscillazioni di pressione ad elevata lunghezza d'onda, ossia per frequenze del rumore inferiori a  $250 \text{ Hz}$ , a causa degli attuatori utilizzati che non consentono di fornire sufficiente energia alle basse frequenze. Per far fronte a questa limitazione è possibile utilizzare una tipologia di attuatori diversi, come ad esempio casse acustiche di tipo *sub-woofer*, ma anche considerare tecniche di abbattimento completamente differenti basate ad esempio sull'utilizzo di dispositivi passivi. Questi dispositivi presentano un range di frequenze di funzionamento che comprende in parte quello delle tecniche attive e si estende maggiormente verso le basse frequenze, inoltre essi non richiedono energia esterna per effettuare l'abbattimento, pertanto risultano vantaggiosi in alcune situazioni.





## Capitolo 4

# Sistemi per l'abbattimento con attuazione indiretta

### 4.1 Abbattimento con attuazione indiretta

L'utilizzo delle tecniche di abbattimento attivo mediante altoparlanti, viste nei capitoli precedenti, è limitato per tre fattori principali:

- le prestazioni raggiungibili possono in certi casi essere inferiori rispetto all'utilizzo di altri metodi;
- l'energia deve essere immessa direttamente nel sistema, e per impianti di grandi dimensioni questi sistemi possono non essere la soluzione ottimale;
- i costi di altri metodi di abbattimento possono essere inferiori.

Inoltre, la presenza di un trasporto di massa, o di una pressione nel mezzo superiore a quella atmosferica, nei condotti in cui si desidera abbattere le onde di pressione, rende complicato l'utilizzo di tali metodologie di abbattimento. Per questi motivi, in alcuni casi l'abbattimento delle oscillazioni di pressione può essere effettuato mediante i sistemi passivi risonanti (figure 1.18 e 1.22). L'analisi proposta nel presente capitolo verrà affrontata in particolare sui dispositivi passivi noti in letteratura come *risonatori di Helmholtz* e *risonatori side-branch*. Questi sistemi, come mostrato nel paragrafo 1.4.1, possono anche essere utilizzati nel campo prettamente acustico, ossia anche nel caso in cui il mezzo di propagazione è l'aria e non vi sia trasporto di massa, tuttavia le loro dimensioni possono crescere notevolmente nel caso in cui si vogliono abbattere oscillazioni a basse frequenze.

I sistemi risonanti conseguono l'abbattimento delle oscillazioni di pressione sfruttando le loro caratteristiche di risonanza: questi sistemi infatti oscillano in modo naturale a delle particolari frequenze, note come *frequenze di risonanza*, che dipendono dalle caratteristiche geometriche degli stessi e dalle caratteristiche fisiche del gas che vi è contenuto. Lo scopo di un dispositivo risonante è quello, una volta inserito in un impianto, di far vibrare il fluido, o il gas in esso contenuto, alla frequenza dell'oscillazione di pressione che si intende abbattere.

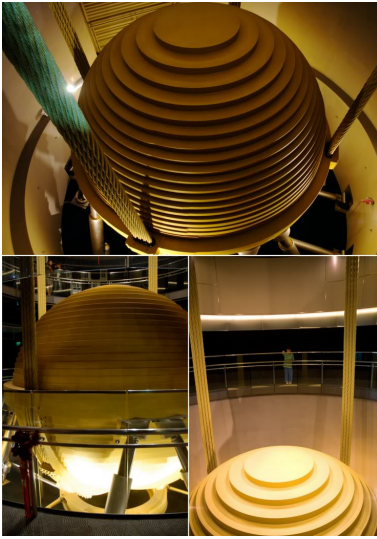


Figura 4.1: Esempio di applicazione di un TMD in una struttura civile.

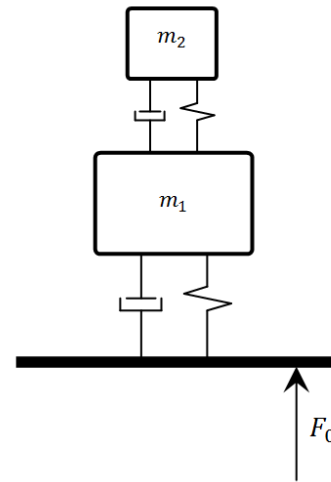


Figura 4.2: Modello di un assorbitore dinamico.

Il principio di funzionamento di questi sistemi può essere paragonato a quello degli assorbitori dinamici, noti in letteratura come *TMD: Tuned Mass Damping*. I TMD sono dei dispositivi che vengono montati su alcune strutture per ridurre l'ampiezza delle vibrazioni meccaniche al fine di aumentare il comfort, prevenire danni o addirittura per evitare cedimenti strutturali [23]. Tra i più importanti casi di utilizzo dei TMD in ambito civile vi sono quelli adottati per ridurre le ampiezze di oscillazione associate a terremoti o forti venti agenti sui grattacieli; nella figura 4.1 si riporta una foto dell'assorbitore dinamico utilizzato sul grattacielo Taipei 101 (Taiwan). Per comprendere il principio di funzionamento di un TMD, si faccia riferimento al modello riportato in figura 4.2. Esso è costituito da due masse  $m_1$  e  $m_2$  che rappresentano rispettivamente la massa del sistema su cui si desidera ridurre le oscillazioni e la massa dello smorzatore, nota come *mass damper*. La massa  $m_2$  è collegata alla massa  $m_1$  per mezzo di un sistema costituito da una molla e da uno smorzatore. La sollecitazione  $F_0$  proveniente

dal basamento si trasmette sulla massa  $m_1$  provocando un'oscillazione indesiderata. L'oscillazione presente su  $m_1$  si trasmette conseguentemente sulla massa  $m_2$  che a sua volta inizia ad oscillare. Esiste una particolare frequenza della sollecitazione in ingresso per la quale l'assorbitore dinamico è in grado di ridurre fortemente le oscillazioni della massa  $m_1$ . In corrispondenza di questa frequenza, nota come frequenza di risonanza del TMD, la massa  $m_2$  trasmette su  $m_1$  una forza uguale ed opposta a quella trasmessa dalle vibrazioni provenienti dal basamento.

Come mostrato nel paragrafo 4.3.1, i sistemi risonanti si comportano come i TMD, dove l'oscillazione della massa  $m_1$  può essere vista come l'oscillazione della pressione del gas propagante nel condotto in cui viene inserito il risonatore, mentre l'oscillazione della massa  $m_2$  è rappresentata dalla massa di aria "movimentata" dal risonatore. La frequenza di oscillazione della massa di aria "movimentata" dal risonatore avviene ad una particolare frequenza, nota come frequenza di risonanza del risonatore, che dipenderà a sua volta dalle caratteristiche geometriche dello stesso (par. 4.3). A differenza dei TMD, nei dispositivi risonanti<sup>1</sup>, in cui il mezzo di propagazione è l'aria, il fattore di smorzamento è pressochè trascurabile, pertanto tali sistemi non sono in grado né di assorbire, né di trasformare l'energia in ingresso in altra forma, ma possono solo immettere nuovamente nel sistema l'energia ricevuta, in modo da ridurre l'ampiezza dell'onda che si desidera abbattere. Nei TMD la presenza di un termine dissipativo consente nella pratica una sola estensione del range di frequenze di vibrazioni in grado di essere ridotte infatti, tali sistemi possono abbattere comunque le vibrazioni anche in assenza del termine di dissipazione ma con un funzionamento ottimale solo in un range di frequenze molto ristretto e concentrato attorno alla frequenza di risonanza a cui sono stati tarati. Il meccanismo di funzionamento dei risonatori non è quindi quello di assorbimento (par. 1.3.2), ma risulta essere un meccanismo intermedio tra quello di assorbimento e quello di pura cancellazione. In quest'ultimo caso sarebbe richiesto al dispositivo un'introduzione di energia nel condotto principale che può essere ottenuta solamente mediante un sistema attivo; si ricorda infatti che il meccanismo di perfetta cancellazione prevede l'introduzione nel sistema di due onde piane con ampiezza pari a quella della sorgente da abbattere, con la necessità quindi, da parte del risonatore, di

---

<sup>1</sup>Come visto nel paragrafo 1.4.1, in realtà è possibile disporre di sistemi risonanti con dissipazione introducendo all'interno di questi dispositivi del materiale fonoassorbente. Ovviamente l'introduzione del materiale fonoassorbente nel risonatore non consente l'utilizzo di quest'ultimo nel caso in cui vi sia un fluido con pressioni statiche superiori a quella ambiente o nel caso in cui vi sia un elevato trasporto di massa o ancora nel caso in cui il mezzo di propagazione non sia un gas.

una potenza totale superiore rispetto a quella contenuta nell'oscillazione di pressione della sorgente. Siccome l'energia restituibile da un sistema risonante è pari al massimo a quella in ingresso a tali dispositivi, ossia pari al massimo all'energia fornita dalla sorgente da abbattere, il meccanismo di abbattimento non può essere assimilato a quello di perfetta cancellazione.

Da quanto appena riportato si evidenzia che i dispositivi risonanti funzionano solo in corrispondenza della loro frequenza di risonanza e risulteranno invece inefficaci alle altre frequenze. È interessante quindi studiare a fondo tali dispositivi al fine di determinare dei metodi atti a modificare in modo automatico la loro frequenza di risonanza. Dato che la regolazione della frequenza di risonanza dei risonatori può avvenire modificando le loro caratteristiche geometriche, si riporta nel paragrafo 4.4 un'analisi di sensibilità della frequenza di risonanza conseguentemente alla variazione dei parametri geometrici. La modifica dei parametri geometrici, al fine di tarare la frequenza di risonanza di questi dispositivi sulla frequenza dell'onda di pressione da abbattere, può avvenire utilizzando ad esempio un sistema costituito da un attuatore e da una trasmissione. A differenza dei metodi attivi, l'attuazione in questo caso non viene utilizzata per abbattere *direttamente* l'onda di pressione, ma per modificare la geometria del dispositivo passivo, pertanto questi dispositivi vengono qui denominati come sistemi con *attuazione indiretta* per l'abbattimento delle oscillazioni di pressione.

Per studiare il comportamento fisico dei sistemi risonanti si ripropone di seguito la teoria di propagazione delle onde già affrontata in precedenza nel capitolo 1. La forma matematica utilizzata in questo capitolo, differente rispetto a quella precedente, consente di trattare in modo sistematico i sistemi costituiti da un condotto principale cui vengono collegate differenti appendici [27], rappresentate dai dispositivi risonanti utilizzati per attenuare le oscillazioni di pressione.

## 4.2 Teoria e modellazione di un condotto

Nel caso di propagazione di un gas in condotti di elevata lunghezza è necessario considerare in modo congiunto l'inerzia e l'elasticità del mezzo [29, 30]. Per convenienza di trattazione si ricava ora l'equazione delle onde considerando la posizione della particella di aria nel condotto. Con questo approccio risulterà semplice considerare aspetti come la presenza di dissipazione viscosa dell'aria che si propaga all'interno del condotto e disporre di un approccio sistematico per la risoluzione dell'equazione delle onde nei sistemi costituiti da un condotto principale cui vengono collegate differenti appendici

[28, 31]. La necessità di considerare in modo congiunto l'inerzia e l'elasticità del gas consente comunque sotto opportune ipotesi di utilizzare nella trattazione il modulo di comprimibilità. Verrà mostrato in seguito che tale ipotesi comporta una variazione dell'ampiezza di oscillazione della pressione dinamica limitata entro un certo range rispetto al valore statico di pressione nel condotto.

Un'altra ipotesi necessaria in questa trattazione riguarda la monodimensionalità della propagazione, ipotesi valida con buona approssimazione considerando condotti di elevata lunghezza (relazione 1.54). L'ipotesi di monodimensionalità consente quindi di studiare solo onde di pressione esenti da modi trasversali, ossia si considera del tutto trascurabile la presenza di oscillazioni di pressione normali all'asse del condotto.

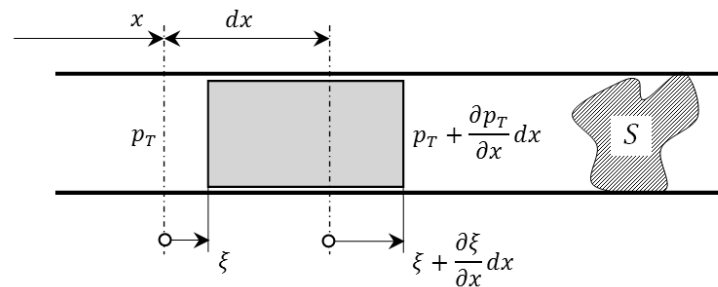


Figura 4.3: Modello del condotto con elemento infinitesimo di gas.

Per la determinazione del modello analitico si consideri la figura 4.3: a seguito di un disturbo presente nel sistema, ad esempio una variazione della portata in ingresso, le molecole del gas sono soggette ad uno spostamento. La figura mette in evidenza un elemento infinitesimo di fluido che si muove nella direzione della propagazione. In questa modellazione non risulta necessario specificare la forma della sezione, tuttavia essa deve mantenersi costante lungo il condotto. Si assume inoltre che lo spostamento delle particelle sia approssimativamente planare; tale assunzione comporta che le oscillazioni di velocità non siano laminari e che il profilo di velocità non sia quindi parabolico. Con il termine  $p_T$  si è voluto indicare la somma della pressione statica  $p_0$  e della pressione dinamica  $p$  presenti all'interno del condotto.

Il volume iniziale  $V_0$  dell'elemento di gas evidenziato in figura 4.3 è pari a:

$$V_0 = S dx, \quad (4.1)$$

dove  $S$  rappresenta l'area della sezione trasversale del condotto e  $dx$  la lunghezza iniziale infinitesima dell'elemento. A seguito di una forzante applicata al sistema, ad esempio

ad una variazione della pressione dinamica di fluido in ingresso al condotto, il volume dell'elemento di gas si incrementa di una quantità  $dV$  secondo quanto espresso dalla seguente relazione:

$$V_M = V_0 + dV = S \left( dx + \frac{\partial \xi}{\partial x} dx \right), \quad (4.2)$$

quindi l'incremento di volume  $dV$  è pari a:

$$dV = S \frac{\partial \xi}{\partial x} dx. \quad (4.3)$$

Per la legge di conservazione della massa, la massa del volume iniziale deve essere uguale a quella del volume modificato  $V_M$ . Il legame tra la pressione e la variazione di volume può essere rappresentato dal modulo di comprimibilità  $K_0$  se si è disposti a considerare un processo di compressione lineare:

$$\Delta p = -K_0 \frac{dV}{V_0}, \quad (4.4)$$

dove  $K_0 = \rho_0 c^2$ . Per semplicità di notazione si sostituisce al termine  $\Delta p$  il solo termine  $p$ , dove quest'ultimo rappresenta la variazione di pressione rispetto al suo valore medio  $p_0$ . Quest'ultima relazione consente di esprimere il legame tra la pressione e lo spostamento della particella di gas:

$$p = -K_0 \frac{\partial \xi}{\partial x}. \quad (4.5)$$

La relazione 4.5 è formalmente identica all'equazione di Eulero presentata nel capitolo 1 infatti, considerando la propagazione monodimensionale, è possibile riscrivere l'equazione di Eulero 1.12 come segue:

$$\frac{\partial p}{\partial x} = -\rho_0 \frac{\partial v}{\partial t}, \quad (4.6)$$

dove la velocità di particella  $v$  altro non è che la derivata temporale dello spostamento di particella  $\xi$ :

$$v = \dot{\xi} = \frac{\partial \xi}{\partial t}. \quad (4.7)$$

Derivando rispetto al tempo ambo i membri della relazione 4.5 e scambiando l'ordine di derivazione sullo spostamento  $\xi$  si ha:

$$\frac{\partial p}{\partial t} = -\rho_0 c^2 \frac{\partial v}{\partial x}, \quad (4.8)$$

da cui:

$$\frac{\partial v}{\partial x} = -\frac{1}{\rho_0 c^2} \frac{\partial p}{\partial t}. \quad (4.9)$$

Partendo ora dalla relazione di Eulero 4.6, derivando ambo i membri rispetto alla coordinata spaziale e scambiando l'ordine di derivazione si ottiene:

$$\frac{\partial^2 p}{\partial x^2} = -\rho_0 \frac{\partial}{\partial t} \frac{\partial v}{\partial x}. \quad (4.10)$$

Sostituendo la relazione 4.9 nella 4.10 si ottiene l'equazione delle onde monodimensionale espressa in termini di pressioni (si veda capitolo 1 equazione 1.26):

$$\frac{\partial^2 p}{\partial x^2} = -\rho_0 \frac{\partial}{\partial t} \left( -\frac{1}{\rho_0 c^2} \frac{\partial p}{\partial t} \right), \quad (4.11)$$

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2}. \quad (4.12)$$

L'equazione delle onde appena ottenuta altro non è che una conferma dell'uguaglianza tra l'approccio di modellazione presentato nel capitolo 1 e quello appena presentato.

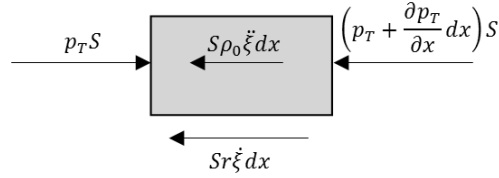


Figura 4.4: Forze agenti sull'elemento infinitesimo di gas.

La figura 4.4 mostra le differenti forze agenti sull'elemento infinitesimo di gas in cui si è considerata la presenza di azioni viscosse che dipendono dalla velocità di particella  $\dot{\xi}$ . Scrivendo un bilancio delle forze agenti sull'elemento di gas si ha:

$$S(p_0 + p) - S\rho_0 \ddot{\xi} dx - S \left( p_0 + p + \frac{\partial p}{\partial x} dx \right) - S r \dot{\xi} dx = 0, \quad (4.13)$$

$$\ddot{\xi} + \frac{r}{\rho_0} \dot{\xi} + \frac{1}{\rho_0} \frac{\partial p}{\partial x} = 0, \quad (4.14)$$

dove  $\rho_0$  è la densità media del fluido espressa in  $[N \text{sec}^2/m^4]$  e  $r$  è il coefficiente di smorzamento viscoso equivalente espresso in  $[N \text{sec}/m^4]$ . Sostituendo la relazione 4.5 nella 4.14 si ottiene l'equazione delle onde espressa in termini di spostamento di particella:

$$\frac{\partial^2 \xi}{\partial t^2} + \bar{\gamma} \frac{\partial \xi}{\partial t} = c^2 \frac{\partial^2 \xi}{\partial x^2}, \quad (4.15)$$

dove  $\bar{\gamma} = \frac{r}{\rho_0}$ . L'appendice A.3 vuole mettere in particolare evidenza l'equivalenza delle due trattazioni affrontate mostrando i passaggi matematici per ottenere l'equazione delle onde 1.26, espressa in termini di oscillazione di pressione, a partire dall'equazione delle onde 4.15, espressa in termini di spostamento di particella.

Considerando onde armoniche la soluzione dell'equazione 4.15 è la seguente:

$$\xi(x, t) = A_1 e^{j(\omega t - \bar{k}x)} + B_1 e^{j(\omega t + \bar{k}x)}, \quad (4.16)$$

dove  $\bar{k} = \frac{\omega}{\bar{c}}$  è il numero d'onda modificato e  $\bar{c}$  è la velocità del suono modificata. Sostituendo la soluzione nell'equazione di moto 4.15 si ha:

$$-\omega^2 \xi + \bar{\gamma} (j\omega) \xi = -\bar{k}^2 c^2 \xi, \quad (4.17)$$

da cui

$$\bar{k} = \frac{\omega}{c} \sqrt{1 - j \frac{\bar{\gamma}}{\omega}}, \quad (4.18)$$

dove

$$\bar{c} = \frac{c}{\sqrt{1 - j \frac{\bar{\gamma}}{\omega}}}. \quad (4.19)$$

Dato che il rapporto  $\frac{\bar{\gamma}}{\omega} \ll 1$  è possibile espandere la radice quadrata e scrivere il numero d'onda modificato approssimato:

$$\bar{k} = \frac{\omega}{c} \left( 1 - j \frac{\bar{\gamma}}{2\omega} \right). \quad (4.20)$$

Rielaborando l'equazione 4.20 si ha:

$$j\bar{k} = j \frac{\omega}{c} + \frac{\bar{\gamma}}{2c}, \quad (4.21)$$

$$\gamma = jk + a, \quad (4.22)$$

avendo introdotto il numero d'onda  $k = \frac{\omega}{c}$  e il parametro  $a = \frac{\bar{\gamma}}{2c}$ . Sostituendo la relazione 4.21 nella 4.16, è possibile riscrivere la soluzione dell'equazione delle onde come:



$$\xi(x, t) = (A_1 e^{-\gamma x} + B_1 e^{\gamma x}) e^{j\omega t}. \quad (4.23)$$

Considerando il legame (4.5) tra pressione dinamica  $p$  e spostamento di particella  $\xi$  è possibile determinare la soluzione in termini di pressione dinamica:

$$p(x, t) = \gamma \rho_0 c^2 (A_1 e^{-\gamma x} - B_1 e^{\gamma x}) e^{j\omega t}. \quad (4.24)$$

La soluzione può anche essere espressa in termini di portata volumetrica come segue:

$$q(x, t) = S \dot{\xi}, \quad (4.25)$$

$$q(x, t) = j\omega S (A_1 e^{-\gamma x} + B_1 e^{\gamma x}) e^{j\omega t}. \quad (4.26)$$

Trascurando la dipendenza temporale è possibile determinare la massima ampiezza della distribuzione di pressione e portata dinamica lungo la coordinata di propagazione  $x$ :

$$P(x) = \gamma \rho_0 c^2 (A_1 e^{-\gamma x} - B_1 e^{\gamma x}), \quad (4.27)$$

$$Q(x) = j\omega S (A_1 e^{-\gamma x} + B_1 e^{\gamma x}). \quad (4.28)$$

Nel capitolo 2.1 è stata introdotta la definizione di impedenza acustica come il rapporto la pressione e la velocità di particella. In questo caso si fornisce una definizione di impedenza leggermente differente, ottenuta come il rapporto tra la pressione e la portata dinamica; la differenza rispetto alla definizione esposta nel capitolo 1 riguarda la presenza della sezione  $S$  che consente di esprimere il legame 4.25 tra la velocità di particella  $\xi$  e la velocità di volume  $q$ . L'impedenza in ingresso  $Z_0$  può pertanto essere ottenuta come:

$$Z_0 = Z(x = 0) = \frac{P(0)}{Q(0)}. \quad (4.29)$$

Si consideri ora un generico tratto di condotto di lunghezza  $x = L$  come quello rappresentato in figura 4.5; grazie al presente modello si vogliono determinare le costanti  $A_1$  e  $B_1$  della soluzione armonica (rel. 4.27 e 4.28) al fine di applicare questa soluzione ad un fluido all'interno di un generico condotto. Questa formulazione consentirà di definire un approccio sistematico per la determinazione di modelli di sistemi più complessi costituiti da un insieme di condotti di diametri differenti. Lo scopo dell'approccio

sistematico è quello di valutare la dinamica del flusso di gas nel sistema a seguito dell'inserimento nel condotto principale di un dispositivo risonante (ed eventualmente ad attuazione indiretta) per l'abbattimento delle oscillazioni di pressione.

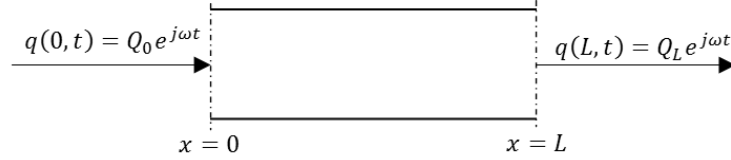


Figura 4.5: *Condotto elementare per la determinazione delle costanti  $A_1$  e  $B_1$ .*

Applicando le seguenti condizioni al contorno riferite alla portata dinamica:

$$Q(0) = Q_0, \quad (4.30)$$

$$Q(L) = Q_L, \quad (4.31)$$

all'equazione 4.28, si ottiene:

$$Q(0) = j\omega S(A_1 + B_1) = Q_0, \quad (4.32)$$

$$Q(L) = j\omega S(A_1 e^{-\gamma L} + B_1 e^{\gamma L}) = Q_L, \quad (4.33)$$

da cui è possibile determinare le costanti  $A_1$  e  $B_1$ :

$$A_1 = \frac{1}{2j\omega S \sinh \gamma L} (Q_0 e^{\gamma L} - Q_L), \quad (4.34)$$

$$B_1 = -\frac{1}{2j\omega S \sinh \gamma L} (Q_0 e^{-\gamma L} - Q_L). \quad (4.35)$$

Introducendo le costanti  $A_1$  e  $B_1$  nelle relazioni 4.27 e 4.28 si determinano le equazioni in forma chiusa di  $P(x)$  e  $Q(x)$ :

$$P(x) = \frac{\rho_0 c^2 \gamma}{2j\omega S \sinh \gamma L} [Q_0 \cosh \gamma(L-x) - Q_L \cosh \gamma x], \quad (4.36)$$

$$Q(x) = \frac{1}{\sinh \gamma L} [Q_0 \sinh \gamma(L-x) + Q_L \sinh \gamma x]. \quad (4.37)$$

Valutando la pressione  $P(x)$  in  $x = 0$  e  $x = L$ :

$$P(0) = \frac{\rho_0 c^2 \gamma}{2j\omega S \sinh \gamma L} [Q_0 \cosh \gamma L - Q_L] = P_0, \quad (4.38)$$

$$P(L) = \frac{\rho_0 c^2 \gamma}{2j\omega S \sinh \gamma L} [Q_0 - Q_L \cosh \gamma L] = P_L, \quad (4.39)$$

è possibile determinare le condizioni di pressione e portata dinamica nella sezione di ingresso del condotto ( $x = 0$ ) conoscendo le condizioni di pressione e portata nella sezione di uscita:

$$P_0 = P_L \cosh \gamma L + Q_L \frac{\rho_0 c^2 \gamma}{j\omega S} \sinh \gamma L, \quad (4.40)$$

$$Q_0 = P_L \frac{j\omega S}{\rho_0 c^2 \gamma} \sinh \gamma L + Q_L \cosh \gamma L. \quad (4.41)$$

Le relazioni 4.40 e 4.41 possono essere riscritte in forma matriciale come segue:

$$\begin{Bmatrix} Q_0 \\ P_0 \end{Bmatrix} = \begin{bmatrix} \cosh \gamma L & \frac{j\omega S}{\rho_0 c^2 \gamma} \sinh \gamma L \\ \frac{\rho_0 c^2 \gamma}{j\omega S} \sinh \gamma L & \cosh \gamma L \end{bmatrix} \begin{Bmatrix} Q_L \\ P_L \end{Bmatrix}, \quad (4.42)$$

$$\begin{Bmatrix} Q_0 \\ P_0 \end{Bmatrix} = \begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix} \begin{Bmatrix} Q_L \\ P_L \end{Bmatrix}, \quad (4.43)$$

in cui le costanti  $\tilde{A}$ ,  $\tilde{B}$ ,  $\tilde{C}$  e  $\tilde{D}$  sono pari a:

$$\tilde{A} = \tilde{D} = \cosh \gamma L, \quad (4.44)$$

$$\tilde{B} = \frac{j\omega S}{\rho_0 c^2 \gamma} \sinh \gamma L, \quad (4.45)$$

$$\tilde{C} = \frac{\rho_0 c^2 \gamma}{j\omega S} \sinh \gamma L. \quad (4.46)$$

Se risultano note le condizioni nella sezione di ingresso e si vogliono determinare le condizioni di pressione e portata dinamica in  $x = L$ , con alcune rielaborazioni della 4.43 si ottiene:

$$\begin{Bmatrix} Q_L \\ P_L \end{Bmatrix} = \begin{bmatrix} \tilde{A} & -\tilde{B} \\ -\tilde{C} & \tilde{D} \end{bmatrix} \begin{Bmatrix} Q_0 \\ P_0 \end{Bmatrix}. \quad (4.47)$$

### 4.2.1 Applicazione del modello ad un risonatore di Helmholtz

Utilizzando l'approccio matriciale, che mette in relazione le pressioni e le portate dinamiche nelle sezioni di ingresso e di uscita, si vuole determinare l'andamento di queste grandezze lungo il condotto principale a seguito dell'inserimento di un risonatore di Helmholtz.

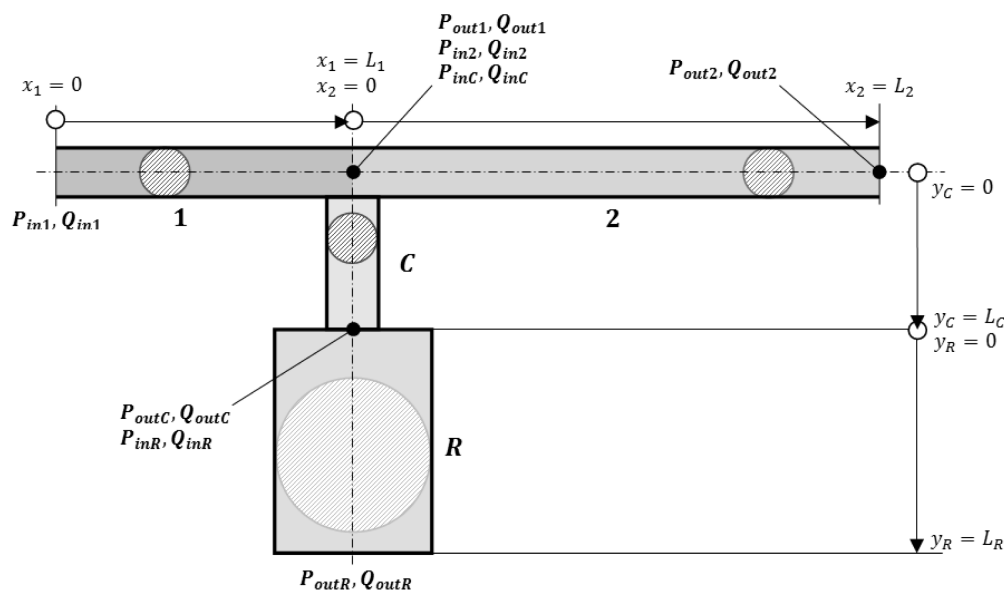


Figura 4.6: Modello condotto con risonatore di Helmholtz.

Si consideri la figura 4.6 che rappresenta graficamente il modello oggetto di studio dove sono state messe in evidenza le pressioni e le portate dinamiche in differenti sezioni. L'approccio ottenuto nel paragrafo precedente prevede infatti la suddivisione del sistema in semplici sottosistemi, costituiti da condotti a sezione costante, in cui è possibile esprimere delle relazioni sulle pressioni e sulle portate nelle sezioni di ingresso e di uscita dei singoli sottosistemi. Se si è interessati a determinare le sole condizioni di pressioni e portate nelle sezioni di ingresso e di uscita del condotto principale è sufficiente suddividere il sistema nei quattro sottosistemi rappresentati in figura 4.6, tuttavia qualora si intenda conoscere i valori di queste grandezze in corrispondenza di una generica sezione del sistema è necessario effettuare ulteriori suddivisioni che tendono a complicare la modellazione. Siccome nel caso in analisi si è interessati a conoscere l'andamento delle pressioni e delle portate lungo il condotto principale sarebbe necessario suddividere ulteriormente il sistema. Tuttavia, considerando la continuità di queste grandezze nelle sezioni di ulteriore suddivisione del sistema, è possibile correlare

le pressioni e le portate, in corrispondenza di queste sezioni, direttamente ai valori di pressione e portata nelle sezioni di ingresso e di uscita del condotto principale. Per semplificare ulteriormente il modello, tutte le appendici al condotto principale possono essere rappresentate per mezzo di una impedenza equivalente  $Z_R$  come mostrato in figura 4.7.

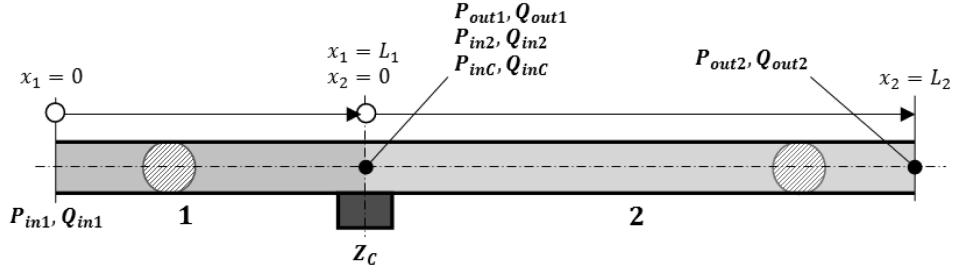


Figura 4.7: Modello condotto con impedenza equivalente.

L'approccio presentato nel paragrafo 4.2 prevede innanzitutto la scrittura della relazione 4.43 per ciascun tratto in cui si è suddiviso il sistema (figura 4.6):

$$\begin{Bmatrix} Q_{in1} \\ P_{in1} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_1 & \tilde{B}_1 \\ \tilde{C}_1 & \tilde{D}_1 \end{bmatrix} \begin{Bmatrix} Q_{out1} \\ P_{out1} \end{Bmatrix}, \quad (4.48)$$

$$\begin{Bmatrix} Q_{in2} \\ P_{in2} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_2 & \tilde{B}_2 \\ \tilde{C}_2 & \tilde{D}_2 \end{bmatrix} \begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix}, \quad (4.49)$$

$$\begin{Bmatrix} Q_{inC} \\ P_{inC} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_C & \tilde{B}_C \\ \tilde{C}_C & \tilde{D}_C \end{bmatrix} \begin{Bmatrix} Q_{outC} \\ P_{outC} \end{Bmatrix}, \quad (4.50)$$

$$\begin{Bmatrix} Q_{inR} \\ P_{inR} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_R & \tilde{B}_R \\ \tilde{C}_R & \tilde{D}_R \end{bmatrix} \begin{Bmatrix} Q_{outR} \\ P_{outR} \end{Bmatrix}, \quad (4.51)$$

dove  $P_{in\bar{x}}$ ,  $Q_{in\bar{x}}$ ,  $P_{out\bar{x}}$  e  $Q_{out\bar{x}}$  indicano rispettivamente la pressione in ingresso, la portata in ingresso, la pressione in uscita e la portata in uscita del generico sottosistema  $\bar{x}$  in cui è stato suddiviso il sistema complessivo. Le costanti  $\tilde{A}_{\bar{x}}$ ,  $\tilde{B}_{\bar{x}}$ ,  $\tilde{C}_{\bar{x}}$  e  $\tilde{D}_{\bar{x}}$  vengono determinate secondo quanto mostrato nelle relazioni 4.44, 4.45 e 4.46:

$$\tilde{A}_{\bar{x}} = \tilde{D}_{\bar{x}} = \cosh \gamma L_{\bar{x}}, \quad (4.52)$$

$$\tilde{B}_{\bar{x}} = \frac{j\omega S_{\bar{x}}}{\rho_0 c^2 \gamma} \sinh \gamma L_{\bar{x}}, \quad (4.53)$$

$$\tilde{C}_{\bar{x}} = \frac{\rho_0 c^2 \gamma}{j\omega S_{\bar{x}}} \sinh \gamma L_{\bar{x}}. \quad (4.54)$$

Una volta note le relazioni che mostrano i legami tra gli ingressi e le uscite dei singoli sottosistemi, vengono definiti i legami tra questi e le condizioni al contorno. In particolare si ha:

$$P_{out1} = P_{in2} = P_{inC}, \quad (4.55)$$

$$Q_{out1} = Q_{in2} + Q_{inC}, \quad (4.56)$$

$$P_{outC} = P_{inR}, \quad (4.57)$$

$$Q_{outC} = Q_{inR}, \quad (4.58)$$

$$P_{out2} = 0, \quad (4.59)$$

$$Q_{outR} = 0. \quad (4.60)$$

La relazione 4.55 indica che la pressione dinamica in corrispondenza di  $x_1 = L_1$  non può essere diversa da quella in ingresso nel condotto 2 e quella in ingresso nel collo del risonatore  $C$ , infatti la pressione in questo punto può variare solamente nel caso in cui vi sia una sorgente esterna in grado di incrementare il livello della pressione dinamica. La relazione 4.56 mostra invece che la portata dinamica in uscita dal condotto 1 si suddivide equamente tra la portata in ingresso nel condotto 2 e quella in ingresso nel collo del risonatore, in questo caso infatti parte del fluido proveniente dalla sezione di ingresso del condotto principale sarà deviato all'interno del risonatore<sup>2</sup>. Per la relazione 4.57 e 4.58 valgono le medesime considerazioni appena esposte. Considerando invece  $L_2$  sufficientemente grande o comunque considerando la presenza di un grosso serbatoio collegato nella sezione di uscita del condotto principale si può considerare nulla la pressione in  $x_2 = L_2$ , come espresso nella relazione 4.59. Infine la relazione 4.60 mette

<sup>2</sup>Si ponga attenzione al fatto che, nella modellazione in esame, si stanno considerando le pressioni e le portate dinamiche quindi, non viene considerato il valore statico di tali grandezze ma solamente la parte oscillante. A meno delle perdite concentrate che possono esserci in presenza di un'appendice collegata al condotto principale, ad esempio un risonatore di Helmholtz, con buona approssimazione è sensato considerare che parte della portata dinamica proveniente dalla sezione di ingresso del condotto principale prosegua verso l'uscita e che una parte venga deviata all'interno del risonatore.

in evidenza che non può esservi variazione della portata in corrispondenza della sezione finale del risonatore ( $y_R = L_R$ ).

Come evidenziato in precedenza il risonatore può essere visto come un'impedenza equivalente collegata al condotto principale:

$$Z_R = \frac{P_{inC}}{Q_{inC}}. \quad (4.61)$$

La determinazione dell'impedenza  $Z_R$  viene effettuata a partire dalle relazioni matriciali 4.50 e 4.51:

$$\begin{Bmatrix} Q_{inC} \\ P_{inC} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_C & \tilde{B}_C \\ \tilde{C}_C & \tilde{D}_C \end{bmatrix} \begin{bmatrix} \tilde{A}_R & \tilde{B}_R \\ \tilde{C}_R & \tilde{D}_R \end{bmatrix} \begin{Bmatrix} Q_{outR} \\ P_{outR} \end{Bmatrix}, \quad (4.62)$$

dove  $Q_{outR} = 0$ . Rielaborando la relazione 4.62 è possibile ottenere l'impedenza  $Z_R$  come:

$$Z_R = \frac{\tilde{C}_C \tilde{B}_R + \tilde{D}_C \tilde{D}_R}{\tilde{A}_C \tilde{B}_R + \tilde{B}_C \tilde{D}_R}. \quad (4.63)$$

Considerando la definizione dell'impedenza  $Z_R$ , la portata e la pressione in  $x_1 = L_1$  possono essere riscritte come:

$$\begin{cases} Q_{out1} = Q_{in2} + \frac{P_{inC}}{Z_R} = Q_{in2} + \frac{P_{in2}}{Z_R}, \\ P_{out1} = P_{out2} \end{cases}, \quad (4.64)$$

e in forma matriciale come:

$$\begin{Bmatrix} Q_{out1} \\ P_{out1} \end{Bmatrix} = \begin{bmatrix} 1 & \frac{1}{Z_R} \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} Q_{in2} \\ P_{in2} \end{Bmatrix}. \quad (4.65)$$

Sostituendo la relazione 4.65 nelle relazioni matriciali precedenti si determinano le condizioni di pressione e portata in ingresso al condotto  $P_{in1}$  e  $Q_{in1}$  in funzione delle condizioni di uscita  $P_{out2}$  e  $Q_{out2}$ :

$$\begin{Bmatrix} Q_{in1} \\ P_{in1} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_1 & \tilde{B}_1 \\ \tilde{C}_1 & \tilde{D}_1 \end{bmatrix} \begin{Bmatrix} Q_{out1} \\ P_{out1} \end{Bmatrix}, \quad (4.66)$$

$$\begin{Bmatrix} Q_{in1} \\ P_{in1} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_1 & \tilde{B}_1 \\ \tilde{C}_1 & \tilde{D}_1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{Z_R} \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} Q_{in2} \\ P_{in2} \end{Bmatrix}, \quad (4.67)$$

$$\begin{Bmatrix} Q_{in1} \\ P_{in1} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_1 & \tilde{B}_1 \\ \tilde{C}_1 & \tilde{D}_1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{Z_R} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{A}_2 & \tilde{B}_2 \\ \tilde{C}_2 & \tilde{D}_2 \end{bmatrix} \begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix}; \quad (4.68)$$

avendo assunto nulla la pressione nella sezione di uscita  $P_{out2} = 0$ , la relazione 4.68 può essere ulteriormente semplificata:

$$\begin{Bmatrix} Q_{in1} \\ P_{in1} \end{Bmatrix} = \begin{Bmatrix} \tilde{A}_1 \left( \tilde{A}_2 + \frac{\tilde{C}_2}{Z_R} \right) + \tilde{B}_1 \tilde{C}_2 \\ \tilde{C}_1 \left( \tilde{A}_2 + \frac{\tilde{C}_2}{Z_R} \right) + \tilde{D}_1 \tilde{C}_2 \end{Bmatrix} Q_{out2}. \quad (4.69)$$

Il modello appena presentato, permette di determinare solamente i valori di pressioni e portate dinamiche in corrispondenza delle sezioni di ingresso e di uscita. Volendo ottenere l'andamento di queste grandezze lungo tutto il condotto principale è necessario considerare il modello rappresentato in figura 4.8, che presenta una suddivisione in sotto-domini del sistema in  $\bar{x}_1$  e  $\bar{x}_2$ , dove valgono le relazioni di continuità di pressione e portata dinamica.

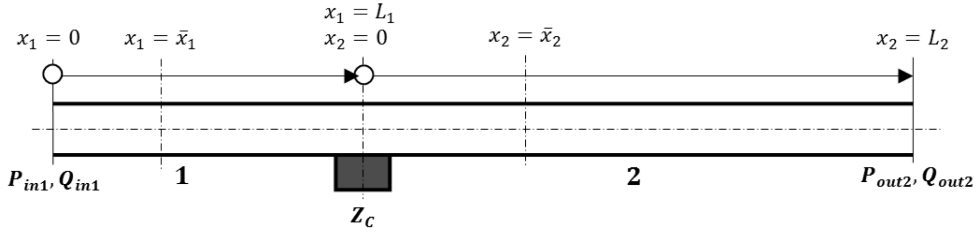


Figura 4.8: Modello condotto con impedenza equivalente, con suddivisione fittizia in  $\hat{x}_1$  e  $\hat{x}_2$ .

Considerando quindi la suddivisione del sistema in  $\bar{x}_1$  e  $\bar{x}_2$  si determinano le relazioni matematiche che descrivono l'andamento della pressione e della portata dinamica in un generico punto del condotto principale. In  $x_1 = \bar{x}_1 \leq L_1$  si ha:

$$\begin{Bmatrix} Q_{\bar{x}_1} \\ P_{\bar{x}_1} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_{\bar{x}_1} & \tilde{B}_{\bar{x}_1} \\ \tilde{C}_{\bar{x}_1} & \tilde{D}_{\bar{x}_1} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{Z_R} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{A}_2 & \tilde{B}_2 \\ \tilde{C}_2 & \tilde{D}_2 \end{bmatrix} \begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix}, \quad (4.70)$$

che può essere scritta in forma più compatta come:

$$\begin{Bmatrix} Q_{\bar{x}_1} \\ P_{\bar{x}_1} \end{Bmatrix} = \begin{bmatrix} \hat{A}_{\bar{x}_1} & \hat{B}_{\bar{x}_1} \\ \hat{C}_{\bar{x}_1} & \hat{D}_{\bar{x}_1} \end{bmatrix} \begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix}, \quad (4.71)$$

avendo effettuato la seguente sostituzione:

$$\begin{bmatrix} \hat{A}_{\bar{x}_1} & \hat{B}_{\bar{x}_1} \\ \hat{C}_{\bar{x}_1} & \hat{D}_{\bar{x}_1} \end{bmatrix} = \begin{bmatrix} \tilde{A}_{\bar{x}_1} & \tilde{B}_{\bar{x}_1} \\ \tilde{C}_{\bar{x}_1} & \tilde{D}_{\bar{x}_1} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{Z_R} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{A}_2 & \tilde{B}_2 \\ \tilde{C}_2 & \tilde{D}_2 \end{bmatrix}. \quad (4.72)$$

In  $x_2 = \bar{x}_2 \leq L_2$  si ha:



$$\begin{Bmatrix} Q_{\bar{x}_2} \\ P_{\bar{x}_2} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_{\bar{x}_2} & \tilde{B}_{\bar{x}_2} \\ \tilde{C}_{\bar{x}_2} & \tilde{D}_{\bar{x}_2} \end{bmatrix} \begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix}. \quad (4.73)$$

Per la definizione delle costanti  $\tilde{A}_{\bar{x}_1}$ ,  $\tilde{B}_{\bar{x}_1}$ ,  $\tilde{C}_{\bar{x}_1}$ ,  $\tilde{D}_{\bar{x}_1}$ ,  $\tilde{A}_{\bar{x}_2}$ ,  $\tilde{B}_{\bar{x}_2}$ ,  $\tilde{C}_{\bar{x}_2}$  e  $\tilde{D}_{\bar{x}_2}$  si faccia riferimento alle relazioni 4.52, 4.53 e 4.54. Rielaborando le relazioni matriciali 4.71 e 4.73 si ottengono i valori di pressioni e portate dinamiche in funzione delle condizioni di pressione  $P_{in1}$  e  $P_{out2}$  imposte al contorno. Infatti, riscrivendo la relazione 4.68 come:

$$\begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix} = \begin{bmatrix} \frac{Z_R}{\tilde{C}_1 \tilde{A}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{C}_2} & -\frac{\tilde{C}_1 \tilde{B}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{D}_2}{\tilde{C}_1 \tilde{A}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{C}_2} \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} P_{in1} \\ P_{out2} \end{Bmatrix}, \quad (4.74)$$

le relazioni 4.70 e 4.73 possono essere espresse nel seguente modo:

$$\begin{Bmatrix} Q_{\bar{x}_1} \\ P_{\bar{x}_1} \end{Bmatrix} = \begin{bmatrix} \hat{A}_{\bar{x}_1} & \hat{B}_{\bar{x}_1} \\ \hat{C}_{\bar{x}_1} & \hat{D}_{\bar{x}_1} \end{bmatrix} \begin{bmatrix} \frac{Z_R}{\tilde{C}_1 \tilde{A}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{C}_2} & -\frac{\tilde{C}_1 \tilde{B}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{D}_2}{\tilde{C}_1 \tilde{A}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{C}_2} \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} P_{in1} \\ P_{out2} \end{Bmatrix}, \quad (4.75)$$

$$\begin{Bmatrix} Q_{\bar{x}_2} \\ P_{\bar{x}_2} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_{\bar{x}_2} & \tilde{B}_{\bar{x}_2} \\ \tilde{C}_{\bar{x}_2} & \tilde{D}_{\bar{x}_2} \end{bmatrix} \begin{bmatrix} \frac{Z_R}{\tilde{C}_1 \tilde{A}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{C}_2} & -\frac{\tilde{C}_1 \tilde{B}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{D}_2}{\tilde{C}_1 \tilde{A}_2 Z_R + (\tilde{C}_1 + \tilde{D}_1 Z_R) \tilde{C}_2} \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} P_{in1} \\ P_{out2} \end{Bmatrix}, \quad (4.76)$$

che consentono di determinare, in tutti i punti del condotto principale, i valori di pressioni e portate dinamiche in funzione delle condizioni  $P_{in1}$  e  $P_{out2}$  imposte al contorno.

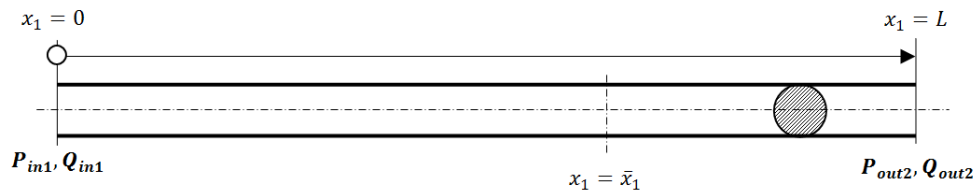


Figura 4.9: Modello del condotto principale in assenza di dispositivi risonanti.

La valutazione dell'abbattimento delle oscillazioni di pressione, legate all'inserimento di un risonatore nel condotto principale, può essere effettuata determinando il rapporto tra la pressione in presenza e quella in assenza del risonatore, valutate in un

generico punto del condotto. A tal fine si rende necessario determinare la relazione che consente di determinare la pressione dinamica in assenza di dispositivi risonanti. Si consideri quindi la figura 4.9 che rappresenta il modello del sistema costituito dal solo condotto principale. Applicando la relazione 4.43 al modello rappresentato in figura si ha:

$$\begin{Bmatrix} Q_{in1} \\ P_{in1} \end{Bmatrix} = \begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix} \begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix}. \quad (4.77)$$

In un generico punto del condotto, ad esempio in  $x_1 = \bar{x}_1 \leq L$ , si ha:

$$\begin{Bmatrix} Q_{\bar{x}_1} \\ P_{\bar{x}_1} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_{\bar{x}_1} & \tilde{B}_{\bar{x}_1} \\ \tilde{C}_{\bar{x}_1} & \tilde{D}_{\bar{x}_1} \end{bmatrix} \begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix}. \quad (4.78)$$

Rielaborando e sostituendo la relazione 4.77 nella relazione 4.78 si determina l'andamento della pressione e della portata dinamica in un generico punto del condotto, in funzione delle condizioni  $P_{in1}$  e  $P_{out2}$  imposte al contorno:

$$\begin{Bmatrix} Q_{\bar{x}_1} \\ P_{\bar{x}_1} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_{\bar{x}_1} & \tilde{B}_{\bar{x}_1} \\ \tilde{C}_{\bar{x}_1} & \tilde{D}_{\bar{x}_1} \end{bmatrix} \begin{bmatrix} \frac{1}{\tilde{C}} & -\frac{\tilde{D}}{\tilde{C}} \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} P_{in1} \\ P_{out2} \end{Bmatrix}. \quad (4.79)$$

Per chiarezza nella trattazione non viene riportata la relazione finale che consente di determinare il rapporto tra la pressione in presenza e quella in assenza del risonatore; la determinazione di questa quantità viene effettuata solo a livello numerico attraverso lo script di MatLab riportato nell'appendice E.2.

#### 4.2.2 Applicazione del modello ad un risonatore side-branch

Nel presente paragrafo si vuole valutare l'andamento delle pressioni e delle portate dinamiche a seguito dell'inserimento nel sistema di un risonatore di tipo side-branch. Un risonatore side-branch, come verrà mostrato in seguito, è costituito da un condotto di lunghezza  $L_R$  con sezione pari a quella del condotto principale in cui viene inserito. La figura 4.10 mostra il modello del sistema costituito dal condotto principale con un'appendice ad esso collegata rappresentata dal risonatore side-branch. Anche in questo caso tale appendice può essere considerata come un'impedenza equivalente  $Z_R$  collegata al condotto principale, si può pertanto fare riferimento alla figura 4.7.

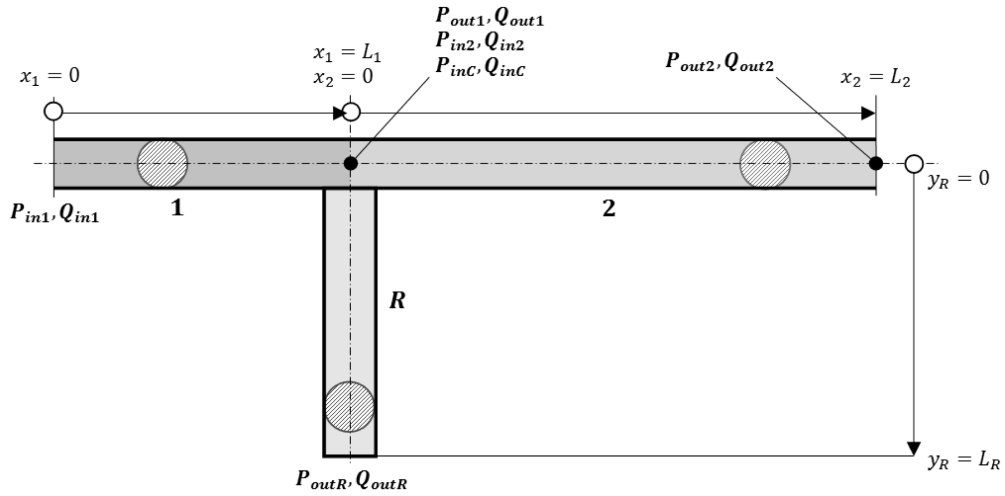


Figura 4.10: Modello condotto con risonatore side-branch.

Riscrivendo la relazione 4.43 per ciascun tratto di condotto in cui si è suddiviso il sistema (figura 4.10):

$$\begin{Bmatrix} Q_{in1} \\ P_{in1} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_1 & \tilde{B}_1 \\ \tilde{C}_1 & \tilde{D}_1 \end{bmatrix} \begin{Bmatrix} Q_{out1} \\ P_{out1} \end{Bmatrix}, \quad (4.80)$$

$$\begin{Bmatrix} Q_{in2} \\ P_{in2} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_2 & \tilde{B}_2 \\ \tilde{C}_2 & \tilde{D}_2 \end{bmatrix} \begin{Bmatrix} Q_{out2} \\ P_{out2} \end{Bmatrix}, \quad (4.81)$$

$$\begin{Bmatrix} Q_{inR} \\ P_{inR} \end{Bmatrix} = \begin{bmatrix} \tilde{A}_R & \tilde{B}_R \\ \tilde{C}_R & \tilde{D}_R \end{bmatrix} \begin{Bmatrix} Q_{outR} \\ P_{outR} \end{Bmatrix}, \quad (4.82)$$

e definendo le seguenti condizioni al contorno:

$$P_{out1} = P_{in2} = P_{inR}, \quad (4.83)$$

$$Q_{out1} = Q_{in2} + Q_{inR}, \quad (4.84)$$

$$P_{out2} = 0, \quad (4.85)$$

$$Q_{outR} = 0, \quad (4.86)$$

è possibile determinare l'impedenza  $Z_R$  dovuta alla presenza di un risonatore side-branch. Infatti, rielaborando la relazione 4.82 e considerando la condizione al contorno 4.86 si ottiene:

$$Z_R = \frac{P_{inR}}{Q_{inR}} = \frac{\tilde{D}_R}{\tilde{B}_R}. \quad (4.87)$$

Considerando la nuova definizione di impedenza 4.87 valgono tutte le relazioni viste in precedenza da 4.64 a 4.79 per il modello con il risonatore di Helmholtz. L'approccio ottenuto in precedenza è infatti generale e dipende essenzialmente dalla sola impedenza  $Z_R$  presente nel modello del sistema.

### 4.3 Sistemi indiretti per l'abbattimento

Per comprendere a fondo i meccanismi di funzionamento di un sistema indiretto per l'abbattimento delle oscillazioni di pressione è necessario applicare l'equazione delle onde e ricavare le relazioni analitiche che consentono di effettuare un loro dimensionamento. L'attenzione verrà concentrata sui sistemi indiretti Helmholtz e side-branch, ossia i dispositivi di cui si intende valutare sperimentalmente le performance di abbattimento. Nel paragrafo 4.4.3 viene proposto uno studio di questi dispositivi dal punto di vista numerico attraverso un'analisi ad elementi finiti. Un approccio numerico consente di validare le relazioni analitiche ottenute a partire dall'equazione delle onde e avvicinarsi al reale comportamento di questi dispositivi.

#### 4.3.1 Risonatore di Helmholtz

Al fine di comprendere il principio di funzionamento di un risonatore di Helmholtz viene affrontato il problema dal punto di vista fisico. Si consideri quindi la figura 4.11, dove viene mostrato un generico risonatore di Helmholtz, ossia un sistema costituito da una camera di risonanza con volume  $V_0$  e da un condotto, noto in gergo come *collo*, con una generica sezione pari a  $S_C$ . La camera di risonanza comunica con il condotto cui viene collegato l'intero dispositivo attraverso il collo del risonatore. Tale sistema con buona approssimazione può essere considerato come un semplice sistema vibrante, infatti tale dispositivo si comporta essenzialmente come se il gas contenuto nel collo fosse costituito da un elemento rigido incomprimibile che vibra su una molla, determinata a sua volta dalla presenza del gas comprimibile contenuto nella camera di risonanza. L'elemento rigido verrà in seguito denominato per comodità *collo d'aria*. L'assunzione dell'incomprimibilità del gas contenuto nel collo può essere giustificata considerando trascurabile il volume di aria contenuto nel collo rispetto a quello contenuto nella camera

di risonanza ossia, essendo talmente esigua la quantità di aria contenuta nel collo, la sua comprimibilità può essere trascurata [24].

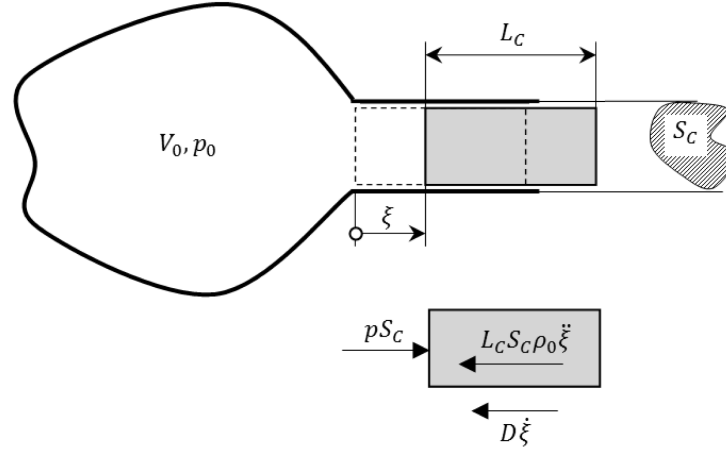


Figura 4.11: *Modello risonatore di Helmholtz.*

Considerando il processo di compressione dell'aria contenuta nella camera di risonanza approssimativamente lineare, è possibile analizzare le oscillazioni della massa di gas contenuta nel collo. Osservando il sistema come un semplice sistema vibrante ad un grado di libertà, si possono individuare principalmente tre forzanti dovute alla presenza del campo di pressione, delle forze d'inerzia, e degli effetti dissipativi cui è sottoposto il collo d'aria. Le grandezze caratteristiche del sistema illustrato in figura 4.11 sono:

- $V_0$ : volume a riposo del gas contenuto nella camera di risonanza espresso in  $[m^3]$ ;
- $p_0$ : pressione a riposo del gas contenuto nella camera di risonanza espressa in  $[Pa]$ ;
- $p$ : pressione dinamica legata all'oscillazione del collo d'aria espressa in  $[Pa]$ ;
- $L_C$ : lunghezza del collo espressa in  $[m]$ ;
- $S_C$ : area della sezione del collo espressa in  $[m^2]$ ;
- $D$ : coefficiente di attrito viscoso equivalente espresso in  $[Nsec/m]$ ;
- $\xi$ : spostamento, espresso in  $m$ , del volume di aria del collo rispetto alla condizione di riposo.

Uno spostamento positivo  $\xi$  del collo determina un incremento di volume  $dV$  del gas contenuto nella camera di risonanza:

$$dV = S_C \xi. \quad (4.88)$$

Questo incremento di volume comporta necessariamente una variazione della pressione  $\Delta p$  che si determina a partire dalla relazione 4.5:

$$p = -K_0 \frac{dV}{V} = -\rho_0 c^2 \frac{S_C \xi}{V_0}. \quad (4.89)$$

Per ottenere la forza dovuta alla variazione della pressione è sufficiente moltiplicare la relazione precedente per la sezione su cui viene applicata tale pressione:

$$F_p = p S_C = -\frac{\rho_0 c^2 S_C^2}{V_0} \xi. \quad (4.90)$$

Considerando l'inerzia del collo d'aria, la forza d'inerzia che si oppone al moto è pari a:

$$F_i = L_C S_C \rho_0 \ddot{\xi}. \quad (4.91)$$

A causa di una eventuale dissipazione di energia dovuta alla presenza di attriti si considera un coefficiente di attrito viscoso equivalente  $D$  e si definisce la forza dissipativa come:

$$F_d = D \dot{\xi}. \quad (4.92)$$

Esprimendo quindi il bilancio delle forze agenti sul collo d'aria (figura 4.11) si ha:

$$F_i + F_d - F_p = 0, \quad (4.93)$$

$$L_C S_C \rho_0 \ddot{\xi} + D \dot{\xi} + \frac{\rho_0 c^2 S_C^2}{V_0} \xi = 0, \quad (4.94)$$

$$m \ddot{\xi} + r \dot{\xi} + k \xi = 0, \quad (4.95)$$

dove  $m$ ,  $r$  e  $k$  sono rispettivamente la massa, lo smorzamento e la rigidità equivalente del sistema vibrante rappresentato dal risonatore di Helmholtz. Dalla relazione 4.95 è possibile determinare alcune importanti informazioni quali, ad esempio, la frequenza di risonanza  $f_R$  del risonatore e il fattore di smorzamento  $\zeta$ :

$$f_R = \frac{c}{2\pi} \sqrt{\frac{S_C}{L_C V_0}}, \quad (4.96)$$

$$\zeta = \frac{f_R D}{4L_C S_C \rho_0 \pi}. \quad (4.97)$$

La frequenza del risonatore è un importante parametro di progetto che consente, data la frequenza dell'oscillazione di pressione che si intende abbattere, di dimensionare il risonatore di Helmholtz al fine di abbattere tale oscillazione. A causa di fenomeni non lineari, come l'accelerazione del gas che si verifica in corrispondenza dell'imbocco del collo del risonatore, si apportano solitamente delle correzioni sulla lunghezza del collo. Tra le correzioni  $\alpha_C$  che si trovano in letteratura vi è la seguente, valida per risonatori di Helmholtz cilindrici:

$$L_C = L_{Ceff} + \alpha_C = L_{Ceff} + \frac{16}{3\pi} D_C, \quad (4.98)$$

dove con  $D_C$  si è indicato il diametro del collo. La lunghezza effettiva del risonatore  $L_{Ceff}$  è inferiore rispetto a quella derivata non considerando la presenza dei fenomeni non lineari. In letteratura vengono proposte differenti correzioni note con il termine *end correction* [26].

### 4.3.2 Risonatore side-branch

Per comprendere il funzionamento di un risonatore di tipo side-branch è necessario applicare la soluzione dell'equazione delle onde considerando il risonatore come un condotto con un'estremità chiusa ed un'estremità aperta. Le condizioni al contorno del sistema rappresentato in figura 4.12 sono le seguenti:

$$\xi(L_R, t) = 0 \quad (4.99)$$

$$\left. \frac{\partial \xi(x, t)}{\partial x} \right|_{x=0} = 0 \quad (4.100)$$

dove la prima condizione afferma che lo spostamento della particella, in corrispondenza dell'estremità chiusa del condotto ( $x = L_R$ ), deve essere nulla in qualsiasi istante infatti, considerando l'estremità rigida, non vi possono essere variazioni dello spostamento delle particelle di gas. La seconda condizione esprime invece che la pressione, in corrispondenza di un'estremità aperta, è costante e pari a quella esterna pertanto, non

essendovi variazione di pressione, deve essere nullo il termine legato alla variazione di volume<sup>3</sup>  $\frac{\partial \xi(x,t)}{\partial x}$ . Considerando la soluzione dell'equazione delle onde 4.16:

$$\xi(x, t) = A_1 e^{j(\omega t - \bar{k}x)} + B_1 e^{j(\omega t + \bar{k}x)}, \quad (4.101)$$

e applicando la prima condizione al contorno si ha:

$$\xi(L, t) = (A_1 e^{-j\bar{k}L} + B_1 e^{j\bar{k}L}) e^{j\omega t} = 0. \quad (4.102)$$

Derivando l'espressione 4.101 rispetto alla coordinata spaziale  $x$  e applicando la seconda condizione al contorno si perviene alla seguente relazione:

$$\left. \frac{\partial \xi(x, t)}{\partial x} \right|_{x=0} = -jk(A_1 - B_1) e^{j\omega t} = 0 \quad \Rightarrow \quad A_1 = B_1. \quad (4.103)$$

Sostituendo  $A_1 = B_1$  nella relazione 4.102, si ottiene la soluzione banale  $A = 0$  e una soluzione data dalla seguente relazione:

$$e^{-j\bar{k}L} + e^{j\bar{k}L} = 0, \quad (4.104)$$

che ammette soluzioni per

$$\bar{k}L = (2n + 1) \frac{\pi}{2} \quad \Rightarrow \quad n = 0, 1, 2, \dots \quad (4.105)$$

Considerando la prima frequenza propria del risonatore ( $n = 0$ ) e la definizione del numero d'onda modificato  $\bar{k}$  si ha:

$$\bar{k}L = \frac{\pi}{2}, \quad (4.106)$$

$$\frac{\omega}{c} \left( 1 - j \frac{\bar{\gamma}}{2\omega} \right) L = \frac{\pi}{2} \quad \Rightarrow \quad \omega = \frac{\pi c}{2L_R} + j \frac{\bar{\gamma}}{2}. \quad (4.107)$$

Dalla relazione 4.107 è possibile quindi determinare la prima frequenza propria del risonatore side-branch come:

$$f_R = \frac{c}{4L_R} + j \frac{\bar{\gamma}}{4\pi}. \quad (4.108)$$

---

<sup>3</sup>Si ricorda che la variazione di pressione è direttamente proporzionale alla derivata spaziale dello spostamento di particella secondo la relazione 4.5 pertanto, dovendo essere nulla la variazione di pressione, sarà nulla anche la derivata dello spostamento di particella.



Se è possibile trascurare gli effetti dissipativi la relazione 4.108 si riconduce a:

$$f_R = \frac{c}{4L_R}. \quad (4.109)$$

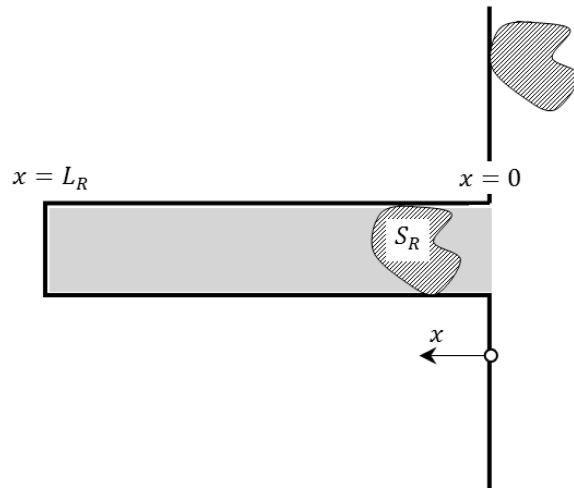


Figura 4.12: *Modello risonatore side-branch.*

#### 4.4 Sensibilità alla variazione dei parametri dei risonatori

Sfruttando le relazioni analitiche viste nel paragrafo 4.3, che descrivono il funzionamento dei risonatori in funzione delle loro caratteristiche geometriche, si presenta in questo paragrafo l'andamento della frequenza di risonanza di questi dispositivi in funzione dei parametri geometrici. Lo scopo è quello di effettuare un'analisi di sensibilità della frequenza di risonanza di questi dispositivi, al variare delle loro caratteristiche geometriche. Questa analisi di sensibilità consentirà di individuare un particolare parametro geometrico del dispositivo, che possa essere agevolmente modificato per mezzo di un generico meccanismo automatico, al fine di regolare la frequenza di risonanza in funzione delle caratteristiche dinamiche del flusso di gas di cui si intende ridurre le oscillazioni.

#### 4.4.1 Analisi di sensibilità di un risonatore di Helmholtz

Si consideri il risonatore di Helmholtz mostrato in figura 4.13, dove si sono assunte circolari sia la sezione del collo che la sezione della camera di risonanza. La scelta di considerare in questa sede un risonatore cilindrico è giustificata dal fatto che risulta sconsigliabile utilizzare risonatori con geometrie più complesse; in questi casi infatti sarebbe più difficoltosa l'analisi e la previsione del reale comportamento fisico di questi dispositivi ed inoltre aumenterebbero in modo non giustificato, rispetto alle prestazioni, i costi associati alla realizzazione di un risonatore con geometrie particolarmente complesse. Per l'applicazione e l'analisi del modello ottenuto nei paragrafi precedenti, è sufficiente disporre di sezioni costanti sia per il collo che per la camera di risonanza del risonatore: da questo punto di vista, il risonatore più semplice risulta essere costituito da due condotti di due diametri differenti con delle opportune flange che delimitano il condotto di diametro superiore.

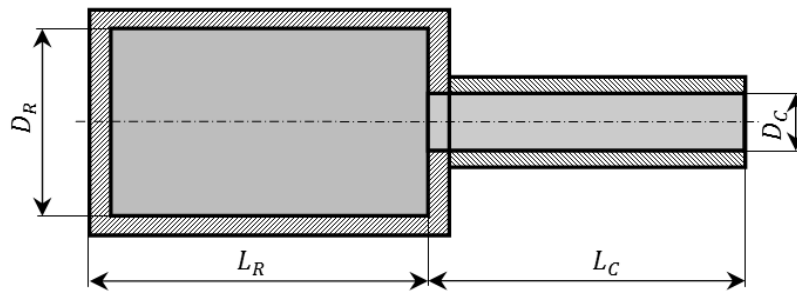


Figura 4.13: *Risonatore di Helmholtz cilindrico.*

La relazione 4.96 può essere quindi riscritta considerando la geometria cilindrica del risonatore:

$$f_R = \frac{c}{2\pi} \frac{D_C}{D_R} \sqrt{\frac{1}{L_C L_R}}, \quad (4.110)$$

dove  $D_C$ ,  $D_R$ ,  $L_C$ ,  $L_R$ , rappresentano rispettivamente il diametro del collo, il diametro della camera di risonanza, la lunghezza del collo a meno della correzione e la lunghezza della camera di risonanza.

È interessante valutare come variano le dimensioni del risonatore in funzione della frequenza delle oscillazioni di pressione che si intendono eliminare. Nella figura 4.14 si riporta l'andamento della lunghezza della camera di risonanza in funzione della frequenza di risonanza e della lunghezza del collo del risonatore avendo considerato i parametri riportati in tabella 4.1.

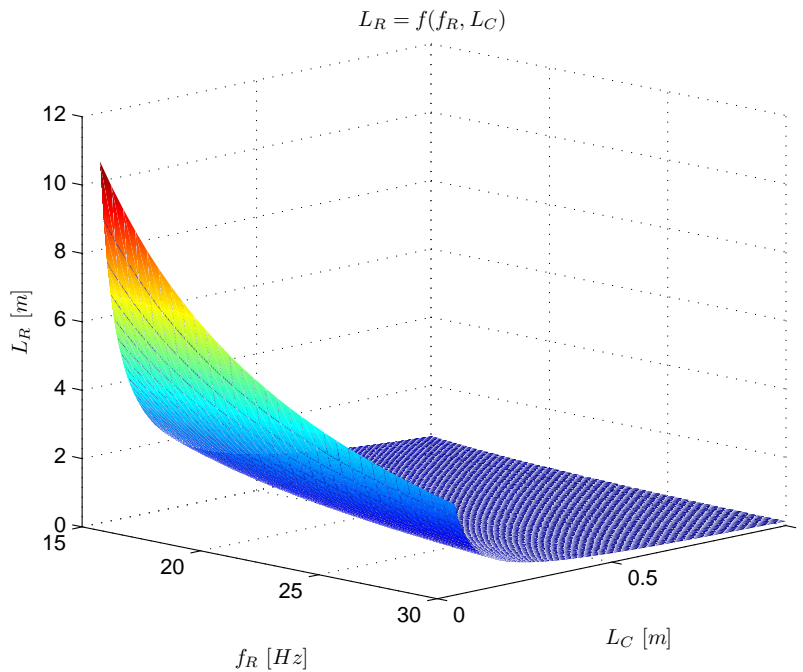


Figura 4.14: *Helmholtz*:  $L_R$  in funzione di  $f_R$  e  $L_C$ .

Parametro	Valore	Unità di misura	Osservazioni
$c$	343.2	$m/s$	$T_{\text{aria}} = 20^\circ C$
$f_R$	15 ÷ 30	$Hz$	
$L_C$	0.05 ÷ 1	$m$	
$D_C$	0.01	$m$	
$D_R$	0.05	$m$	

Tabella 4.1: *Analisi parametri risonatore di Helmholtz*:  $L_R$  in funzione di  $f_R$  e  $L_C$ .

Dalla figura 4.14 si nota come sia impraticabile realizzare un risonatore a basse frequenze di risonanza con lunghezze del collo molto ridotte, poiché cresce eccessivamente la lunghezza della camera di risonanza. Nel caso delle basse frequenze si rende necessario aumentare la lunghezza del collo ed aumentare il diametro del risonatore infatti,

dalla relazione 4.110, risulta che un aumento del volume della camera di risonanza, ottenuta o aumentando il diametro  $D_R$  o aumentando la lunghezza  $L_R$ , comporta un decremento della frequenza di risonanza del dispositivo.

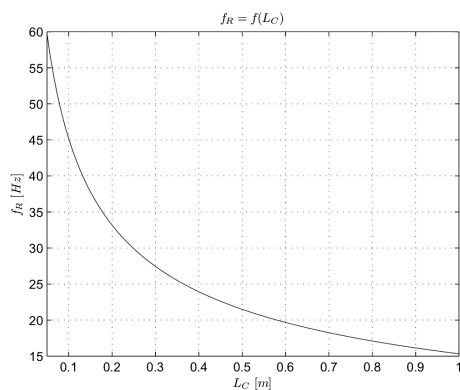


Figura 4.15: Helmholtz:  $f_R$  in funzione di  $L_C$ .

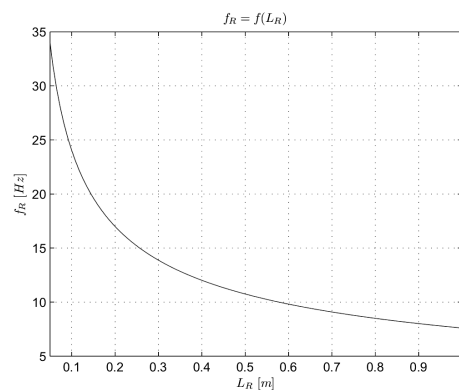


Figura 4.16: Helmholtz:  $f_R$  in funzione di  $L_R$ .

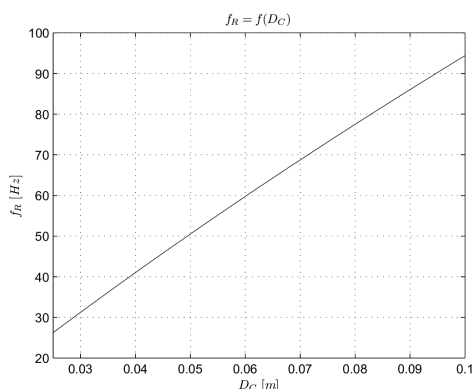


Figura 4.17: Helmholtz:  $f_R$  in funzione di  $D_C$ .

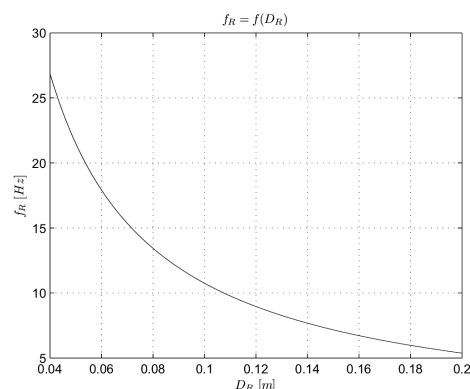


Figura 4.18: Helmholtz:  $f_R$  in funzione di  $D_R$ .

Nelle figure 4.15, 4.16, 4.17 e 4.18 si presentano l'andamento della frequenza di risonanza  $f_R$  del risonatore in funzione dei parametri geometrici  $L_C$ ,  $L_R$ ,  $D_C$  e  $D_R$ . I grafici sono stati ottenuti modificando un parametro alla volta e lasciando invariati tutti gli altri (tabella 4.2). Dalle figure 4.15 e 4.16 si mette in evidenza che per disporre di un risonatore con frequenze di risonanza particolarmente basse, non sia opportuno agire solamente sulle lunghezze del collo e della camera di risonanza poiché, la frequenza di risonanza ha un andamento iperbolico rispetto a questi due parametri. Per ovviare

a questi problemi, oltre ad un incremento della lunghezza del collo o della camera di risonanza, è possibile agire sui diametri  $D_C$  e  $D_R$ , come mostrato nelle figure 4.17 e 4.18.

Parametri risonatore	Riferimento				Unità di misura
	Fig. 4.15	Fig. 4.16	Fig. 4.17	Fig. 4.18	
$c$	343.2	343.2	343.2	343.2	$m/s$
$D_R$	0.05	0.1	0.1	$0.04 \div 0.2$	$m$
$D_C$	0.01	0.01	$0.025 \div 0.1$	0.01	$m$
$L_R$	0.5	$0.05 \div 1.0$	0.5	0.5	$m$
$L_C$	$0.05 \div 1.0$	0.5	0.5	0.5	$m$

Tabella 4.2: Parametri utilizzati per l'analisi di sensibilità mostrata nelle figure 4.15, 4.16, 4.17, 4.18.

La figura 4.17 rappresenta l'andamento della frequenza di risonanza  $f_R$  in funzione del parametro  $D_C$ . Come mostrato nella relazione 4.110, questo parametro ha un andamento opposto rispetto alle altre grandezze, infatti ad un diametro del collo superiore corrisponde una frequenza di risonanza superiore. È possibile pensare quindi di raggiungere abbattimenti di oscillazioni di pressione a basse frequenze agendo sul diametro del collo  $D_C$ , tuttavia come verrà mostrato in seguito, una tale scelta comporta una riduzione delle prestazioni di abbattimento del risonatore. Un risonatore con diametro del collo troppo piccolo rispetto a quello del condotto principale in cui viene inserito non è in grado di ricevere sufficiente energia dal sistema per andare in risonanza e, conseguentemente, non sarà in grado di abbattere l'onda di pressione alla frequenza per la quale è stato progettato. Se si vogliono ottenere prestazioni di abbattimento elevate, risulta evidente da queste ultime considerazioni che il diametro del collo debba essere il più grande possibile, ossia pari al diametro del condotto principale in cui sono presenti le oscillazioni di pressione da abbattere.

A differenza delle lunghezze del collo e del risonatore, dove ad un raddoppio di una di queste grandezze corrisponde una riduzione della frequenza di risonanza di solo il 30 %, agire sul diametro del risonatore risulta più conveniente poiché, ad un raddoppio del diametro  $D_R$  corrisponde un dimezzamento della frequenza di risonanza (riduzione del 50 %). Osservando la geometria di un risonatore di Helmholtz cilindrico, si constata che il parametro geometrico più facilmente modificabile per mezzo di un sistema attuatore-trasmissione è la lunghezza della camera di risonanza. In questo caso infatti

la variazione della lunghezza può essere ottenuta realizzando la camera di risonanza con un pistone a corsa regolabile (figura 4.19).

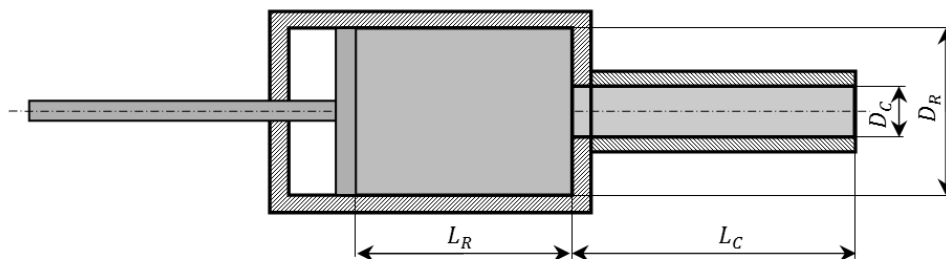


Figura 4.19: *Risonatore di Helmholtz cilindrico a geometria variabile.*

La corsa potrà poi essere modificata in modo manuale o automatico in funzione della frequenza dell'oscillazione di pressione che si intende abbattere. In realtà, per variare la frequenza di risonanza è possibile pensare di variare anche le altre grandezze, tuttavia in questi casi si rende necessario un sistema di movimentazione molto più complesso che potrebbe non giustificare i costi di una tale realizzazione.

Dall'analisi di sensibilità appena mostrata non risultano particolari limitazioni fisiche associate all'utilizzo di un risonatore di Helmholtz, se non quelle associate a dispositivi con dimensioni non realizzabili nella pratica. Tuttavia, il modello matematico ottenuto nel paragrafo 4.2.1 consente di trarre importanti conclusioni sulle performance di abbattimento di risonatori con differenti caratteristiche geometriche. Il modello viene qui utilizzato per determinare i valori di pressioni e portate dinamiche al fine di effettuare un confronto qualitativo delle performance di differenti risonatori. Il modello è costituito da un condotto di lunghezza pari a 20 metri e si è scelto di posizionare il risonatore a 3 metri dalla sezione di ingresso del condotto principale. La scelta dei valori da attribuire alle condizioni al contorno è, in questa sede, puramente arbitraria poiché si vuole effettuare solamente un confronto tra risonatori con differenti caratteristiche geometriche. Il modello acustico presentato in precedenza si limita a rappresentare il solo condotto in cui sono presenti le oscillazioni di pressione, prescindendo dal fatto che queste oscillazioni vengono generate da un sistema che presenta una propria dinamica. Le condizioni al contorno del modello utilizzato sono imposte agli estremi del condotto e non sull'effettiva sorgente delle oscillazioni, poiché questa non è modellata. Questa approssimazione porta ad una valutazione parziale ma sufficiente degli effetti che si verificano all'interno del condotto in presenza del dispositivo. I valori delle condizioni al

contorno  $P_{in1}$  e  $Q_{out2}$ , riportate in tabella 4.3, sono stati applicati in corrispondenza di un certo range di frequenze, al fine di analizzare il comportamento dinamico del flusso di aria nel condotto principale come se fosse soggetto ad un'onda impulsiva in grado di eccitare tutte le frequenze naturali del sistema.

Si pensi di voler realizzare un risonatore in grado di attenuare le oscillazioni di pressione alla frequenza di  $20\text{ Hz}$ . Analizzando la relazione 4.110 si hanno  $\infty^4$  possibili soluzioni. Per ridurre l'infinità di soluzioni possiamo inizialmente considerare un risonatore con diametro del collo pari a quello del condotto, mentre le altre grandezze geometriche vengono scelte arbitrariamente. Nella tabella 4.3 vengono riepilogate tutte le caratteristiche geometriche e fisiche del modello. Con  $f_{an}$  si è indicato il range di frequenze in cui sono state applicate le condizioni al contorno  $P_{in1}$  e  $P_{out2}$ .

Parametro	Valore	Unità di misura
$c$	343.2	$m/s$
$L_1$	3	$m$
$D_1$	0.010	$m$
$L_2$	17	$m$
$D_2$	0.010	$m$
$L_C$	0.800	$m$
$D_C$	0.010	$m$
$L_R$	0.373	$m$
$D_R$	0.050	$m$
$P_{in1}$	$10 \cdot 10^3$	$Pa$
$P_{out2}$	0	$Pa$
$f_{an}$	$5 \div 40$	$Hz$
$r$	17	$Nsec/m^4$
$f_R$	20	$Hz$

Tabella 4.3: Parametri modello per lo studio di un risonatore di Helmholtz.

Come mostrato in tabella 4.3 è stato introdotto nel modello un termine dissipativo in modo da mettere meglio in evidenza le frequenze di risonanza del sistema che tenderebbero altrimenti, in assenza del termine dissipativo, a valori infiniti. Il grafico in figura 4.20 rappresenta l'andamento della pressione dinamica lungo il condotto principale in assenza del risonatore di Helmholtz. È evidente la presenza dei modi di

vibrare propri del sistema che viene eccitato nel range di frequenze  $5 \div 40 \text{ Hz}$ . La figura 4.21 mostra l'andamento della pressione dinamica nel condotto principale a seguito dell'inserimento di un risonatore di Helmholtz accordato alla frequenza di  $20 \text{ Hz}$ : tale risonatore permette di abbattere notevolmente le oscillazioni di pressione a valle del punto di collegamento ( $x_2 \geq 0 \text{ m}$ ), mentre sembra incrementare leggermente il livello delle oscillazioni a monte ( $x_1 < 3 \text{ m}$ ). I valori delle pressioni e delle portate dinamiche ricavate con questo modello non possono essere considerati come valori quantitativi poiché l'imposizione di condizioni al contorno che non possono variare fissano la dinamica del flusso di aria alle estremità del condotto principale. L'ampia banda di abbattimento del risonatore di Helmholtz, visibile in figura 4.21, è dovuta principalmente alle caratteristiche proprie di questo tipo di risonatori, che si comportano essenzialmente come filtri stop-banda, ma in parte è legata anche alla presenza del termine dissipativo introdotto nel modello.

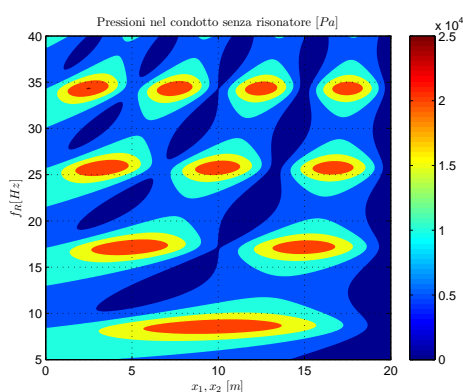


Figura 4.20: *Andamento pressione dinamica in assenza di dispositivi risonanti.*

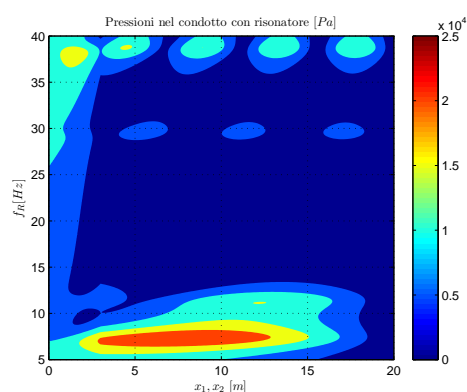


Figura 4.21: *Andamento pressione dinamica in presenza del risonatore di Helmholtz con frequenza  $f_R = 20 \text{ Hz}$ .*

Le figure 4.22 e 4.23 riportano rispettivamente l'andamento della portata dinamica del flusso d'aria in assenza e in presenza del risonatore di Helmholtz. Tali grafici mostrano come parte della portata dinamica del flusso d'aria proveniente dall'estremo sinistro del condotto principale venga assorbita dal risonatore rendendo praticamente trascurabile il livello di portata dinamica presente a valle del punto di inserimento del risonatore ( $x_2 \geq 0$ ).



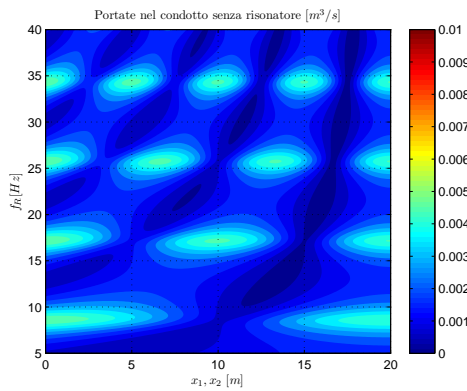


Figura 4.22: *Andamento portata dinamica in assenza di dispositivi risonanti.*

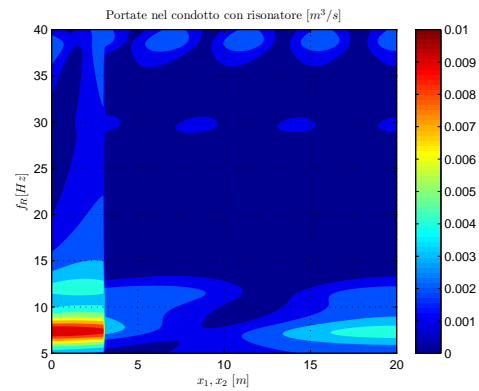


Figura 4.23: *Andamento portata dinamica in presenza del risonatore di Helmholtz con frequenza  $f_R = 20$  Hz.*

Si consideri ora la realizzazione di un risonatore con le caratteristiche geometriche riportate in tabella 4.4, dove si è scelto un diametro del collo  $D_C$  del risonatore pari al 50 % rispetto al risonatore precedente.

Parametro	Valore	Unità di misura
$L_C$	0.800	$m$
$D_C$	0.005	$m$
$L_R$	0.093	$m$
$D_R$	0.050	$m$
$f_R$	20	$Hz$

Tabella 4.4: *Parametri risonatore di Helmholtz a 20Hz con diametro del collo ridotto.*

Dalle constatazioni viste in precedenza e come mostrato nelle figure 4.24 e 4.25 si nota come tale risonatore non sia in grado di raggiungere le performance di abbattimento viste per il risonatore precedente. In questo caso infatti il risonatore riceve meno energia dal condotto principale e di conseguenza l'energia che il risonatore è in grado di reimmettere nel sistema è inferiore. Dal modello realizzato si constata pertanto che la dimensione ottimale del diametro del collo del risonatore risulta essere pari a quella del condotto principale.

Nelle figure 4.26 e 4.27 si vuole mettere in evidenza come, un risonatore con frequenza di risonanza pari a 100 Hz, i cui dati di progetto sono riportati in tabella 4.5, non sia in grado di apportare grosse modifiche alla dinamica del flusso d'aria nel range

5 – 40  $Hz$ , infatti gli andamenti delle pressioni e delle portate sono pressochè identici a quelli in assenza del risonatore (figure 4.20 e 4.22).

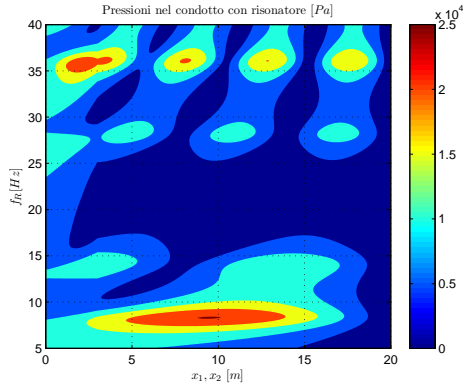


Figura 4.24: *Andamento pressione dinamica con risonatore di Helmholtz con diametro del collo ridotto.*

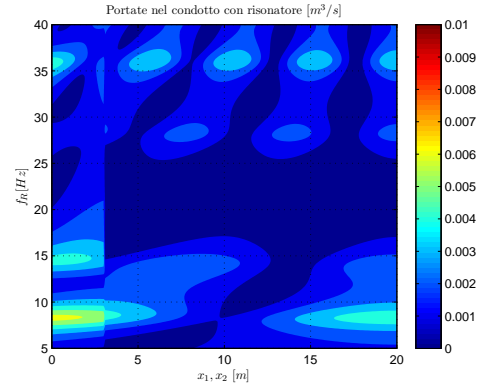


Figura 4.25: *Andamento portata dinamica con risonatore di Helmholtz con diametro del collo ridotto.*

Parametro	Valore	Unità di misura
$L_C$	0.200	$m$
$D_C$	0.010	$m$
$L_R$	0.060	$m$
$D_R$	0.050	$m$
$f_R$	100	$Hz$

Tabella 4.5: *Parametri risonatore di Helmholtz a 100Hz.*

Risulta particolarmente interessante valutare il rapporto tra la pressione in presenza del risonatore e la pressione presente nel condotto principale in assenza del dispositivo risonante. Considerando il rapporto tra queste due pressioni, ad esempio nel punto  $\bar{x}_2 = 6.9 m$  (nel modello in assenza del risonatore la coordinata è pari a  $\bar{x}_1 = 9.9 m$ ), e utilizzando lo script di MatLab riportato nell'appendice E.2, che sfrutta le relazioni 4.76 e 4.79, si ottiene l'andamento rappresentato in figura 4.28. Dalla figura si constata come il risonatore di Helmholtz, tarato alla frequenza di  $f_R = 20 Hz$  sia in grado di apportare benefici di abbattimento in una banda di frequenze molto ampia e centrata attorno alla frequenza di risonanza  $f_R$ . Il risonatore di Helmholtz, comportandosi di fatto come un TMD, sposta le frequenze di eccitazione del sistema causando un possibile incremento dell'ampiezza delle oscillazioni di pressione al di fuori del range di

frequenze entro il quale il risonatore di Helmholtz favorisce un buon abbattimento; in particolare, osservando più in dettaglio la figura 4.28, l'inserimento del risonatore con frequenza di risonanza  $f_R = 20 \text{ Hz}$ , causa un incremento dell'ampiezza delle oscillazioni di pressione del 40 % alla frequenza di  $11.5 \text{ Hz}$ ; ovviamente questo fenomeno si verifica solo nel momento in cui nel condotto principale è effettivamente presente un'oscillazione di pressione con frequenza pari a  $11.5 \text{ Hz}$ .

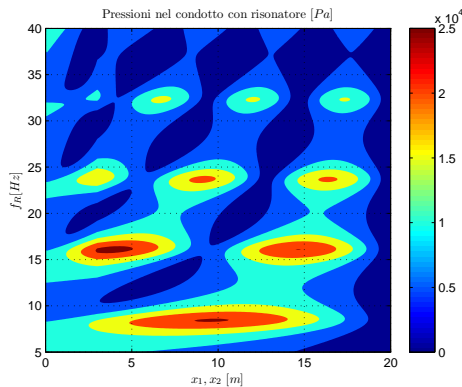


Figura 4.26: *Andamento pressione dinamica con risonatore di Helmholtz tarato a 100 Hz.*

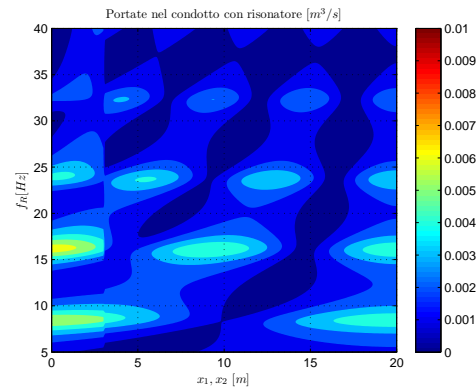


Figura 4.27: *Andamento portata dinamica con risonatore di Helmholtz tarato a 100 Hz.*

È evidente l'importanza del modello realizzato per lo studio ed il confronto del comportamento di differenti risonatori. Questa analisi non sarebbe stata possibile utilizzando la sola relazione 4.110 che descrive l'andamento della frequenza di risonanza in funzione dei parametri geometrici del risonatore e non consente di trarre alcuna informazione sulle prestazioni dei risonatori. Infatti, secondo questa relazione, non risulta evidente alcuna limitazione fisica sulle geometrie dei risonatori che possono essere realizzati. Nel paragrafo seguente si affronta la stessa analisi, effettuata per il risonatore di Helmholtz, utilizzando un dispositivo risonante di tipo side-branch.

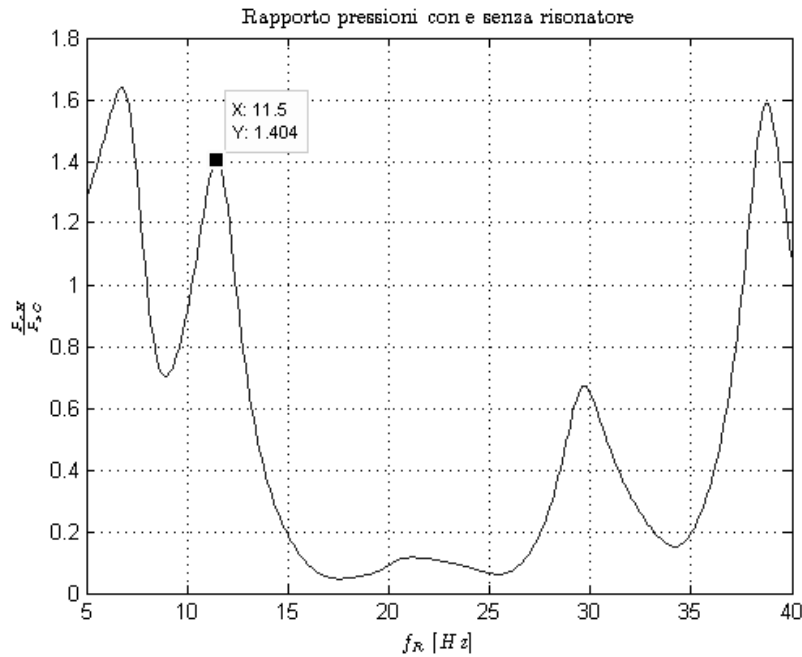


Figura 4.28: Andamento del rapporto delle pressioni con e senza Helmholtz in  $\bar{x}_2 = 5$  m.

#### 4.4.2 Analisi di sensibilità di un risonatore side-branch

Si consideri il risonatore side-branch a sezione cilindrica riportato in figura 4.29.

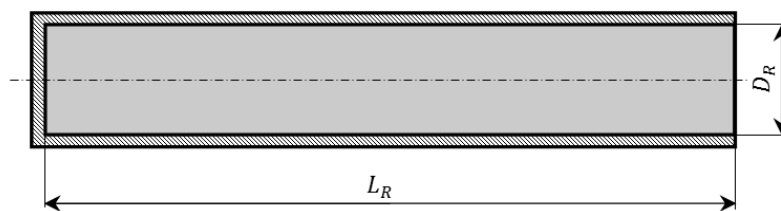


Figura 4.29: Risonatore side-branch cilindrico.

La frequenza di risonanza  $f_R$  di questa tipologia di risonatori può essere modificata agendo solamente sulla lunghezza  $L_R$  poiché le altre grandezze geometriche sono fissate dalla geometria del condotto principale, infatti la sezione del risonatore deve essere identica a quella del condotto principale in cui viene inserito ( $D_R = D_{\text{end}}$ ). La figura 4.30 riporta l'andamento della frequenza di risonanza  $f_R$  di questo dispositivo in

funzione della lunghezza  $L_R$ . Nella tabella 4.6 vengono riassunti i parametri utilizzati per la rappresentazione grafica.

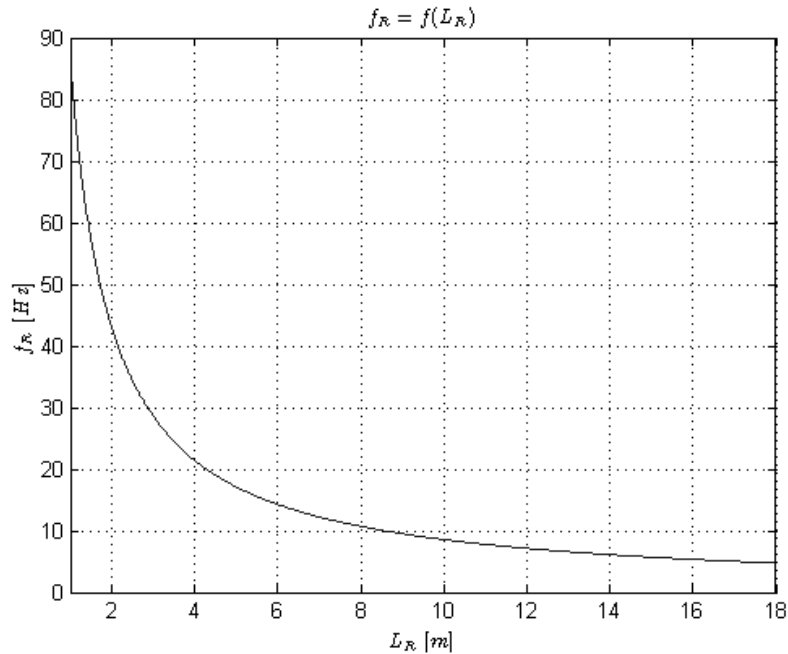


Figura 4.30: Side-branch:  $f_R$  in funzione di  $L_R$ .

Parametro	Valore	Unità di misura	Osservazioni
$c$	343.2	$m/s$	$T_{aria} = 20^\circ C$
$D_R$	0.010	$m$	$D_R = D_{cnd}$
$L_R$	1.0 ÷ 18.0	$m$	

Tabella 4.6: Analisi parametri risonatore side-branch:  $f_R$  in funzione di  $L_R$ .

Risulta evidente dalla figura 4.30 che per abbattere oscillazioni di pressione a bassa frequenza occorre utilizzare risonatori side-branch di elevata lunghezza che potrebbero risultare difficilmente realizzabili nella pratica. Infatti, per abbattere un'oscillazione di pressione a 20  $Hz$ , occorre un risonatore side-branch di lunghezza pari a  $L_R = 4.29 m$ . In modo analogo a quanto effettuato per il risonatore di Helmholtz, viene ora utilizzato il modello matematico proposto nel paragrafo 4.2.2 per valutare eventuali limitazioni fisiche associate all'utilizzo di questo tipo di risonatori. Per la scelta delle caratteristiche geometriche del condotto principale, delle condizioni al contorno, del termine di dissipazione viscosa e del range di frequenze di analisi, si faccia riferimento

al paragrafo 4.4.1 e alla tabella 4.3. Sulla scelta di questi parametri, riportati in tabella 4.7, valgono le medesime considerazioni esposte precedentemente. Essendo la geometria del condotto principale identica a quella utilizzata per l'analisi delle performance del risonatore di Helmholtz, l'andamento delle pressioni e delle portate dinamiche in assenza di dispositivi è identico a quello presentato nelle figure 4.20 e 4.22.

Parametro	Valore	Unità di misura
$c$	343.2	$m/s$
$L_1$	3	$m$
$D_1$	0.010	$m$
$L_2$	17	$m$
$D_2$	0.010	$m$
$L_R$	4.29	$m$
$D_R$	0.010	$m$
$P_{in1}$	$10 \cdot 10^3$	$Pa$
$P_{out2}$	0	$Pa$
$f_{an}$	$5 \div 40$	$Hz$
$r$	17	$Nsec/m^4$
$f_R$	20	$Hz$

Tabella 4.7: Parametri modello per lo studio di un risonatore side-branch.

Sfruttando le relazioni 4.75 e 4.76 è possibile ottenere l'andamento delle pressioni e delle portate dinamiche lungo il condotto principale (figure 4.31 e 4.32) in presenza di un risonatore side-branch tarato alla frequenza di risonanza  $f_R = 20 Hz$ . Le figure mettono in evidenza che i risonatori side-branch sono essenzialmente dei filtri *notch* (*filtro a spillo*) in grado di abbattere le oscillazioni di pressione in un range di frequenze piuttosto ridotto e centrato attorno alla frequenza di risonanza  $f_R$  del risonatore. Confrontando le figure 4.31 e 4.21 si nota che un risonatore di Helmholtz ha delle performance migliori poiché è in grado di abbattere le oscillazioni di pressione in un range più ampio rispetto a quello di un risonatore side-branch, con l'ulteriore vantaggio che il risonatore di Helmholtz risulta di dimensioni più compatte. Tuttavia uno svantaggio del risonatore di Helmholtz rispetto al risonatore side-branch, sembra essere quello di incrementare il livello delle oscillazioni di pressione a monte del punto di inserimento dello stesso. Per valutare questi fenomeni è necessario procedere attraverso analisi sperimentali.

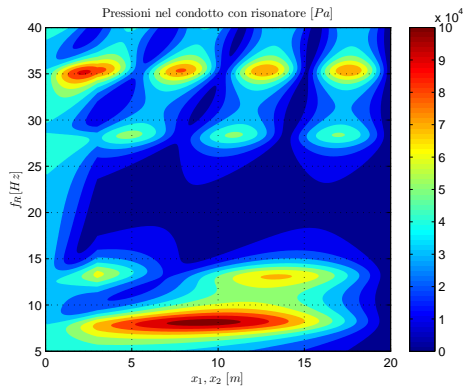


Figura 4.31: *Andamento pressione dinamica con risonatore side-branch con frequenza  $f_R = 20$  Hz.*

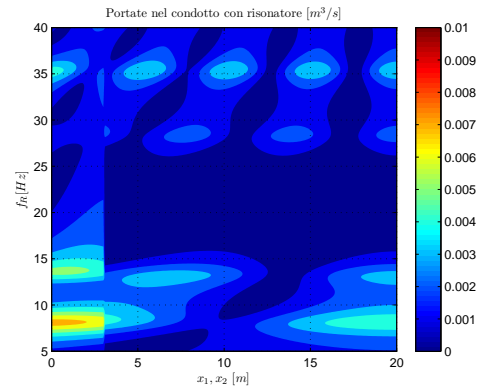


Figura 4.32: *Andamento portata dinamica con risonatore side-branch con frequenza  $f_R = 20$  Hz.*

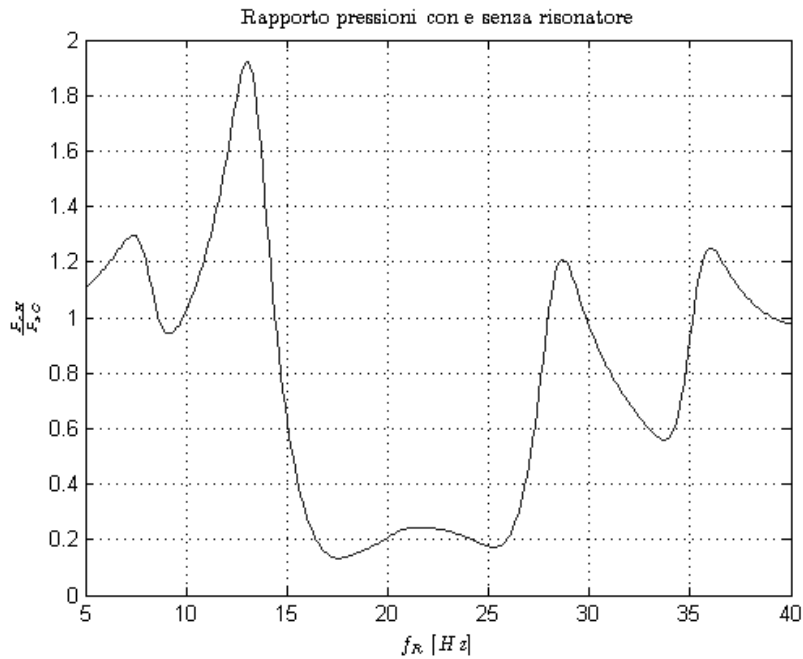


Figura 4.33: *Andamento del rapporto delle pressioni con e senza side-branch in  $\bar{x}_2 = 6.9$  m.*

La ridotta banda di abbattimento di un risonatore side-branch può essere osservata analizzando il rapporto tra la pressione dinamica in assenza e quella in presenza del risonatore side-branch determinata in un punto a valle della posizione di inserimento del risonatore ( $\bar{x}_1 = 3$  m). Considerando il rapporto tra queste due pressioni, ad esempio nel punto  $\bar{x}_2 = 6.9$  m (nel modello in assenza del risonatore la coordinata è pari a  $\bar{x}_1 =$

9.9 m), e utilizzando lo script di MatLab riportato nell'appendice E.3, che sfrutta le relazioni 4.76 e 4.79, si ottiene l'andamento rappresentato in figura 4.33. Confrontando questo grafico con quello ottenuto mediante l'utilizzo di un risonatore di Helmholtz (fig. 4.28) si nota immediatamente come il risonatore side-branch abbia un'efficacia minore rispetto al risonatore di Helmholtz; infatti, l'ampiezza della banda in frequenza in cui, il risonatore side-branch, riesce ad apportare un sostanziale abbattimento delle oscillazioni di pressione è molto ridotta rispetto a quella ottenibile con un risonatore di Helmholtz.

Per la valutazione delle limitazioni fisiche, legate alla presenza di fenomeni non lineari, come la discontinuità del flusso in corrispondenza dell'imbocco del risonatore, occorre studiare il problema dal punto di vista fluidodinamico utilizzando le equazioni di Navier-Stokes o procedere con indagini sperimentali. Da un'attenta analisi associata allo sviluppo di un adeguato modello numerico fluidodinamico si è constatata l'ineadeguatezza di questa implementazione rispetto ai bassi costi legati alla realizzazione di prototipi fisici. Nel paragrafo 4.4.3 verrà riportata un'analisi numerica ad elementi finiti di tipo acustico, derivante dall'applicazione dell'equazione delle onde ai sistemi appena analizzati. Questo studio numerico consentirà di validare i risultati analitici riportati nel presente paragrafo. Nei capitoli seguenti verrà invece affrontato il problema dal punto di vista sperimentale, analizzando dapprima l'andamento delle oscillazioni di pressione in un reale compressore industriale e realizzando successivamente un apparato sperimentale per testare l'efficacia di abbattimento di differenti risonatori, progettati facendo riferimento ai modelli presentati in questo capitolo.

#### 4.4.3 Modellazione numerica dei risonatori

Con lo scopo di validare il modello analitico presentato nel paragrafo 4.2 e 4.2. si esegue una modellazione dei risonatori proposti utilizzando la tecnica degli elementi finiti, mediante il software di calcolo Comsol Multiphysics®. Con esso è possibile eseguire analisi acustiche in ambiente tridimensionale, inoltre è possibile effettuare sia il calcolo di soluzioni in regime stazionario, sia il calcolo in transitorio, integrando le equazioni differenziali nel tempo. Con il seguente approccio si vuole validare il modello analitico sviluppato, mediante il confronto dei risultati ottenuti dai due metodi su casi analoghi. Nel seguito si riportano le analisi effettuate su un condotto rettilineo su cui vengono testati un risonatore di tipo side-branch e un risonatore di Helmholtz.



### Modellazione del risonatore di Helmholtz

Si vuole analizzare l'andamento delle pressioni in un condotto in cui è inserito un risonatore di Helmholtz, in un punto a valle del risonatore. Le caratteristiche geometriche del sistema sono riportate nella tabella 4.3. La rappresentazione geometrica del sistema è mostrata in figura 4.34.

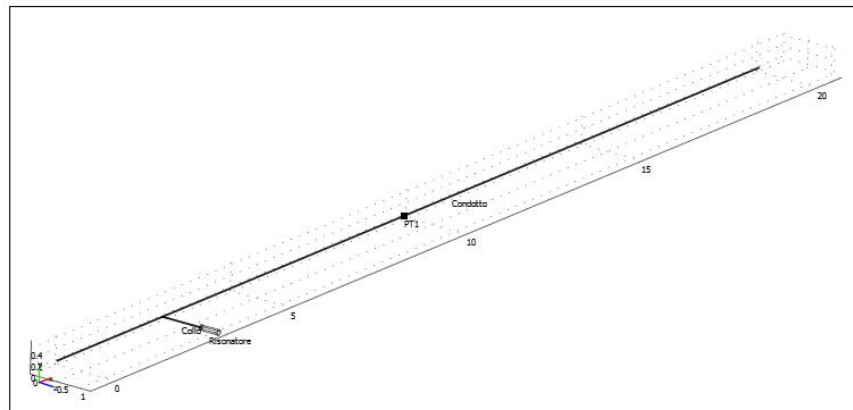


Figura 4.34: *Modello numerico del sistema di analisi del risonatore di Helmholtz.*

Il sistema è costituito da un condotto di lunghezza pari a 20 m dove alla distanza di 3 m dall'ingresso è inserito il risonatore di Helmholtz. In ingresso al sistema viene definita una forzante determinata da una pressione dinamica variabile di ampiezza pari a  $P_{in} = 10^4 Pa$  e di frequenza variabile da 5 a 40 Hz. La pressione dinamica in uscita dal condotto viene fissata a  $P_{out} = 0 Pa$ , che rappresenta la condizione di condotto aperto. Lo scopo dell'analisi è quello di calcolare l'andamento delle pressioni nel punto  $P_1$ , che si trova ad una distanza pari a 9.9 m dall'ingresso del condotto e al centro della sezione dello stesso, e confrontarlo con l'andamento delle pressioni nello stesso punto, del sistema senza il risonatore.

Una volta definita la geometria del sistema, le caratteristiche del fluido<sup>4</sup> e le condizioni al contorno, occorre definire la mesh in cui definire gli elementi. Per questa analisi si utilizzano elementi tetraedrali quadratici: la definizione della mesh viene effettuata sfruttando l'algoritmo di meshing del software, adottando i criteri per la definizione della massima grandezza degli elementi descritti nella sezione 2.1.1. Il risultato di questa operazione porta alla definizione di un sistema costituito da 3164 elementi e da 9188 gradi di libertà.

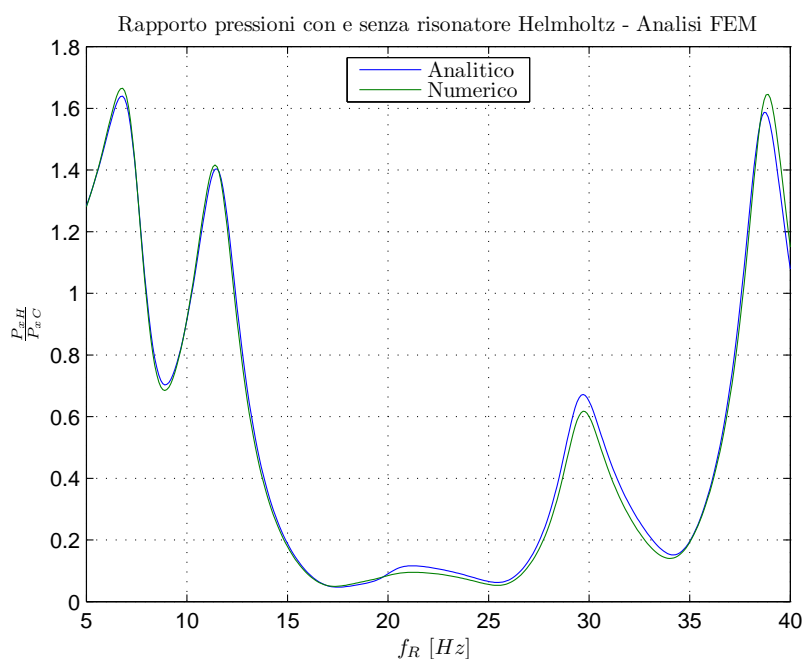


Figura 4.35: Rapporto delle pressioni con e senza risonatore di Helmholtz nel punto  $P_1$ : confronto fra i risultati del modello analitico e del modello numerico.

Nella figura 4.35 è mostrato il rapporto delle pressioni ottenute nel punto  $P_1$  con e senza l'inserimento del risonatore. Si nota come l'andamento della pressione rispecchia l'andamento calcolato nella figura 4.28 e riportato per comodità in figura 4.35. Da questa analisi, osservando anche altri punti di misura, si può ritenere che il modello analitico sviluppato sia validato dal punto di vista numerico. L'analisi sperimentale fornirà il riscontro finale al modello presentato.

<sup>4</sup>Il fluido è aria avente le caratteristiche fisiche come velocità del suono e coefficiente di smorzamento viscoso equivalente pari a quelle della tabella 4.3.

### Modellazione del risonatore di side-branch

Analogamente a quanto svolto per la validazione del modello del risonatore di Helmholtz, si esegue la stessa analisi per il risonatore di tipo side-branch. In figura 4.36 è mostrata la rappresentazione del modello del sistema costituito dal condotto e dal risonatore side-branch. Le grandezze del sistema sono pari a quelle della tabella 4.6.

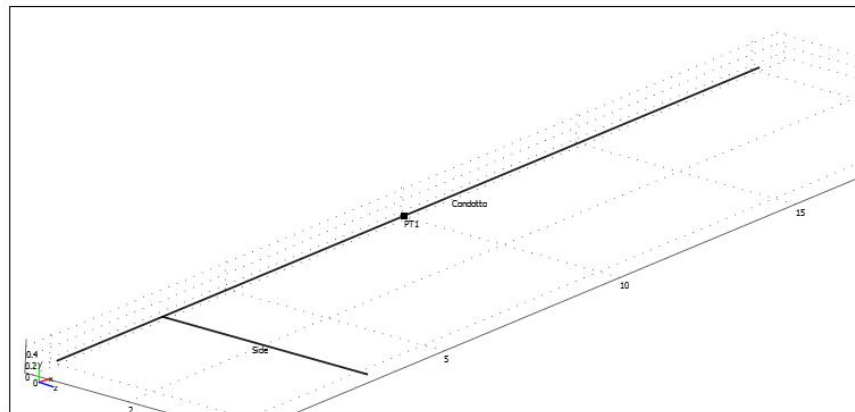


Figura 4.36: *Modello numerico del sistema di analisi del risonatore di Helmholtz.*

Il sistema è costituito da un condotto di lunghezza pari a  $20\text{ m}$  dove alla distanza di  $3\text{ m}$  dall'ingresso è inserito il risonatore side-branch. In ingresso al sistema viene definita la stessa forzante del caso precedente, avente una pressione dinamica variabile di ampiezza pari a  $P_{in} = 10^4\text{ Pa}$  e una frequenza variabile da  $5$  a  $40\text{ Hz}$ ; la pressione dinamica in uscita dal condotto viene fissata a  $P_{out} = 0\text{ Pa}$ . Anche in questo caso lo scopo dell'analisi è quello di calcolare l'andamento delle pressioni nel punto  $P_1$ , che si trova ad una distanza pari a  $9.9\text{ m}$  dall'ingresso del condotto e al centro della sezione dello stesso, e confrontarlo con l'andamento delle pressioni nello stesso punto, del sistema senza il risonatore.

Si definiscono le caratteristiche geometriche del sistema, le condizioni al contorno e la mesh. Gli elementi finiti utilizzati sono i medesimi del caso precedente, il sistema in questo caso risulta composto da  $3497$  elementi e da  $10337$  gradi di libertà.

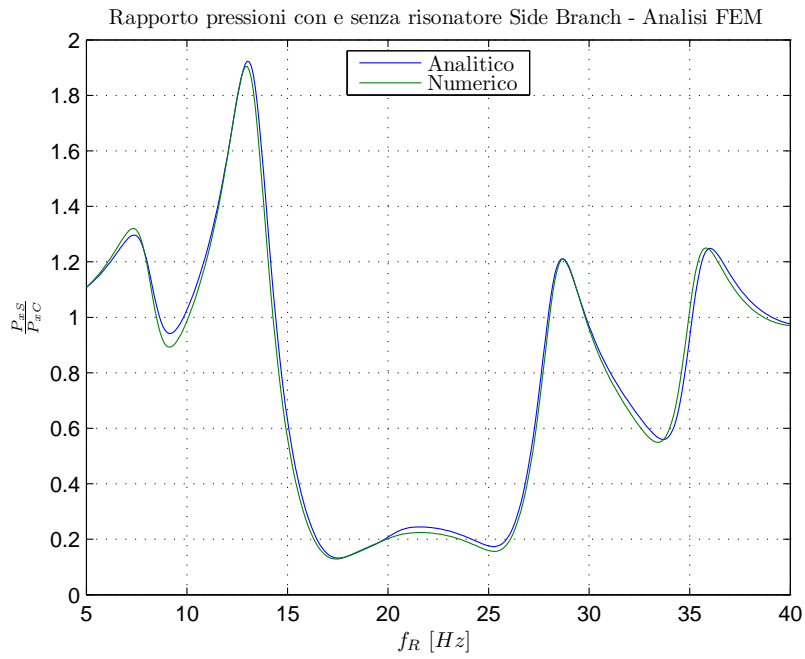


Figura 4.37: Rapporto delle pressioni con e senza risonatore side-branch nel punto  $P_1$ : confronto fra i risultati del modello analitico e del modello numerico.

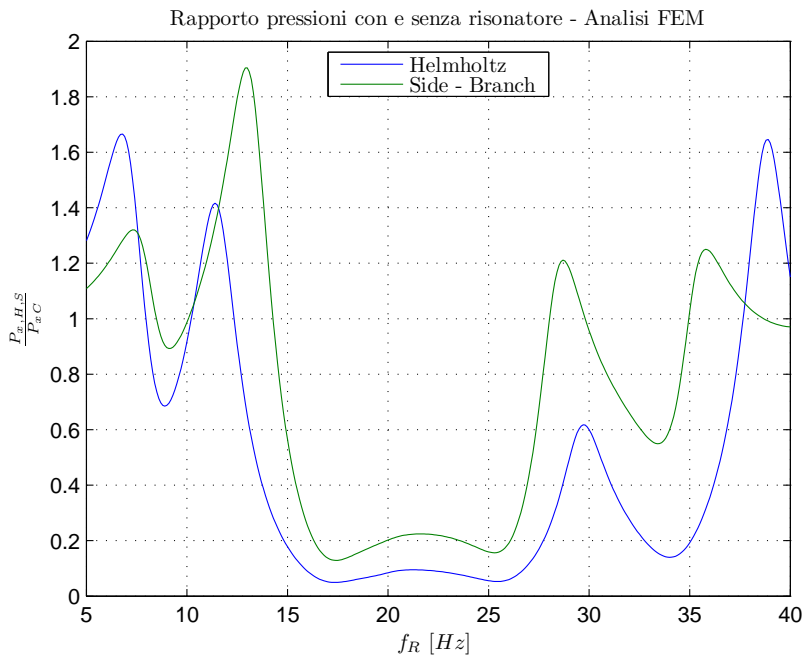


Figura 4.38: Confronto tra il comportamento dinamico del risonatore di Helmholtz e il risonatore side-branch. Risultati dell'analisi numerica

Nella figura 4.37 è mostrato il rapporto delle pressioni ottenute nel punto  $P_1$  con e senza l'inserimento del risonatore. Anche in questo caso si nota come l'andamento della pressione rispecchia l'andamento calcolato nella figura 4.33 e riportato per comodità in figura 4.37. Nella figura 4.38 è rappresentato il confronto tra il comportamento del risonatore di Helmholtz e quello del risonatore side-branch. Valgono le osservazioni descritte nel paragrafo 4.4.2, ovvero il risonatore side-branch presenta un'efficacia minore rispetto al risonatore di Helmholtz. Infatti l'ampiezza della banda in frequenza in cui il risonatore side-branch riesce ad apportare un sostanziale abbattimento delle oscillazioni di pressione è molto ridotta rispetto a quella ottenibile con un risonatore di Helmholtz. Da questa ulteriore analisi, si può ritenere che il modello analitico sviluppato sia validato dal punto di vista numerico.



## Capitolo 5

# Impianto di test per i sistemi di abbattimento

### 5.1 Introduzione al problema

Lo studio effettuato nel capitolo 4 ha messo in luce le relazioni che legano le grandezze fisiche implicate nella propagazione delle onde sonore e il comportamento dinamico di alcuni dispositivi per il loro abbattimento. La propagazione delle onde sonore, dal punto di vista teorico prevede l'assunzione di alcune ipotesi atte ad esempio, a prevedere la propagazione di tali onde lungo una direzione preferenziale (problema monodimensionale), lungo un piano (problema bidimensionale) oppure nello spazio (problema tridimensionale). Appare dunque necessario studiare un caso reale per poter giustificare le ipotesi fatte e per acquisire competenza sperimentale sulla tematica trattata. Nei paragrafi successivi verrà affrontata la misura sperimentale delle oscillazioni di pressione in un compressore industriale; questa misura è necessaria per capire le reali problematiche connesse alle pulsazioni di pressione e per verificare, almeno su di un caso, le cause che portano all'insorgenza di questa oscillazioni di pressione. Questo studio porta alla definizione di un impianto sperimentale su cui effettuare i test dei dispositivi oggetto dello studio. L'impianto sperimentale dovrà essere tale da riprodurre le caratteristiche delle oscillazioni di pressione rilevate dalla campagna sperimentale sull'apparato industriale, e tale da poter essere gestito in un laboratorio di ricerca.

### 5.1.1 Campagna di misura su compressore industriale

I compressori e in generale gli impianti per il trattamento dell'aria possono presentare problematiche legate al rumore e alle oscillazioni di pressione, connesse generalmente al lavoro svolto sul fluido o alla particolare conformazione delle condotte che trasportano il fluido. In questo ambito, i compressori industriali sono quelli che maggiormente risentono delle problematiche legate alle oscillazioni di pressione. Se da un lato gli impianti per il trattamento dell'aria possono risentire di queste problematiche nell'ambito del comfort acustico degli ambienti soggetti al trattamento dell'aria, dall'altra i compressori sono soggetti a problematiche legate a problematiche energetiche e di sollecitazioni meccaniche dei componenti.

L'architettura costruttiva dei compressori industriali varia in funzione dell'impiego per cui essi sono progettati. Le architetture che presentano le maggiori problematiche dal punto di vista delle oscillazioni di pressione sono costituite dalle macchine alternative che effettuano la compressione del fluido mediante pistoni mobili. Le pulsazioni di pressione causano problemi di vibrazioni e quindi di sollecitazioni a fatica sia nei condotti fra i vari stadi di compressione della macchina, sia nelle condutture a valle del sistema di compressione e quindi nell'impianto stesso. Le tecniche finora utilizzate per la riduzione delle pulsazioni di pressione prevedevano ad esempio l'inserimento all'interno del condotto di zone di perdita di carico concentrata<sup>1</sup>, il cui effetto però è anche quello di incrementare l'energia richiesta per effettuare la compressione del fluido. Un'altra tecnica che viene ad esempio citata nella normativa americana API Standard 618, che prescrive i minimi requisiti dei compressori alternativi e dei loro dispositivi [20], è legata all'inserimento di accumulatori (per il cui principio di funzionamento si faccia riferimento al par. 1.4.1) in determinati punti del circuito di compressione. Questa tecnica comporta un reale abbattimento delle pulsazioni di pressione ma un aumento degli ingombri della macchina, non sempre consentito per via della sua trasportabilità. Nasce dunque l'esigenza di trovare dei metodi alternativi che consentano di coniugare le esigenze rivolte al risparmio energetico legato al funzionamento della macchina, con quelle legate alla salvaguardia dei componenti meccanici e della vita utile dell'impianto.

---

<sup>1</sup>All'interno del condotto viene inserito un setto a sezione ridotta, che comporta una perdita di energia concentrata.



In generale è possibile affermare che la riduzione delle pulsazioni di pressione porta ai seguenti benefici:

- aumento della vita utile dell'impianto;
- migliore funzionamento delle sue utenze;
- riduzione degli ingombri complessivi della macchina;
- riduzione del costo della macchina;
- riduzione dei costi associati ad interventi di manutenzione;
- miglioramento del rendimento complessivo della macchina;
- aumento dell'affidabilità della macchina.

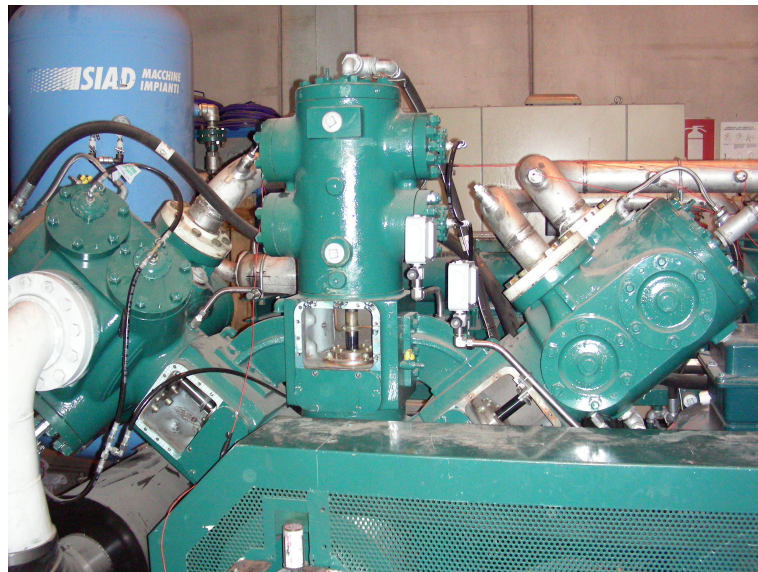


Figura 5.1: *Compressore industriale TEMPO 3.*

La macchina che si intende studiare è un compressore alternativo industriale per la produzione di aria compressa in impianti di produzione di contenitori in PET. Il compressore è il modello TEMPO 3 prodotto da SIAD Macchine e Impianti S.p.A. Esso è costituito da tre stadi di compressione in serie tra loro che effettuano complessivamente un rapporto di compressione 40 : 1. La macchina è movimentata da un motore elettrico asincrono trifase comandato da inverter. In questo modo è possibile variare il regime di rotazione della macchina e quindi la portata volumetrica immessa nell'impianto. In

questo impianto è possibile effettuare anche delle parzializzazioni che consentono di non effettuare la compressione del fluido tra gli stadi, pur mantenendo in moto la macchina. La misura che si intende effettuare riguarda la determinazione del comportamento dinamico del flusso d'aria all'interno dei condotti del compressore.

L'analisi acustica sperimentale viene effettuata per mezzo di un analizzatore che permette di elaborare i segnali acquisiti dai sensori ad esso collegati. I segnali acquisiti vengono successivamente rielaborati in modo da conoscere le principali caratteristiche del flusso in termini variazione di pressione e temperatura nel tempo, quindi delle frequenze associate alle pulsazioni di pressione generate dal compressore. Questa fase permette di conoscere il comportamento dinamico del sistema al variare di alcuni parametri della macchina quali ad esempio la portata, oppure al variare della temperatura del flusso. L'importanza della conoscenza del comportamento sperimentale del sistema è associata alla possibilità di effettuare una validazione del modello analitico/numerico (definito nel cap. 4) in termini di accuratezza nella descrizione dei risultati. Un modello analitico o numerico che descrive accuratamente il reale comportamento del sistema ossia, la conoscenza accurata dei parametri del sistema che governano la sua dinamica, è uno dei requisiti essenziali per la progettazione di un buon sistema di controllo.

### 5.1.2 Definizione della catena di misura

La campagna di misura sperimentale ha una duplice valenza: da un lato permette di rilevare alcune grandezze sul sistema e di stabilire quali di queste sono maggiormente implicate nella problematica in oggetto, dall'altro permette di valutare l'entità del problema in funzione delle condizioni operative della macchina.

Le grandezze rilevate durante la campagna di misura sono state:

- Pressione dinamica in differenti punti dei condotti di aspirazione e mandata.
- Velocità istantanea di rotazione dell'albero del compressore.
- Pressione statica negli stadi del compressore.
- Temperatura negli stadi del compressore.
- Potenza elettrica assorbita e sfasamento tra tensione e corrente.

Per poter acquisire tutte queste grandezze è necessario definire un'adeguata catena di misura. Per quanto riguarda la misura delle grandezze come la pressione statica, la

temperatura tra gli stadi, la potenza elettrica assorbita e lo sfasamento tra tensione e corrente vengono acquisite attraverso la strumentazione già presente a bordo del compressore. Nella presente trattazione si farà riferimento prevalentemente all'analisi delle altre grandezze, pressione dinamica e velocità istantanea, per le quali è necessario allestire una opportuna catena di misura.

La catena di misura per la rilevazione delle pressioni dinamiche è composta dai trasduttori di pressione, che trasformano le oscillazioni di pressione in variazioni di carica, avendo come elemento sensibile un elemento piezo-elettrico. La scelta di tali trasduttori è dettata dal fatto che le temperature del fluido compresso potrebbero raggiungere quelle che si sviluppano nella compressione di un gas ideale. Dalla preventiva osservazione della macchina si sono osservate temperature del fluido in alcuni punti prossime a  $150^{\circ}\text{C}$ : queste temperature risultano deleterie per il sistema di amplificazione dei comuni trasduttori di pressione che trasformano la variazione di pressione in una variazione di tensione elettrica. È stato dunque deciso di porre dei trasduttori di pressione che trasformassero la variazione di pressione in una variazione di carica elettrica, in modo tale da poter disporre la parte di amplificazione del segnale in prossimità dei condotti ma non a contatto di essi. I sensori di pressione e i relativi amplificatori di carica sono stati scelti in modo da poter osservare il range di frequenze e di ampiezze che si riteneva poter essere presente durante il funzionamento della macchina, considerando questo range sufficientemente ampio. Nella tabella 5.1 vengono elencati gli otto sensori di pressione utilizzati per eseguire la campagna di misura, costruiti da Dytran. I primi quattro presentano un fondoscala pari a  $100\text{ psi} \approx 689\text{ kPa}$ , e sono adatti alla rilevazione delle pressioni dinamiche nei primi stadi di compressione della macchina, mentre gli ultimi quattro presentano un fondo scala pari a  $1000\text{ psi} \approx 6894\text{ kPa}$  e sono più adatti a rilevare le oscillazioni di pressione negli ultimi stadi di compressione. A monte degli amplificatori di carica, prima di acquisire il segnale nella opportuna strumentazione, i segnali dei trasduttori di pressione sono stati opportunamente amplificati e filtrati per mezzo un amplificatore di segnale con integrati dei filtri analogici. L'amplificatore utilizzato è il Bruel & Kjaer NEXUS 2690 (fig. 5.2).

Modello	Sensore pressione			Amplificatore di carica		Sensibilità $\left[\frac{mV}{Pa}\right]$	Max output level [V]	Sensibilità Output Nexus $\left[\frac{mV}{Pa}\right]$	Max output level Nexus [V]	
	S/N	FS [psi]	Sensibilità $\left[\frac{mV}{psi}\right]$	Modello	S/N					Sensibilità $\left[\frac{mV}{pC}\right]$
2200C1	2291	100	1.02	4751B1	3530	52.50	7.7668 · 10 <sup>-3</sup>	5.36	10 <sup>-2</sup>	6.89
2200C1	2293	100	0.98	4751B1	3532	52.50	7.4622 · 10 <sup>-3</sup>	5.15	10 <sup>-2</sup>	6.89
2200C1	2294	100	0.99	4751B1	3619	51.50	7.3947 · 10 <sup>-3</sup>	5.10	10 <sup>-2</sup>	6.89
2200C1	2298	100	1.04	4751B1	3622	51.40	7.7531 · 10 <sup>-3</sup>	5.35	10 <sup>-2</sup>	6.89
2200C4	2448	1000	1.06	4751B3	4113	1.00	1.5374 · 10 <sup>-4</sup>	1.06	10 <sup>-3</sup>	6.89
2200C4	2449	1000	0.99	4751B3	4114	1.00	1.4359 · 10 <sup>-4</sup>	0.99	10 <sup>-3</sup>	6.89
2200C4	2450	1000	1.12	4751B3	4115	1.00	1.6244 · 10 <sup>-4</sup>	1.12	10 <sup>-3</sup>	6.89
2200C4	2451	1000	1.06	4751B3	4116	1.00	1.5374 · 10 <sup>-4</sup>	1.06	10 <sup>-3</sup>	6.89

Tabella 5.1: Caratteristiche dei sensori di pressione, degli amplificatori di carica e dell'amplificatore NEXUS, utilizzati per la campagna di misura su compressore industriale.

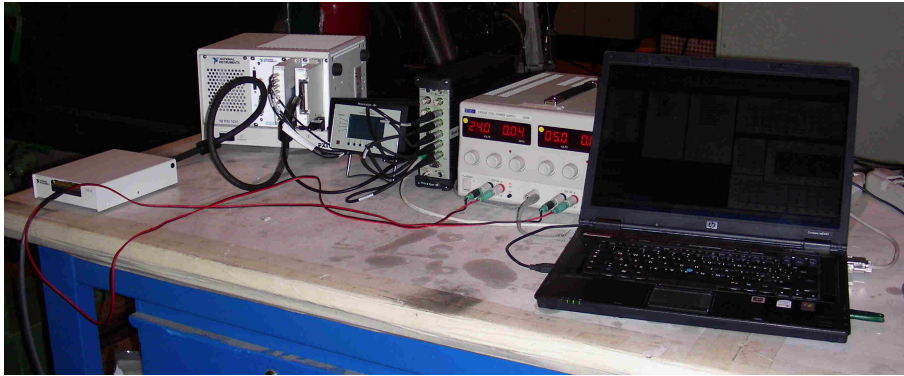


Figura 5.2: *Strumentazione utilizzata per la campagna di misura su compressore industriale. Da sinistra a destra: NI PXI, Bruel & Kjaer NEXUS 2690, Bruel & Kjaer PULSE 3560B, Alimentatore per encoder, PC per registrazione dati.*

Per eseguire l'acquisizione dei segnali vengono utilizzati due analizzatori sincronizzati tra loro attraverso un opportuno segnale di trigger. I due sistemi di acquisizione sono il sistema di acquisizione general purpose NI PXI e l'analizzatore Bruel & Kjaer PULSE 3560B (mostrati in figura 5.2). Le prestazioni di tali sistemi di misura, descritte nell'appendice B.1, consentono di effettuare in modo adeguato le misure necessarie sul sistema. Per poter eseguire la rilevazione della velocità istantanea di rotazione dell'albero del compressore si utilizza un encoder calettato su di esso, in particolare è stato scelto il modello IH103 prodotto da GES, avente una risoluzione pari a 1000 impulsi per giro e un circuito di output di tipo push-pull. Il segnale dell'encoder è rilevato tramite il sistema di acquisizione NI PXI e sincronizzato con gli altri segnali rilevati. Dato che la velocità di rotazione dell'albero del compressore può variare da circa 550 *rpm* a 1500 *rpm*, e si vogliono analizzare le componenti spettrali multiple della frequenza di rotazione, appare adeguato scegliere una frequenza di campionamento pari a:

$$f_c = 4096 \text{ Hz}. \quad (5.1)$$

Di conseguenza la frequenza di taglio dei filtri analogici è stata impostata a  $f_{cut} = 1000 \text{ Hz}$ , ben al di sotto della frequenza di Nyquist  $f_{Ny} = f_c/2$ , poiché i filtri analogici utilizzati sono filtri del primo ordine e si vuole assicurare un adeguato abbattimento del segnale in corrispondenza della frequenza di Nyquist, per evitare fenomeni di aliasing.

### 5.1.3 Definizione delle prove sperimentali

Come descritto nel paragrafo 5.1.1, si vuole indagare l'andamento delle oscillazioni di pressione lungo i condotti del compressore. A tale scopo sono stati previsti opportuni bocchelli, che consentono la misura della pressione in 19 punti, suddivisi fra aspirazioni e mandate dei tre stadi.

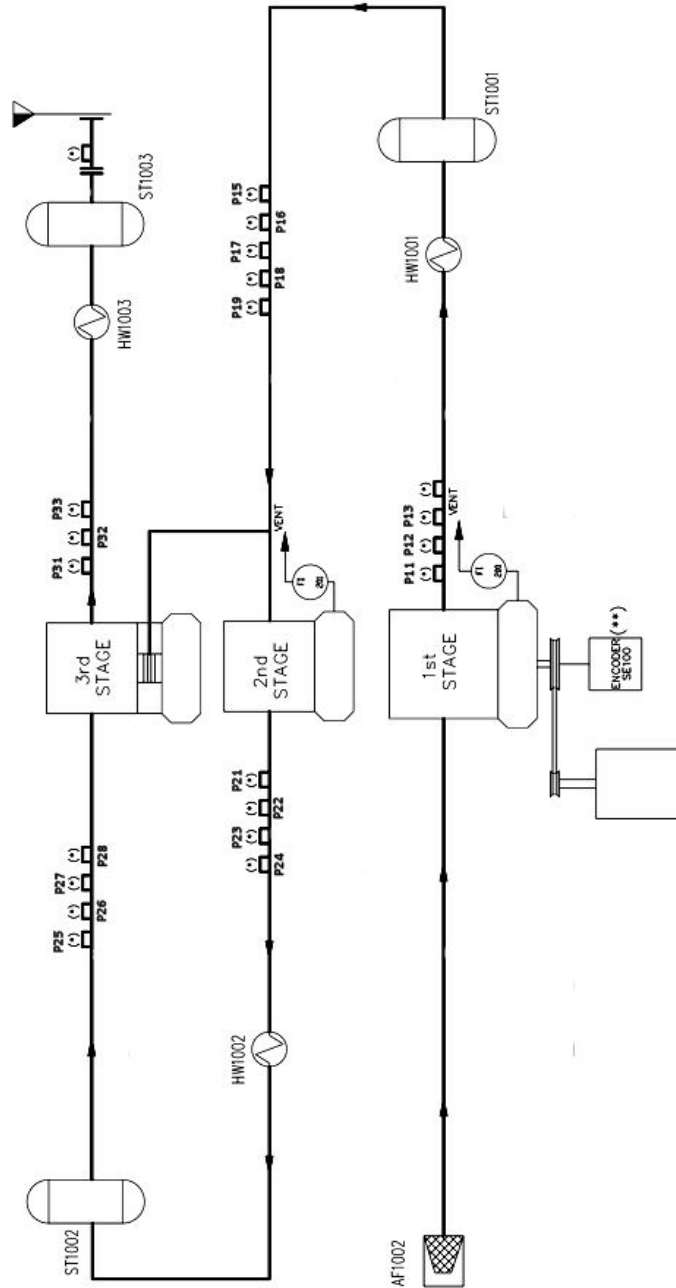


Figura 5.3: Schema dei punti di misura su compressore industriale TEMPO 3.

Un quadro d'insieme dei punti di misura può essere dato dalla figura 5.3, mentre nelle figure 5.4, 5.5, 5.6, 5.7 e 5.8 sono mostrati le conformazioni reali dei condotti e le posizioni fisiche dei sensori. Particolare attenzione deve essere posta durante la rilevazione della pressione in tratti non rettilinei, in quanto il flusso può avere dei comportamenti dei comportamenti dinamici che sono influenzati dalla forma del condotto.

Nella figura 5.3 sono indicati con HW1001, HW1002 e HW1003 le posizioni dei sensori di temperatura, mentre con  $P11 \div P33$  le posizioni dei sensori di pressione. Tra gli stadi del compressore sono anche presenti le unità di raffreddamento, indicate con ST1001, ST1002 e ST1003, necessarie per abbassare la temperatura del fluido prima della successiva compressione o immissione nell'impianto. Dopo aver subito la fase di compressione, il fluido viene fatto sfociare in ambiente, e la pressione finale viene regolata attraverso una valvola. Per ridurre la rumorosità e per evitare di subire danni alla macchina, alla mandata del primo stadio è previsto un filtro di aspirazione, che viene mantenuto anche durante l'esecuzione delle prove. Con questa configurazione di impianto è possibile studiare il comportamento dinamico del compressore senza l'influenza di un impianto ad esso connesso.

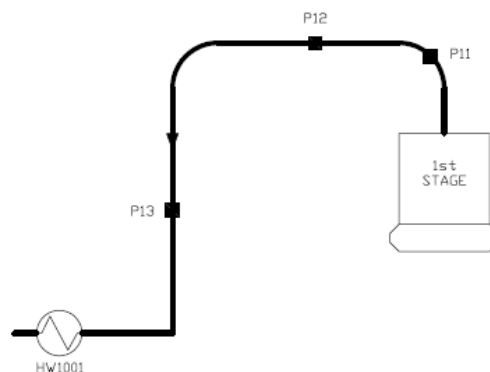


Figura 5.4: Posizione dei punti di misura su compressore industriale: mandata 1° stadio.

Dopo aver definito questa configurazione di punti di misura sono state determinate la tipologia di prove da effettuare:

- Misura a pressione media pari a  $2000 \text{ kPa}$  in condizioni di massimo RPM.
- Misura a pressione media pari a  $4000 \text{ kPa}$  in condizioni di massimo RPM (condizione nominale).

- Misura a pressione media pari a 4000  $kPa$  con mantenimento della pressione a differenti regimi di rotazione.
- Misura con parzializzazione al 100%: pressione media nulla (pressione ambiente).

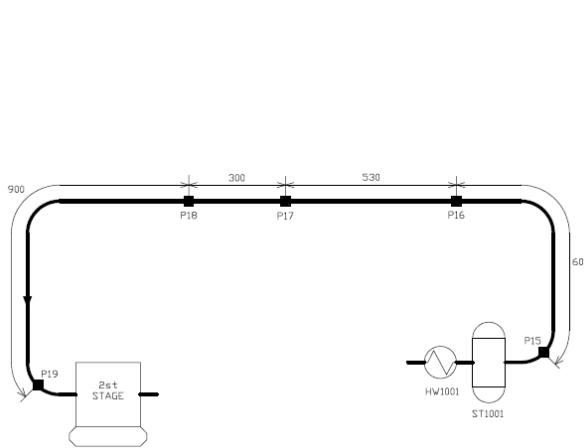


Figura 5.5: Posizione dei punti di misura su compressore industriale: aspirazione 2° stadio.

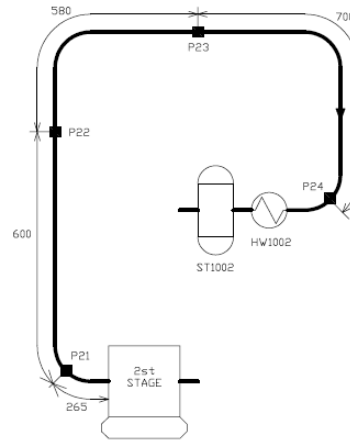


Figura 5.6: Posizione dei punti di misura su compressore industriale: mandata 2° stadio.

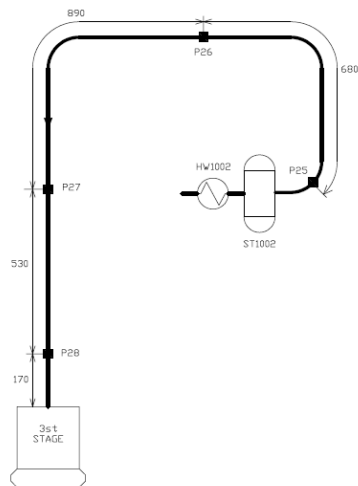


Figura 5.7: Posizione dei punti di misura su compressore industriale: aspirazione 3° stadio.

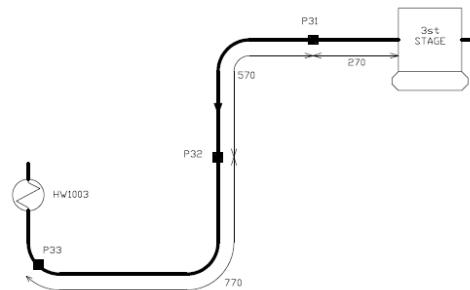


Figura 5.8: Posizione dei punti di misura su compressore industriale: mandata 3° stadio.



Le prove sono state effettuate al variare del numero di giri del motore, come mostrato in tabella 5.2. Il rapporto di riduzione tra l'albero motore e l'albero del compressore è pari a  $i = 1.2533$ .

RPM motore desiderati							
56	608	670	732	794	856	918	980
1042	1104	1166	1228	1290	1352	1414	1476

Tabella 5.2: *Elenco di velocità di rotazione del motore del compressore in cui vengono eseguite le misure.*

Sono state individuate nove configurazioni per la disposizione dei sensori sulla macchina, definiti "gruppi di misura". La conformazione dei gruppi di misura è mostrata in tabella 5.3. Sono indicati con  $P$  i sensori della serie 2200C1, che vengono acquisiti l'analizzatore PULSE, mentre sono indicati con  $N$  i sensori della serie 2200C4 che vengono acquisiti con il sistema NI PXI.

I gruppi di misura sono stati scelti in modo tale da avere delle misure che comprendessero contemporaneamente tutti condotti di aspirazione e mandata della macchina (gruppo 1) o misure che analizzassero l'andamento delle oscillazioni di pressione all'interno dello stesso condotto (gruppi 2, 3 e 4). Si è voluto ulteriormente indagare l'andamento delle pressioni a cavallo del singolo stadio (gruppi da 5 a 9) sia considerando che escludendo dalla misura i tratti in cui sono interposte le unità di raffreddamento del fluido. Per ogni gruppo di misura sono state eseguite le misure alle varie condizioni operative e per i vari regimi di rotazione dell'albero del compressore, in modo tale avere un quadro completo del comportamento dinamico della macchina.

Gruppi di Misura	Stadio 1				Stadio 2								Stadio 3						
	P11	P12	P13	P15	P16	P17	P18	P19	P21	P22	P23	P24	P25	P26	P27	P28	P31	P32	P33
1	N1	N2	N2	N4	N3		N4		P1	P1	P3	P4		P2				P3	P4
2	N1	N2	N3	N4									P1	P2					
3					N1	N1	N3	N4								P4	P1	P2	P3
4		N1	N2			N3	N3	P4							N4				
5					N1	N2	N3	N4	P1	P2	P3	P4							
6												P4	N1	N2	N3	N4	P1	P2	P3
7	N4	N1	N2	P1	P2	P3	N3	P4											
8								N4	P1	P2	P3	P4	N1	N2	N3	N4			
9								N4	N3	N1	P1					P2	P4	N1	P3

Tabella 5.3: Raggruppamento dei sensori in "gruppi di misura" per l'esecuzione della campagna di misura su compressore industriale.

### 5.1.4 Analisi dei risultati sperimentali

La campagna di misura sperimentale ha portato all'esecuzione di 368 acquisizioni, nelle diverse condizioni operative. Per l'analisi delle acquisizioni si è sviluppato un opportuno script di MATLAB, disponibile nell'appendice E.1 che consente la "navigazione" all'interno delle prove effettuare e la selezione delle analisi da effettuare sul segnale o sui gruppi di segnali acquisiti. Nel seguito, a titolo di esempio si mostrano alcune analisi svolte sui segnali acquisiti, con lo scopo di dare una visione globale dei risultati ottenuti, che hanno poi consentito lo sviluppo dell'applicazione oggetto di studio.

#### Corrispondenza tra valori teorici e sperimentali

Le prime analisi effettuate sui segnali acquisiti riguardano l'analisi nel dominio del tempo. In particolare si è osservato il segnale e si è valutato l'andamento della pressione sull'angolo giro di rotazione dell'albero del compressore. Per esigenze legate alla proprietà intellettuale, la rappresentazione di alcune grandezze rilevate avverrà in forma adimensionale: questa rappresentazione consente comunque di poter trarre utili informazioni in questa trattazione. Indicazioni sulle adimensionalizzazioni effettuate sono indicate nelle note a piè pagina.

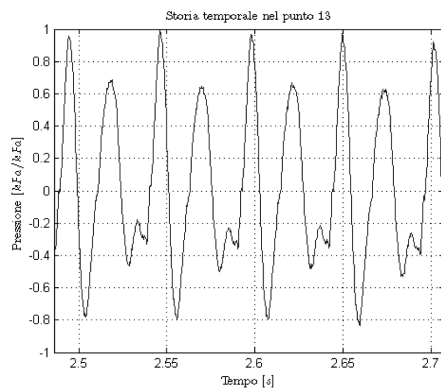


Figura 5.9: Campagna di misura su compressore industriale: andamento temporale delle pressioni nel punto P13.

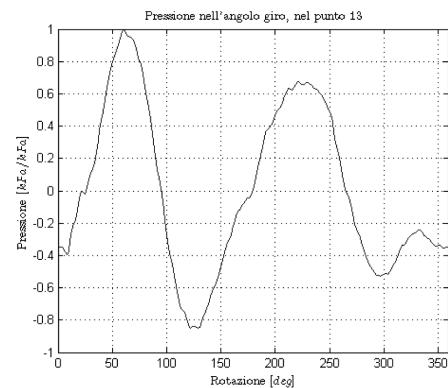


Figura 5.10: Campagna di misura su compressore industriale: andamento della pressione sull'angolo giro (punto P13).

Nella figura 5.9 è riportato l'andamento temporale della pressione rilevata in queste condizioni:

- Gruppo di misura:  $G1$ .
- Tipologia di prova: Prova in condizioni nominali.
- Punto di misura:  $P13$ .
- Durata record: 20 s.
- RPM motore:  $1476 \frac{giri}{min}$  (impostati).
- RPM medi motore:  $1452 \frac{giri}{min}$ .
- RPM albero:  $1158 \frac{giri}{min}$ .

Si nota come l'andamento temporale presenti delle periodicità, che possono essere studiate mediante un'analisi in frequenza del segnale stesso<sup>2</sup>. Nella figura 5.10 la storia temporale mostrata nella figura 5.9 è stata divisa in sottorecord della lunghezza pari alla rotazione completa dell'albero del compressore. Tale suddivisione è possibile perché l'acquisizione dei segnali in pressione sono sincrone con quelle del segnale dell'encoder. I sottorecord sono stati tra loro mediati, con lo scopo di ridurre il rumore random che poteva affliggere il segnale. Il risultato mostra un andamento della pressione che rispecchia le caratteristiche costruttive del compressore: infatti essendo a doppio effetto, si possono individuare le due fasi di compressione del fluido, rispettivamente a circa  $60^\circ$  e a  $220^\circ$  di rotazione rispetto al punto morto superiore del pistone centrale<sup>3</sup>. Dalle analisi nel dominio del tempo risulta molto complicato fare valutazioni legate alla composizione armonica delle pulsazioni di pressione. Pertanto si è eseguita l'analisi spettrale su tutti i segnali rilevati.

Il regime di rotazione dell'albero del compressore per questa acquisizione è pari a  $RPM_{albero} = 1158 \frac{giri}{min}$ . La frequenza di rotazione corrispondente è pari a:

$$f_{1x} = \frac{RPM_{albero}}{60s} = 19.3 \text{ Hz}. \quad (5.2)$$

Le frequenze teoriche multiple della fondamentale sono descritte nella tabella 5.4.

<sup>2</sup>Il segnale proposto è stato adimensionalizzato rispetto al valore di pressione massimo rilevato all'interno del record.

<sup>3</sup>Adimensionalizzazione eseguita rispetto al valore massimo della pressione risultante.

Indice	Frequenza [Hz]
1 x giro	19.3
2 x giro	38.6
3 x giro	57.9
4 x giro	77.2
5 x giro	96.5
6 x giro	115.8

Tabella 5.4: Frequenze multiple della frequenza di rotazione dell'albero del compressore.

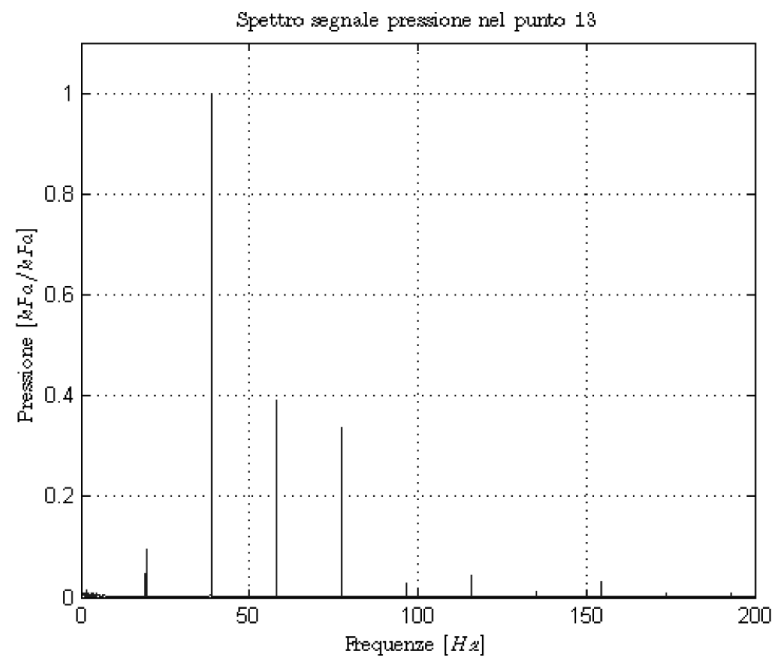


Figura 5.11: Campagna di misura su compressore industriale: analisi spettrale su un'acquisizione nel punto P13.

In figura 5.11 si può notare come le componenti spettrali del segnale analizzato siano analoghe a quelle teoriche della tabella 5.4, evidenziando un buon riscontro tra l'andamento teorico e i risultati sperimentali<sup>4</sup>. In figura 5.12 è mostrato lo stesso spettro di figura 5.11 ma con l'asse delle ascisse adimensionalizzato rispetto alla frequenza di rotazione dell'albero del compressore, o *frequenza fondamentale*.

<sup>4</sup>Adimensionalizzazione eseguita rispetto al valore massimo delle componenti spettrali.

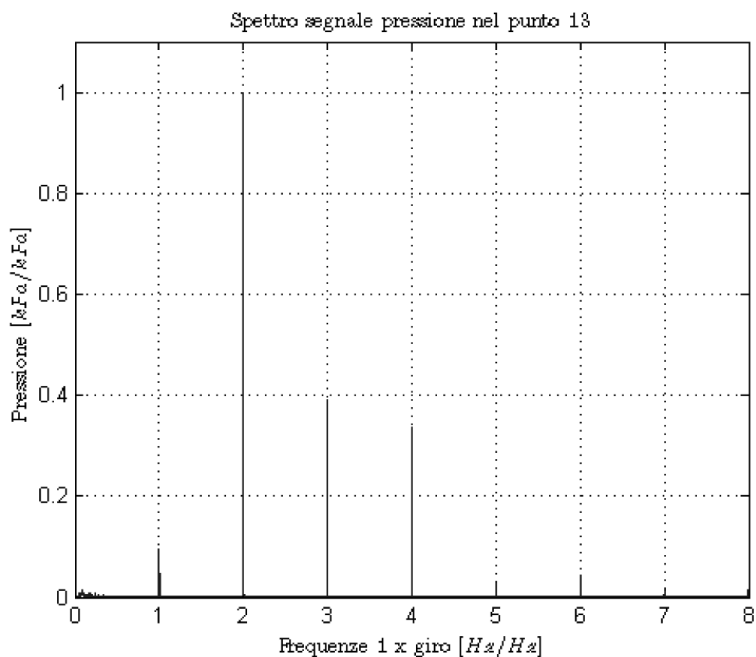


Figura 5.12: *Campagna di misura su compressore industriale: analisi spettrale e rappresentazione delle frequenze in multipli della frequenza di rotazione dell'albero del compressore.*

Anche in questo caso si nota maggiormente la buona aderenza tra risultati sperimentali e calcoli teorici: non esistono infatti componenti di frequenza spurie di ampiezza significativa nel range di interesse. Tali osservazioni sono state effettuate per tutti i punti e tutte le condizioni di funzionamento, riportando sempre risultati analoghi a quelli mostrati.

### Verifica di stazionarietà del segnale

Occorre verificare che le componenti armoniche rilevate dall'analisi spettrale siano stazionarie e costanti nel tempo. Per questa verifica occorre eseguire uno spettrogramma sul record acquisito, che permette la verifica di stazionarietà. Il risultato di questa analisi è riportato in figura 5.13. Lo spettrogramma delle frequenze<sup>5</sup> mostra un andamento costante delle ampiezze spettrali, pertanto si può concludere che l'andamento delle pulsazioni di pressione sia stazionario. L'analisi è stata condotta per tutti i punti, per tutte le condizioni di funzionamento e per tutti i regimi di rotazione.

<sup>5</sup>Adimensionalizzazione eseguita rispetto al valore massimo delle componenti spettrali.

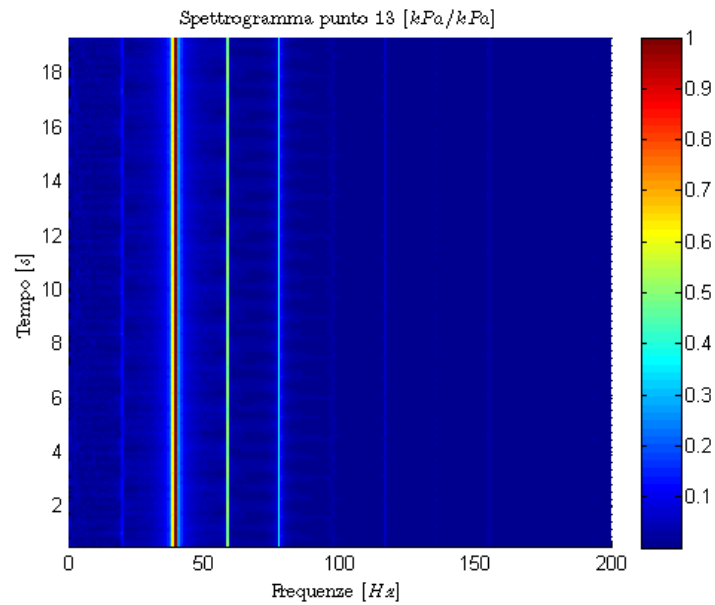


Figura 5.13: Campagna di misura sperimentale su compressore industriale: spettrogramma del punto P13.

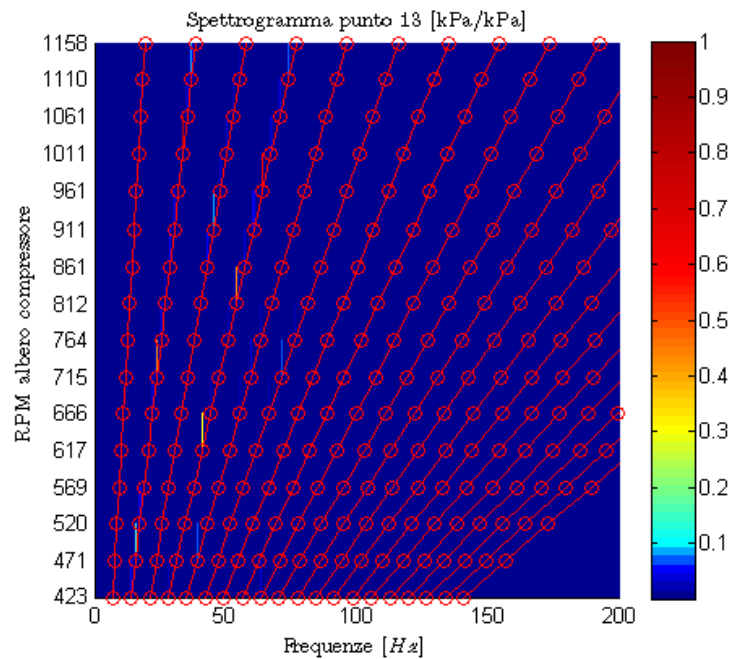


Figura 5.14: Campagna di misura sperimentale su compressore industriale: spettrogramma del punto P13 per tutti i regimi di rotazione. In rosso vengono indicate le frequenze di pulsazione teoriche.

In figura 5.14 viene rappresentato l'andamento spettrale delle oscillazioni di pressione nel punto  $P13$  in funzione del regime di rotazione dell'albero motore. si può notare come le componenti spettrali siano analoghe a quelle teoriche calcolate e indicate in figura con linee rosse, ovvero non vi è la presenza di componenti spettrali non multiple della fondamentale.

Accanto all'analisi delle pulsazioni di pressione è stato verificato anche l'andamento della velocità istantanea dell'albero del compressore. In figura 5.15 è mostrato l'andamento della velocità istantanea di rotazione, ottenuto derivando numericamente il segnale dell'encoder: si nota come anche la velocità istantanea non sia costante, ma abbia delle componenti in frequenza, ma la variazione di velocità è comunque minore del 2 %, pertanto risulta poco influente sul comportamento dinamico della macchina.

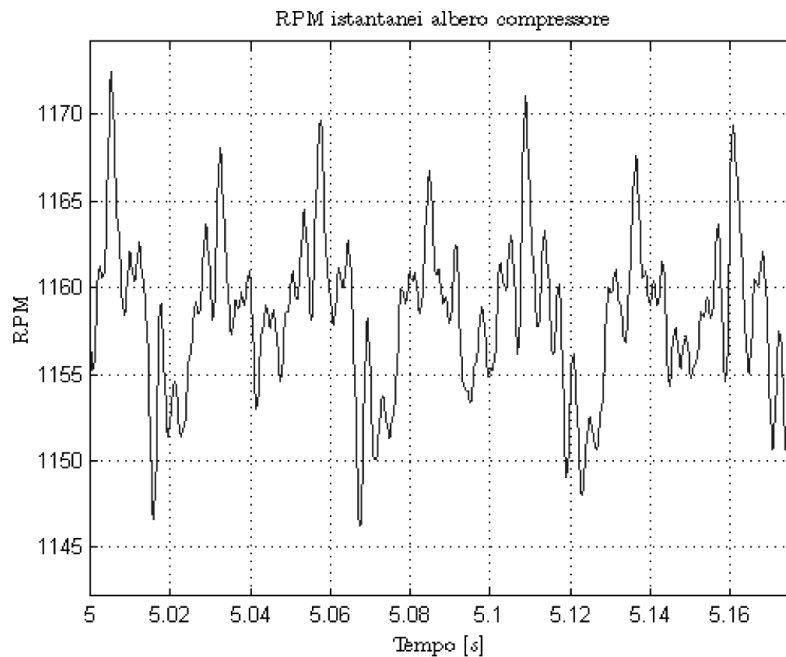


Figura 5.15: *Campagna di misura sperimentale su compressore industriale: velocità istantanea di rotazione dell'albero del compressore.*

### Stazionarietà delle pulsazioni di pressione

Osservando un condotto in cui sono presenti più punti di misura, si è notato come le ampiezze delle componenti spettrali delle pressioni siano molto diverse tra loro, nei diversi punti di misura dello stesso condotto. Si considerino le figure 5.16, 5.17, 5.18 e 5.19.



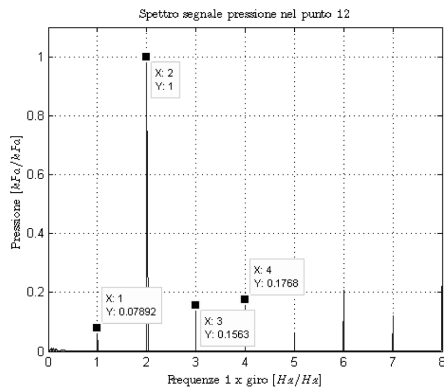


Figura 5.16: *Campagna di misura sperimentale su compressore industriale: spettro delle pulsazioni di pressione nel punto P12 per l'acquisizione effettuata nel gruppo G2.*

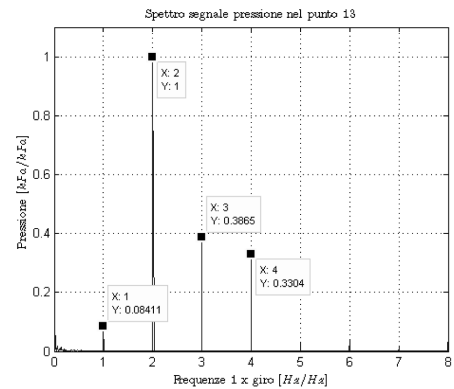


Figura 5.17: *Campagna di misura sperimentale su compressore industriale: spettro delle pulsazioni di pressione nel punto P13 per l'acquisizione effettuata nel gruppo G2.*

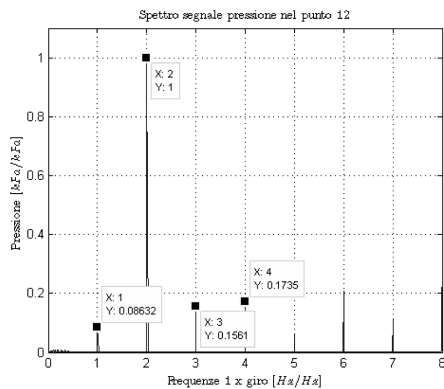


Figura 5.18: *Campagna di misura sperimentale su compressore industriale: spettro delle pulsazioni di pressione nel punto P12 per l'acquisizione effettuata nel gruppo G4.*

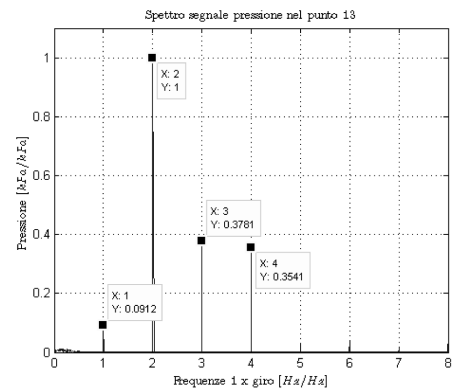


Figura 5.19: *Campagna di misura sperimentale su compressore industriale: spettro delle pulsazioni di pressione nel punto P13 per l'acquisizione effettuata nel gruppo G4.*

Esse rappresentano gli spettri delle pulsazioni di pressione<sup>6</sup> nei punti P12 e P13, per le acquisizioni effettuate nei gruppi G2 e G4. Si nota come tra i punti P12 e P13 vi sia una netta differenza nell'ampiezza tra i vari picchi spettrali, sia nell'acquisizione svolta nel gruppo G2 sia in quella svolta nel gruppo G4. Appare opportuno verificare che l'andamento di tali differenze nelle ampiezze spettrali delle pulsazioni di pressione sia costante nelle varie prove effettuate. In figura 5.20 si mostra il rapporto<sup>7</sup> tra le

<sup>6</sup> Adimensionalizzazione eseguita rispetto al valore massimo delle componenti spettrali.

<sup>7</sup> Il rapporto tra le grandezze viene eseguito sulle pressioni non adimensionalizzate.

ampiezze dei picchi spettrali dei punti P12 e P13, per i vari gruppi di misura che hanno eseguito la misura di questi punti, nella medesima condizione di funzionamento. Dalla figura si nota come tali rapporti vengano mantenuti costanti nelle varie acquisizioni effettuate.

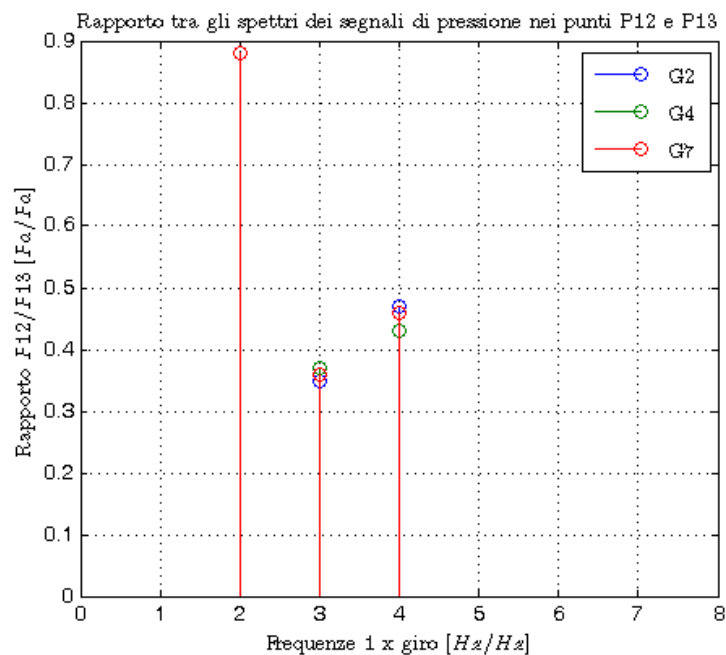


Figura 5.20: Campagna di misura sperimentale su compressore industriale: rapporto tra i picchi spettrali dei segnali rilevati nei punti P12 e P13, per i vari gruppi di acquisizione.

### Conclusioni dell'indagine sperimentale

La grande quantità di misure effettuate ha reso possibile l'esecuzione della mappatura del comportamento dinamico della macchina in analisi, rispetto ai diversi regimi e condizioni di funzionamento. Sulle acquisizioni rilevate sono state effettuate analisi sia nel dominio del tempo che della frequenza, ricercando corrispondenze tra i valori teorici e sperimentali, ed evidenziando le stazionarietà dei fenomeni dinamici.

Le osservazioni finali possono essere riassunte nel seguente elenco:

- le frequenze delle pulsazioni di pressione sono multiple intere della velocità di rotazione dell'albero del compressore;
- vi è corrispondenza tra valori teorici e sperimentali;

- le analisi condotte mostrano la presenza dei fenomeni pulsatori in tutti gli stadi;
- i fenomeni risultano stazionari: nello stesso punto di misura le pulsazioni sono sempre a regime (frequenza e ampiezza fissa);
- le ampiezze delle pulsazioni (in funzione delle frequenze) rimangono costanti tra più misure effettuate nelle medesime condizioni operative;
- sono costanti anche i rapporti tra le ampiezze delle pulsazioni per una determinata condizione operativa;
- i valori delle ampiezze delle pulsazioni dipendono sia dal condotto in cui viene effettuata la misura, sia dal punto in cui viene effettuata la misura;
- le frequenze delle pulsazioni  $2X_{\text{giro}}$  e  $4X_{\text{giro}}$  mostrano le maggiori ampiezze, in accordo all'architettura costruttiva del compressore.

## 5.2 Progettazione e caratterizzazione di un impianto di test

Dal capitolo 5.1.4 si evince che il problema delle pulsazioni di pressione all'interno dei condotti riveste ad oggi un interesse rilevante a causa delle implicazioni sulla vita utile degli impianti nonché sull'affidabilità dei componenti e sul costo dell'impianto. Le normative internazionali propongono spesso delle possibili tecniche per l'implementazione di filtri per l'attenuazione dei fenomeni di interesse in particolari tipologie di sistemi industriali, tuttavia è necessario prescindere da tali sistemi in modo da adeguare le tecniche proposte dalle normative al reale caso di interesse.

Questo studio prevede la valutazione di un sistema per il controllo semi-attivo delle pulsazioni di pressione generate da compressore alternativo all'interno di un condotto a partire dall'analisi delle tecniche passive di abbattimento di tali fenomeni proposte dalla normativa [20]. Con il presente studio si vuole sviluppare un prototipo in cui viene messa in evidenza la capacità di un apposito sistema di controllo di abbattere i fenomeni non desiderati sulla macchina; il fine è quello di migliorare il comportamento complessivo della macchina quindi ridurre le pulsazioni delle onde di pressione generate dal normale funzionamento della macchina tenendo in particolare considerazione i fenomeni delle perdite di carico legate alla presenza dei componenti aggiunti al compressore. Le conseguenze associate all'implementazione di apposite tecniche

di controllo si ripercuotono nel conseguente aumento della vita utile dell'impianto e nel migliore funzionamento delle sue utenze grazie alla riduzione delle sollecitazioni e quindi della fatica a cui risultano sottoposti gli elementi della macchina; inoltre, l'implementazione delle sopra citate tecniche di controllo potrebbe portare anche ad un miglioramento complessivo del rendimento della macchina. L'aumento della vita utile della macchina, o comunque la riduzione degli interventi di manutenzione sulla stessa, e quindi il miglioramento del rendimento dell'intero sistema, permettono di aumentare l'affidabilità e quindi ridurre i costi complessivi associati agli interventi manutentivi. Poiché negli impianti tradizionali vengono già impiegati dei componenti per ridurre le pulsazioni di pressioni, un risultato implicito dello studio sarà quello della riduzione degli ingombri di questi componenti della macchina e quindi della macchina stessa nel suo complesso: l'attenuazione delle pulsazioni delle onde di pressione permetterebbe ad esempio di ridurre le dimensioni degli accumulatori inizialmente dimensionati per attenuare tali fenomeni. La riduzione della dimensione degli accumulatori e quindi dell'ingombro complessivo della macchina porta infine necessariamente ad una riduzione dei costi complessivi della macchina stessa.

Le normative internazionali considerano solamente tecniche passive il cui funzionamento ottimale è limitato a piccole variazioni delle condizioni nominali nelle quali è stato progettato il sistema di attenuazione. L'utilizzo di sistemi risonanti passivi per l'attenuazione delle pulsazioni di pressione non richiede l'ulteriore introduzione di energia nel sistema, risultando di per sé un sistema economico sia nella realizzazione che nell'utilizzo. Il principale svantaggio legato all'utilizzo dei sistemi passivi è legato al fatto che essi devono essere progettati per funzionare in determinate condizioni operative; variazioni delle condizioni operative possono determinare elevati decadimenti delle prestazioni del sistema di attenuazione. Lo scopo di questo studio è quello di indagare la possibilità di introdurre sistemi semi-attivi di attenuazione delle pulsazioni che consentono di ottenere prestazioni elevate al variare delle condizioni operative e con il minimo consumo di energia. Per garantire il minimo dispendio di energia, il sistema di attuazione non agisce direttamente sul sistema da controllare ma sul sistema di attenuazione passivo, modificandone le caratteristiche dinamiche ed adattandole alle variazioni delle condizioni operative.

Per effettuare lo studio occorre definire e realizzare un impianto di test che riproduca i fenomeni misurati su un compressore industriale e che sia flessibile in modo tale che possano essere effettuate prove in diverse condizioni operative e in diverse configurazioni

dell'impianto e del sistema di attuazione. Nel seguito verrà presentato l'impianto di test realizzato e la sua caratterizzazione sperimentale.

### 5.2.1 Progettazione dell'impianto

Il principio di funzionamento dei compressori alternativi utilizzati negli impianti industriali è tale da generare portate discontinue, che possono essere la fonte di problematiche legate alle pulsazioni di pressione nei condotti del sistema.

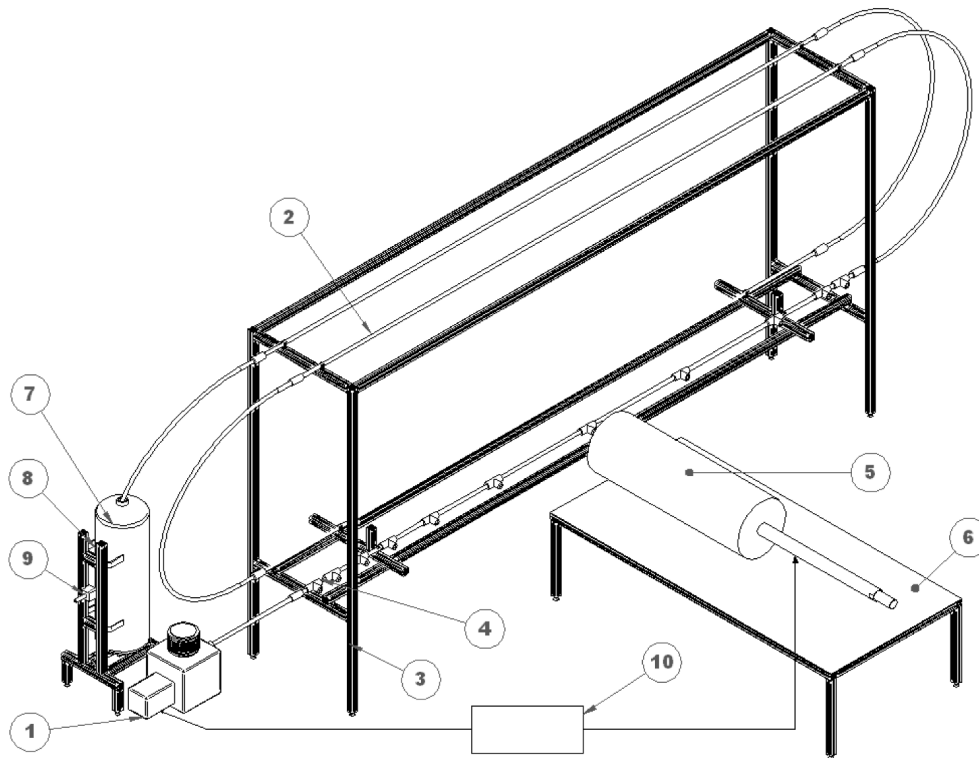


Figura 5.21: *Schema costruttivo del dimostratore: 1. Compressore alternativo monostadio; 2. Condotto di mandata; 3. Telaio di sostegno del condotto; 4. Sensori; 5. Sistema di attenuazione; 6. Tavolo di sostegno del sistema di attenuazione; 7. Serbatoio; 8. Telaio di sostegno del serbatoio; 9. Valvola di regolazione della pressione; 10. Controllore.*

L'impianto di test il cui schema è riportato in figura 5.21 deve essere in grado di poter riprodurre il fenomeno delle pulsazioni di pressione e consentire al tempo stesso di inserire dei dispositivi per la loro attenuazione. Di seguito viene riportata una breve descrizione dei principali componenti del sistema:

1. Compressore alternativo monostadio.

Il compressore alternativo monostadio, utilizzato per la generazione delle pulsazioni di pressione, garantisce una pressione di esercizio all'interno del condotto pari a 4 *bar*.

Il rapporto di compressione tra l'albero del motore e l'albero del compressore è stato scelto pari a  $\frac{1}{2}$  in funzione del motore disponibile e dell'inverter selezionato (tabella 5.5). Il compressore è movimentato da un motore elettrico comandato da inverter: in questo modo è possibile variare la velocità di funzionamento e quindi la frequenza delle pulsazioni di pressione generate.

## 2. Condotto di mandata.

Il condotto di mandata, realizzato in più parti, consente di riconfigurare la posizione degli elementi del condotto al fine di disporre di un'ampia gamma di configurazioni possibili per il differente posizionamento dei sensori e del sistema di attenuazione delle pulsazioni. Le differenti parti che costituiscono il condotto sono collegate tra loro da raccordi disponibili commercialmente. I tubi di differenti lunghezze che costituiscono il condotto di mandata consentono, anche se sono in numero limitato, di disporre di un ampio numero di configurazioni possibili. Il diametro interno dei tubi è pari a  $d_{cond} = 12.7 \text{ mm}$ , scelto da quelli disponibili a stock dai vari produttori. La lunghezza totale del condotto è pari a  $l_{tot} = 20 \text{ m}$  e consente di ottenere un ambiente sonoro che simuli un impianto molto esteso, ovvero che non risenta degli effetti di bordo tipici dei condotti corti rispetto alle frequenze a cui vengono eccitati. Inoltre si sceglie di far terminare il condotto in un serbatoio di dimensioni sufficientemente grandi, in modo tale che le onde di pressione che dovessero raggiungere la fine del condotto non vengano riflesse all'interno dello stesso. In questo modo è anche possibile fissare la pressione statica dell'intero impianto mediante una valvola di regolazione.

## 3. Telaio di sostegno del condotto.

Il telaio di sostegno del condotto è realizzato mediante profilati di alluminio ed è progettato in modo tale da garantire una regolazione in altezza delle staffe di sostegno del condotto.

## 4. Sensori.

Per la misurazione del comportamento del flusso all'interno del condotto si utilizzano due tipologie di sensori: sensori di pressione e sensori di temperatura. I

sensori di pressione sono scelti in modo tale da poter rilevare la componente la componente dinamica della pressione, dovuta alle pulsazioni generate dal compressore. I sensori vengono posizionati a monte e a valle del risonatore, con lo scopo di verificare l'effetto del risonatore sulle pulsazioni di pressione. In particolare si utilizzeranno quattro sensori per la misura della pressione dinamica e due termocoppie. Per la misura della velocità di rotazione del motore del compressore verrà utilizzato un opportuno encoder incrementale. I sensori per la rilevazione della pressione dinamica sono i sensori modello 2200C1 e 2200C4 prodotti da Dytran e utilizzati nella campagna di misura sperimentale sul compressore industriale. Analogamente, l'encoder incrementale è il modello IH103 prodotto da GES e già utilizzato nella precedente campagna di misura.

#### 5. Sistema di attenuazione.

Il sistema di attenuazione delle pulsazioni di pressione che si intende studiare è costituito da un risonatore di Helmholtz. Le caratteristiche fisiche del risonatore di Helmholtz sono strettamente legate alla sua frequenza di risonanza. La modifica di queste caratteristiche può essere effettuata mediante un'opportuna attuazione del dispositivo in grado di regolare alcune caratteristiche geometriche del risonatore. La progettazione, la realizzazione e l'analisi di questi dispositivi verrà effettuata nel capitolo 6.

#### 6. Tavolo di sostegno del sistema di attenuazione.

Il risonatore di Helmholtz e gli organi necessari alla sua movimentazione vengono fissati su un tavolo realizzato con profilati di alluminio e progettato in modo da garantire il posizionamento dell'intero sistema di attenuazione in differenti punti del condotto.

#### 7. Serbatoio.

Il volume dell'ambiente sonoro del condotto di mandata è pari a:

$$V_{cond} = \frac{d_{cond}^2}{4} \pi l_{tot} = 2.53 \cdot 10^6 \text{ mm}^3. \quad (5.3)$$

Per attenuare le onde di pressione che giungono al termine del condotto, occorre inserire un serbatoio avente un volume sufficientemente grande in modo da assorbire le oscillazioni di pressione. Si è scelto un serbatoio con volume pari a 18 l,

più di sette volte maggiore rispetto al volume del condotto, disponibile commercialmente. Esso viene posizionato in corrispondenza del termine del condotto di mandata e connesso allo stesso mediante un condotto opportunamente curvato. Sul serbatoio sono presenti dei punti di connessione per i seguenti dispositivi:

- Ingresso del condotto.
- Valvola di sicurezza.
- Valvola di regolazione della pressione.
- Scarico della condensa.
- Scarico aria.

In figura 5.22 si riporta uno schema costruttivo del serbatoio.

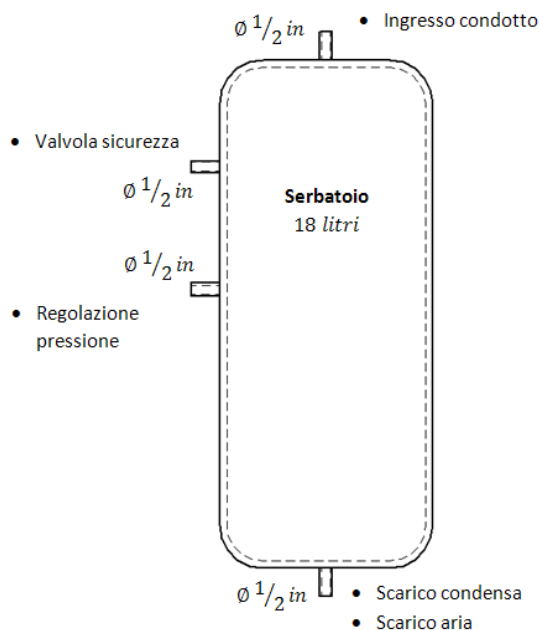


Figura 5.22: *Schema costruttivo del serbatoio.*

#### 8. Telaio di sostegno del serbatoio.

Tale struttura garantisce il posizionamento del serbatoio nella posizione più adeguata agli spazi disponibili per l'impianto di test. Esso è collocato sullo stesso *skid* che porta il compressore alternativo monostadio e il relativo impianto elettrico, per rendere più compatto l'impianto.

#### 9. Scarico dell'aria.



A valle della valvola per la regolazione della pressione viene predisposto un opportuno condotto di scarico dell'aria, che all'occorrenza può essere dotato anche di silenziatore.

#### 10. Controllore.

Il controllore mostrato in figura 5.21 permette di monitorare la velocità di rotazione del compressore in modo da determinare la frequenza delle onde di pressione all'interno del condotto. Inoltre, mediante un opportuno algoritmo di controllo che considera le misure rilevate dai sensori, il controllore permette di comandare l'attuatore del risonatore, adattando la geometria dello stesso alle reali condizioni del flusso presente nel condotto.

Per meglio delineare tutti i componenti necessari alla realizzazione dell'impianto di test, si riporta un elenco dei principali componenti del sistema.

Rif.	Descrizione componente	Specifiche	Q.tà	Note
1	Compressore alternativo monostadio		1	Fornitore: SIAD
2	Condotto di mandata	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $l = 70 \text{ mm}$ , Hy-Lok 16.	1	Diametro interno calibrato. Fornitore: Hy-Lok
3	Condotto di mandata	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $l = 120 \text{ mm}$ , Hy-Lok 16.	1	Diametro interno calibrato. Fornitore: Hy-Lok
4	Condotto di mandata	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $l = 195 \text{ mm}$ , Hy-Lok 16.	1	Diametro interno calibrato. Fornitore: Hy-Lok
5	Condotto di mandata	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $l = 220 \text{ mm}$ , Hy-Lok 16.	1	Diametro interno calibrato. Fornitore: Hy-Lok

6	Condotto di mandata	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $l = 310 \text{ mm}$ , Hy-Lok 16.	1	Diametro interno calibrato. Fornitore: Hy-Lok
7	Condotto di mandata	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $l = 401 \text{ mm}$ , Hy-Lok 16.	1	Diametro interno calibrato. Fornitore: Hy-Lok
8	Condotto di mandata	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $l = 470 \text{ mm}$ , Hy-Lok 16.	3	Diametro interno calibrato. Fornitore: Hy-Lok
9	Condotto di raccordo	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $r_{curva} = 500 \text{ mm}$ , Hy-Lok 16.	2	Diametro interno calibrato. Fornitore: Hy-Lok
10	Condotto di raccordo	$\phi_{est} = 16 \text{ mm}$ , $\phi_{int} = 12.7 \text{ mm}$ , $r_{curva} = 510 \text{ mm}$ , Hy-Lok 16.	1	Diametro interno calibrato. Fornitore: Hy-Lok
11	Serbatoio	18 l	1	Fornitore: SIAD
12	Raccordo X connessione risonatore e sensore pressione	CXA-16M	8	Fornitore: Hy-Lok
13	Raccordo T connessione sensore di pressione	CTA-16M	2	Fornitore: Hy-Lok
14	Raccordo lineare connessione condotti	CUA-16M	6	Fornitore: Hy-Lok
15	Regolatore di pressione		1	Fornitore: SIAD
16	Silenziatore		1	Fornitore: SIAD
17	Valvola di sicurezza		1	Fornitore: SIAD
18	Regolatore di pressione		1	Fornitore: SIAD

19	Scarico condensa		1	Fornitore: SIAD
20	Scarico aria		1	Fornitore: SIAD
21	Tappi di chiusura per raccordi		1	Fornitore: SIAD
22	Struttura sostegno risonatore		1	Fornitore: ITEM
23	Struttura sostegno condotto		1	Fornitore: ITEM
24	Inverter comando compressore	ACS350-03E-12A5-4 Telaio: NEMA 1	1	Fornitore: ABB

Tabella 5.5: *Elenco dei componenti principali dell'impianto.*



Figura 5.23: *Compressore alternativo monostadio utilizzato nell'impianto di test.*



Figura 5.24: *Condotto di mandata dell'impianto di test.*

Nelle figure 5.23 e 5.24 si mostrano rispettivamente l'apparato del compressore alternativo monostadio per la generazione delle pulsazioni di pressione e il condotto di test, realizzato secondo le specifiche di progetto.

Per facilitare l'utilizzo del compressore alternativo monostadio e per verificare il comportamento dei dispositivi oggetto di studio nelle differenti condizioni di funzionamento, il motore del compressore viene alimentato da un opportuno azionamento che

consente di variare il regime di rotazione e quindi le condizioni di funzionamento della macchina. Tale azionamento, riportato nella tabella 5.5 al punto 24, non era presente originariamente a bordo della macchina: è stato scelto in funzione delle caratteristiche del motore presente sull'impianto e integrato nell'impianto elettrico della macchina.

### 5.2.2 Caratterizzazione sperimentale dell'impianto

Come effettuato per il compressore industriale, anche nell'impianto di test è necessario effettuare una campagna di misura per la caratterizzazione sperimentale del comportamento dinamico della macchina, relativo alle pulsazioni di pressione immesse nel condotto. L'approccio alla misura è il medesimo presentato per la misura sul compressore industriale nel paragrafo 5.1.2. Si utilizzano gli stessi sensori di pressione definiti nella tabella 5.1, e lo stesso tipo di encoder, montato direttamente sull'albero del compressore. La risoluzione dei sensori è tale da poter apprezzare anche le pulsazioni di pressione generate da un impianto di piccole dimensioni, come quello di test. Le misure vengono effettuate nelle seguenti condizioni:

- Condizione nominale: misura a differenti regimi di rotazione con pressione media pari a 4 *bar*.
- Condizione parzializzata: misura a differenti regimi di rotazione con pressione media pari a 0 *bar*.

La condizione di funzionamento nominale prevede la chiusura di tutte le valvole nell'impianto ad eccezione di quella per la regolazione della pressione, che viene imposta per il mantenimento di una pressione media di 4 *bar*. La condizione parzializzata si ottiene mediante l'apertura della valvola di parzializzazione, che porta la pressione relativa media del condotto pari a 0 *bar*, pari alla pressione ambiente. Le prove vengono effettuate anche per variazioni della velocità di rotazione del motore, connesso all'albero del compressore per mezzo di un sistema a pulegge e volano, che presenta il seguente rapporto di trasmissione:

$$\tau = \frac{RPM_{albero}}{RPM_{motore}} = 0.50. \quad (5.4)$$

I regimi di rotazione del motore a cui sono state eseguite le prove sono elencati nella tabella 5.6. La velocità massima di rotazione è pari a 1600 *rpm*, che corrisponde ad una frequenza della pulsazione sull'albero del compressore pari a:

$$f_{1X} = \frac{vel_{rot}\tau}{60} = 13.3 \text{ Hz.} \quad (5.5)$$

La frequenza di acquisizione del segnale, considerando di voler analizzare anche le componenti multiple della fondamentale, è stata scelta anche in questo caso pari a:

$$f_c = 4096 \text{ Hz,} \quad (5.6)$$

di conseguenza la frequenza di taglio dei filtri analogici è stata impostata a  $f_{cut} = 1000 \text{ Hz}$ , ben al di sotto della frequenza di Nyquist  $f_{Ny} = f_c/2$ , poiché i filtri analogici utilizzati sono filtri del primo ordine e si vuole assicurare un adeguato abbattimento del segnale in corrispondenza della frequenza di Nyquist, per evitare fenomeni di aliasing.

RPM motore desiderati							
450	500	550	600	650	700	750	800
850	900	950	1000	1050	1100	1150	1200
1250	1300	1350	1400	1450	1500	1550	1600

Tabella 5.6: *Elenco di velocità di rotazione del motore del compressore in cui vengono eseguite le misure.*

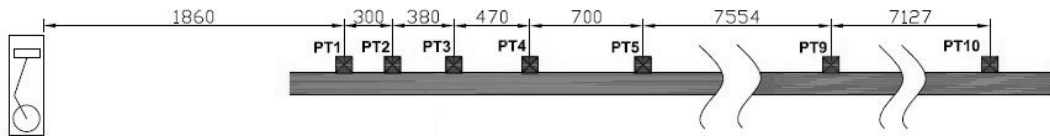


Figura 5.25: *Punti di misura sull'impianto di test.*

I punti di misura presenti sull'impianto sono sette, pertanto non vi è la necessità di suddividere la misura in gruppi di misura, dato che con gli otto sensori a disposizione si riescono a coprire i punti di misura di interesse. Tali punti di misura verranno utilizzati quali punti di collegamento per i sistemi di abbattimento passivo che si intendono studiare. Nella figura 5.25 si riportano i punti di misura, indicati con PT1 ÷ PT10, e le distanze tra i vari punti di misura. Le misure effettuate sono state elaborate in post-processing per eseguire le seguenti analisi:

- Andamento temporale della pressione.
- Analisi spettrale della pressione.

- Verifica della stazionarietà dell'andamento spettrale della pressione.
- Misura della velocità di rotazione istantanea dell'albero del compressore.

Si considera, a titolo di esempio la misura condotta nelle seguenti condizioni:

- Regime di rotazione: 1600 rpm.
- Punti di misura: PT1 e PT5.
- Condizioni di funzionamento: condizione nominale.
- Durata del record di acquisizione: 10 s.

L'andamento temporale della pressione nei punti PT1 e PT5 è mostrato nelle figure 5.26 e 5.27. Si nota come anche su questo impianto l'andamento temporale della pressione dinamica presenti delle periodicità, che possono essere studiate mediante un'analisi in frequenza del segnale stesso.

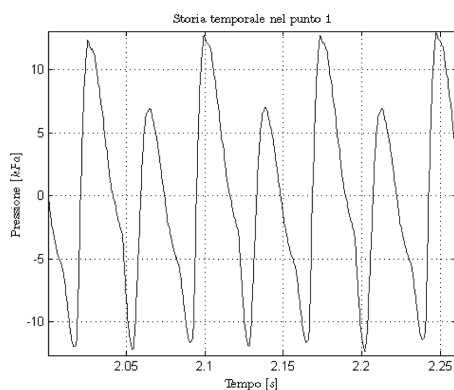


Figura 5.26: Campagna di misura su impianto di test: andamento temporale delle pressioni nel punto PT1.

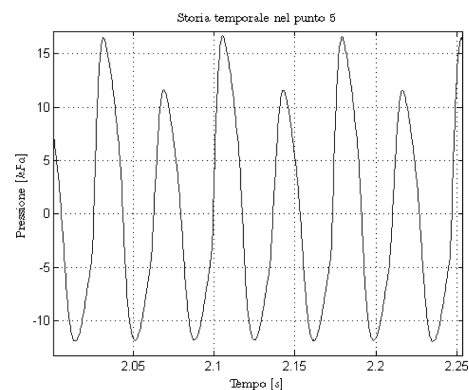


Figura 5.27: Campagna di misura su impianto di test: andamento temporale delle pressioni nel punto PT5.

Poiché dalle analisi nel dominio del tempo non è possibile effettuare delle valutazioni esaustive, legate alla composizione armonica delle pulsazioni di pressione, si è eseguita l'analisi spettrale su tutti i segnali rilevati. Nelle figure 5.28 e 5.28 si riporta rispettivamente l'analisi eseguite per il punto PT1 e per il punto PT5 nelle condizioni descritte.

Considerando una velocità di rotazione pari a  $RPM_{motore} = 1600 \frac{giri}{min}$ , la frequenza fondamentale della forzante risulta:

$$f_{1x} = \frac{RPM_{motore}\tau}{60 s} = 13.3 Hz. \quad (5.7)$$

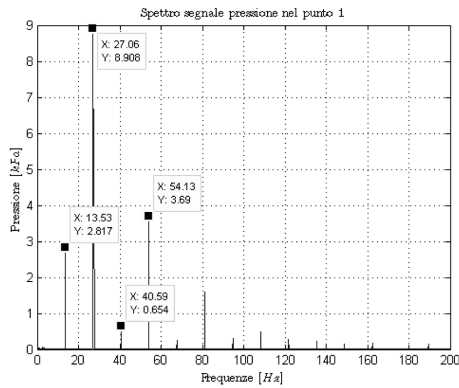


Figura 5.28: Campagna di misura su impianto di test: analisi spettrale su un'acquisizione nel punto PT1.

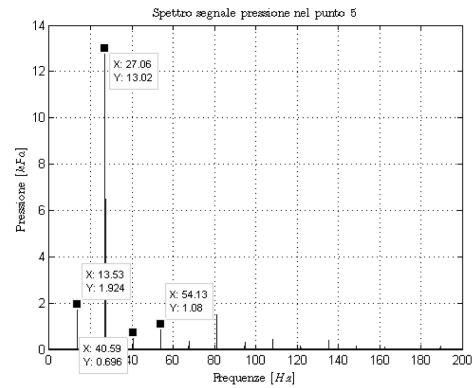


Figura 5.29: Campagna di misura su impianto di test: analisi spettrale su un'acquisizione nel punto PT5.

Indice	Frequenza [Hz]
1 x giro	13.3
2 x giro	26.6
3 x giro	40
4 x giro	53.3
5 x giro	66.6
6 x giro	80

Tabella 5.7: Analisi dell'impianto di test: frequenze multiple della frequenza di rotazione dell'albero del compressore.

Le frequenze teoriche multiple della fondamentale sono riportate nella tabella 5.2.2. Si nota come queste frequenze siano effettivamente riscontrabili dall'analisi spettrale dei segnali acquisiti. È possibile adimensionalizzare l'asse delle ascisse rispetto alla frequenza di rotazione dell'albero del compressore, o frequenza fondamentale, ottenendo l'andamento mostrato nelle figure 5.30 e 5.31.

Infine si è proceduto alla verifica della stazionarietà nel tempo delle componenti armoniche rilevate dall'analisi spettrale. Per questa verifica occorre eseguire uno spettrogramma sul record acquisito. Il risultato di questa analisi è riportato nelle figure

5.32 e 5.33 Lo spettrogramma delle frequenze mostra anche per questo impianto un andamento costante delle ampiezze spettrali, pertanto si può concludere che l'andamento delle pulsazioni di pressione sia stazionario. Risultati medesimi a quelli mostrati sono stati ottenuti per le analisi svolte ai diversi regimi di rotazione dell'albero del compressore.

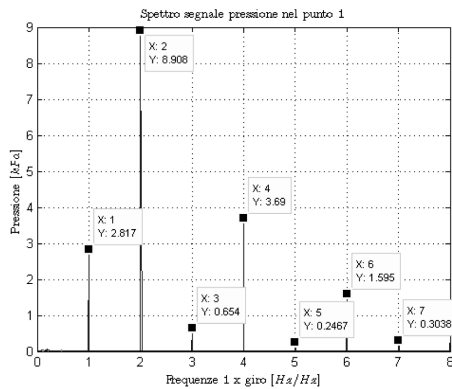


Figura 5.30: Campagna di misura su impianto di test: analisi spettrale e rappresentazione delle frequenze in multipli della frequenza di rotazione dell'albero del compressore (punto PT1).

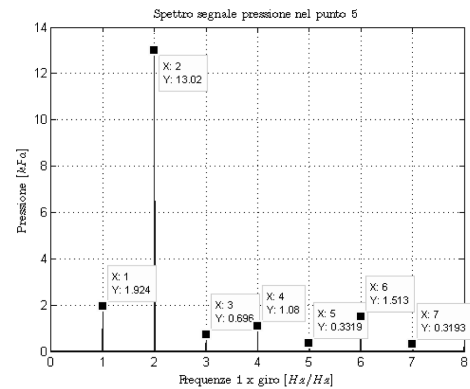


Figura 5.31: Campagna di misura su impianto di test: analisi spettrale e rappresentazione delle frequenze in multipli della frequenza di rotazione dell'albero del compressore (punto PT5).

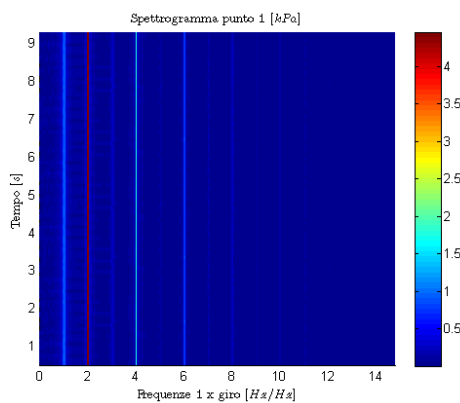


Figura 5.32: Campagna di misura sperimentale su impianto di test: spettrogramma del punto PT1.

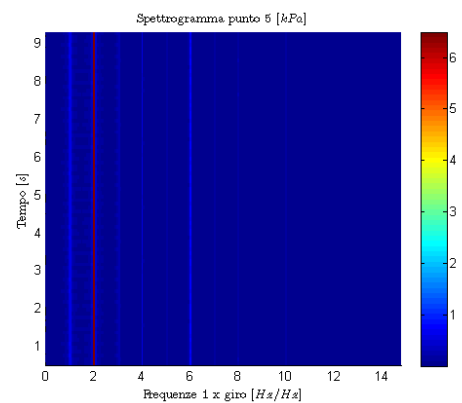


Figura 5.33: Campagna di misura sperimentale su impianto di test: spettrogramma del punto PT5.



In figura 5.34 è mostrato l'andamento della velocità istantanea di rotazione: si nota come essa non sia costante e contenga delle componenti in frequenza, ma tale variazione di velocità è comunque minore del 2 % come nel caso del compressore industriale, pertanto risulta poco influente sul comportamento dinamico della macchina.

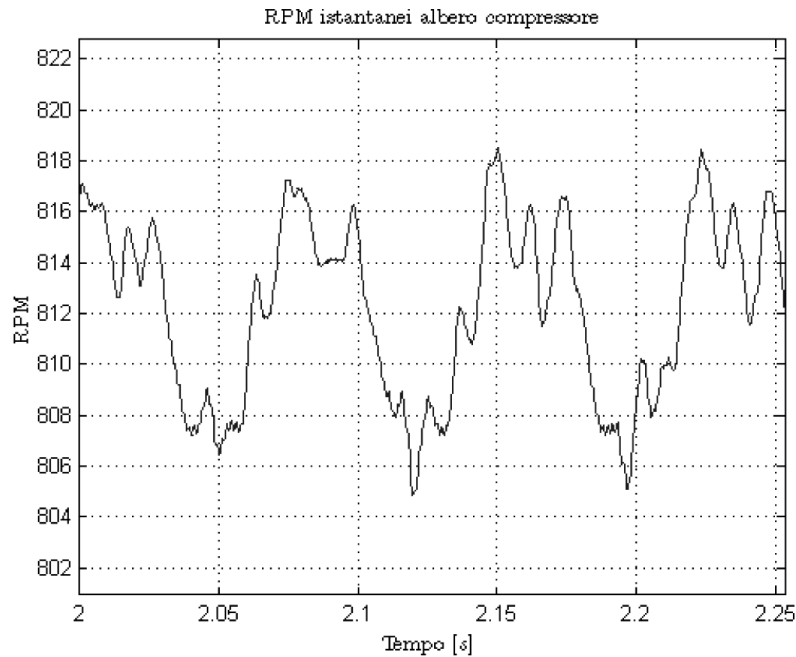


Figura 5.34: *Campagna di misura sperimentale su impianto di test: velocità istantanea di rotazione dell'albero del compressore.*

### Conclusioni dell'indagine sperimentale

È stata effettuata una vasta attività di misura che ha reso possibile la mappatura del comportamento dinamico della macchina in analisi rispetto ai diversi regimi e condizioni di funzionamento. I dati acquisiti sono stati elaborati in post-processing effettuando analisi sia nel dominio del tempo che della frequenza, ricercando corrispondenze tra i valori teorici e sperimentali, ed evidenziando le stazionarietà dei fenomeni dinamici. Le osservazioni finali possono essere riassunte nel seguente elenco:

- Le frequenze delle pulsazioni di pressione sono multiple intere della velocità di rotazione dell'albero del compressore.
- Vi è corrispondenza tra valori teorici e sperimentali.

- I fenomeni risultano stazionari: nello stesso punto di misura le pulsazioni sono sempre a regime (frequenza e ampiezza fissa).
- Le frequenze delle pulsazioni  $2X_{\text{giro}}$  e  $4X_{\text{giro}}$  mostrano le maggiori ampiezze.

## Capitolo 6

# Analisi sperimentale dei sistemi ad attuazione indiretta

### 6.1 Progettazione e realizzazione dei risonatori fissi

Il progetto dei dispositivi di attenuazione delle pulsazioni ad attuazione indiretta esposti nel capitolo 4 è stato effettuato considerando le caratteristiche dinamiche del flusso di aria dell'impianto di test presentato nel capitolo 5. Nella tabella 6.1 vengono riassunte le caratteristiche principali dell'impianto di test di interesse per la progettazione dei sistemi risonanti.

Parametro	Descrizione
Fluido di lavoro	Aria
Temperatura fluido di lavoro	$T = 20^\circ C$
Velocità del suono nel mezzo	$c = 343.20 \text{ m/s}$
Velocità di rotazione del motore	$450 \leq RPM_{motore} \leq 1600 \text{ RPM}$
Frequenza 2x giro delle oscillazioni di pressione	$7.5 \leq f_R \leq 26.7 \text{ Hz}$
Pressione statica nel condotto	$p_0 = 4 \text{ bar}$
Diametro interno del condotto principale	$d_{cnd} = 12.4 \text{ mm}$

Tabella 6.1: *Caratteristiche dell'impianto di test di interesse per la progettazione dei dispositivi risonanti.*

Essendo l'impianto alimentato da un compressore alternativo a doppio effetto, le oscillazioni di pressione con ampiezza maggiore si hanno in corrispondenza della componente doppia della velocità di rotazione dell'albero del compressore. Considerando

infatti il range di variazione della velocità del motore, riportato nella tabella 6.1, e il rapporto di riduzione del sistema di trasmissione utilizzato, il range di frequenze delle oscillazioni di pressione a maggiore ampiezza  $f_R$  si determinano mediante la seguente relazione:

$$f_R = \frac{RPM_{motore}\tau}{60} \kappa_{2x}, \quad (6.1)$$

dove  $\kappa_{2x} = 2$  è la componente doppia del numero di giri dell'albero del compressore. I valori di  $f_R$  sono indicati nella tabella 6.1. Dato l'elevato scambio termico tra l'aria presente nel condotto e l'ambiente circostante, si approssima il valore di temperatura dell'aria, nella zona prossima a quella di collegamento del risonatore, pari a quella ambiente, in questo modo la velocità di propagazione del suono è pari a:

$$c = \sqrt{\gamma RT} = 343.2 \text{ m/s}. \quad (6.2)$$

Note le caratteristiche dinamiche del flusso d'aria all'interno del condotto principale dell'impianto di test si vuole indagare l'effetto di abbattimento legato all'inserimento nell'impianto di alcuni risonatori con caratteristiche geometriche fisse, pertanto in grado di abbattere un'oscillazione di pressione ad una certa componente in frequenza. La fase di progetto e di test dei risonatori vedrà la realizzazione e l'analisi del comportamento dinamico e delle performance di differenti risonatori al fine di confrontare gli abbattimenti ottenibili dalle diverse tipologie di risonatori presentati nel capitolo 4, in particolare verranno analizzati:

- risonatori side-branch rettilinei;
- risonatori side-branch a spirale;
- risonatori di Helmholtz.

Per semplicità di notazione verranno utilizzate di seguito le seguenti diciture:

- SS-x-PTy
  - SS: solo impianto in assenza di dispositivi risonanti;
  - x: velocità di rotazione dell'albero del compressore;
  - PTy: punto di misura in cui vengono rilevate le oscillazioni di pressione.

- $R_f^x$ -PTz-w-PTk
  - $R$ : impianto con risonatore collegato al condotto principale;
  - $x$ : tipo di risonatore collegato al condotto principale, in particolare  $SBR$  indica un risonatore side-branch rettilineo,  $SBS$  un risonatore side-branch a spirale, mentre  $H$  un risonatore di Helmholtz;
  - $f$ : frequenza di risonanza  $f_R$  del risonatore;
  - PTz: punto di collegamento del risonatore lungo il condotto principale;
  - $w$ : velocità di rotazione dell'albero del compressore;
  - PTk: punto di misura in cui vengono rilevate le oscillazioni di pressione.

### 6.1.1 Progettazione dei risonatori side-branch rettilinei

La prima tipologia di risonatore che si intende progettare e analizzare sull'impianto di test è il risonatore side-branch rettilineo ( $SBR$ ). Dato l'elevato ingombro di questo tipo di dispositivi al decrescere della loro frequenza di risonanza, ossia della frequenza delle oscillazioni di pressione che si intende abbattere, la lunghezza massima in grado di essere testata sperimentalmente in laboratorio è pari a  $3.5\text{ m}$ , corrispondente alla frequenza di  $24.7\text{ Hz}$ . Considerando il range di frequenze delle oscillazioni di pressione ad ampiezza massima presenti nel condotto principale (tab. 6.1) il range di lunghezze dei risonatori side-branch, determinato mediante la relazione 4.109, è pari a:

$$3.2 \leq L_R \leq 11.4\text{ m}. \quad (6.3)$$

Volendo abbattere una componente di oscillazione di pressione alla frequenza di  $26\text{ Hz}$  viene progettato un risonatore side-branch di lunghezza pari a  $L_R = 3.3\text{ m}$ . Tale risonatore, secondo la nomenclatura esposta nel paragrafo 6.1, sarà denominato in seguito come  $R_{26.0}^{SBR}$ . La sezione del risonatore, secondo quanto esposto nel paragrafo 4.3.2, viene scelta pari a quella del condotto, pertanto il risonatore side-branch viene realizzato utilizzando un tubo delle stesse caratteristiche del condotto principale (fig. 6.1). Per valutare l'abbattimento del risonatore  $R_{26.0}^{SBR}$  è necessario disporre nel condotto di un'oscillazione di pressione alla frequenza di  $26\text{ Hz}$  quindi, secondo la relazione 6.1, il numero di giri del motore deve essere imposto, grazie all'inverter di cui è dotato l'impianto, pari a:

$$RPM_{motore} = \frac{60f_R}{\kappa_{2x}} = 775\text{ RPM}. \quad (6.4)$$

A tale velocità del motore corrisponde una velocità di rotazione dell'albero del compressore pari a 1550 *RPM*.

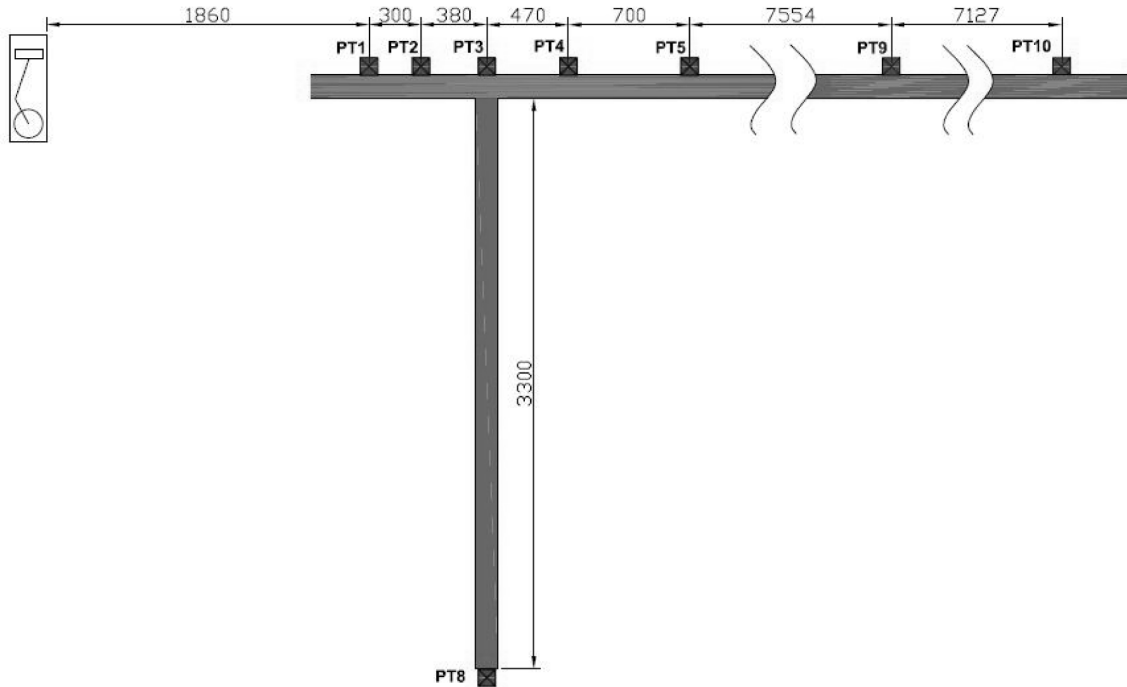


Figura 6.1: Risonatore side-branch rettilineo  $R_{26.0}^{SBR}$ : schema dell'impianto.

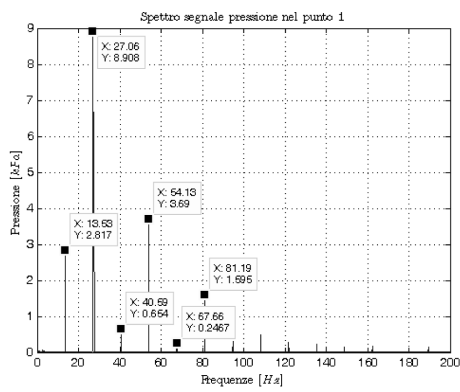


Figura 6.2: Andamento spettrale oscillazioni di pressione: SS-1550-PT1.

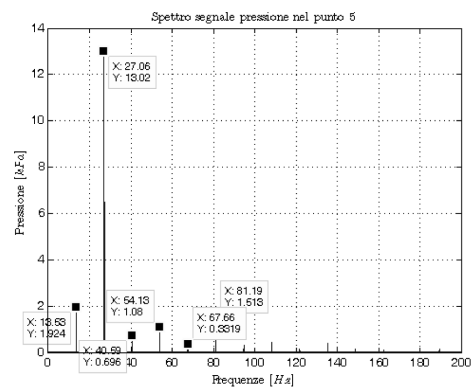


Figura 6.3: Andamento spettrale oscillazioni di pressione: SS-1550-PT5.

Data la difficoltà nell'utilizzo pratico di questa tipologia di risonatori, si sceglie di utilizzare il risonatore appena progettato per una sola valutazione dell'influenza della posizione di inserimento del dispositivo lungo il condotto principale dell'impianto; a tal fine il risonatore  $R_{26.0}^{SBR}$  viene collegato al condotto principale in corrispondenza dei punti PT1, PT2, PT3 e PT4 e vengono valutate le oscillazioni di pressione nei punti di misura PT1 e PT5. Nelle figure 6.2 e 6.3 viene mostrato l'andamento nel dominio delle frequenze delle oscillazioni di pressione nel condotto principale in assenza di dispositivi risonanti per l'abbattimento.

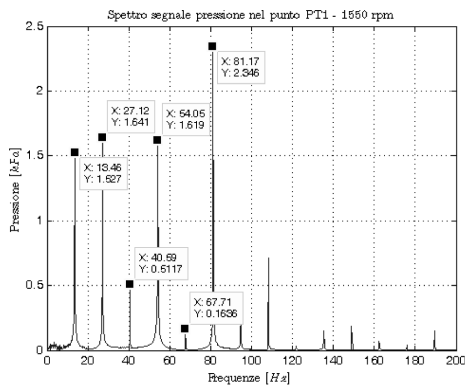


Figura 6.4: *Andamento spettrale oscillazioni di pressione:  $R_{26.0}^{SBR}$ -PT1-1550-PT1.*

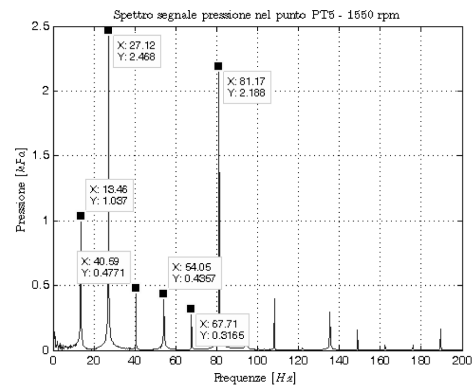


Figura 6.5: *Andamento spettrale oscillazioni di pressione:  $R_{26.0}^{SBR}$ -PT1-1550-PT5.*

Dalla figura 6.4 alla figura 6.11 viene mostrato invece l'andamento delle oscillazioni di pressione nel condotto principale a seguito dell'inserimento nell'impianto del risonatore  $R_{26.0}^{SBR}$  nei punti sopracitati.

Nelle figure 6.12 e 6.13 si riporta l'andamento complessivo dell'attenuazione delle oscillazioni di pressione legato allo spostamento del risonatore lungo il condotto principale: queste figure sono state ottenute rapportando le ampiezze delle oscillazioni di pressione in presenza del risonatore  $R_{26.0}^{SBR}$  con le ampiezze in assenza di dispositivi, in corrispondenza delle frequenze multiple di quelle di rotazione del compressore. Da queste figure è evidente il buon comportamento del risonatore side-branch  $R_{26.0}^{SBR}$  alla frequenza di progetto  $f_R = 26.0 \text{ Hz}$  corrispondente alla componente in frequenza doppia della velocità di rotazione del compressore (componente 2x giro): l'abbattimento è cospicuo e pressoché equivalente al variare della posizione di inserimento del risonatore.

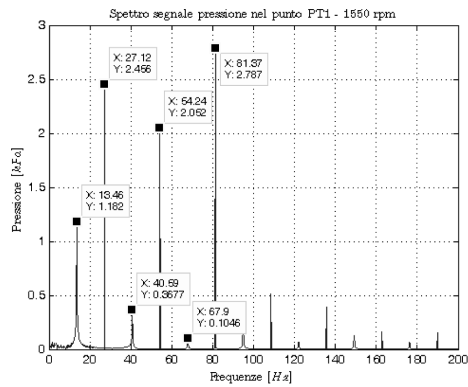


Figura 6.6: Andamento spettrale oscillazioni di pressione:  $R_{26,0}^{SBR-PT2-1550-PT1}$ .

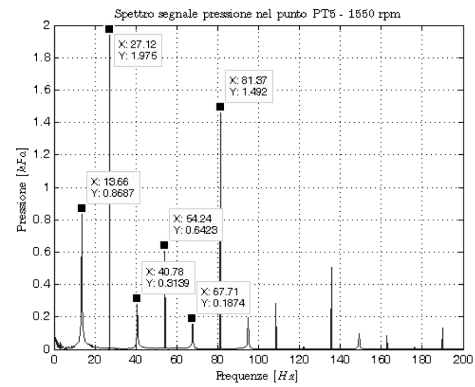


Figura 6.7: Andamento spettrale oscillazioni di pressione:  $R_{26,0}^{SBR-PT2-1550-PT5}$ .

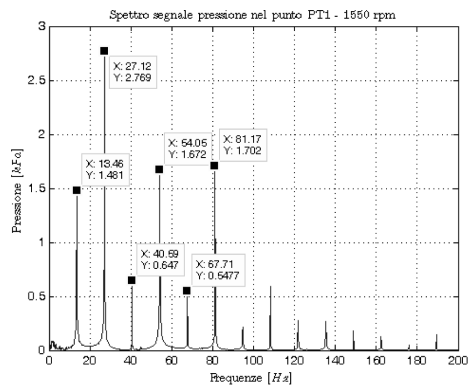


Figura 6.8: Andamento spettrale oscillazioni di pressione:  $R_{26,0}^{SBR-PT3-1550-PT1}$ .

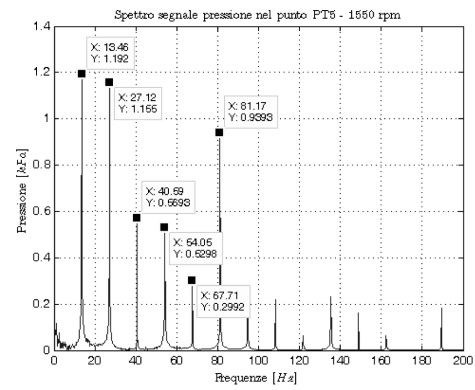


Figura 6.9: Andamento spettrale oscillazioni di pressione:  $R_{26,0}^{SBR-PT3-1550-PT5}$ .

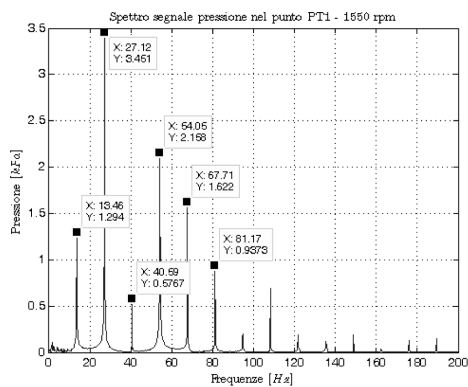


Figura 6.10: Andamento spettrale oscillazioni di pressione:  $R_{26,0}^{SBR-PT4-1550-PT1}$ .

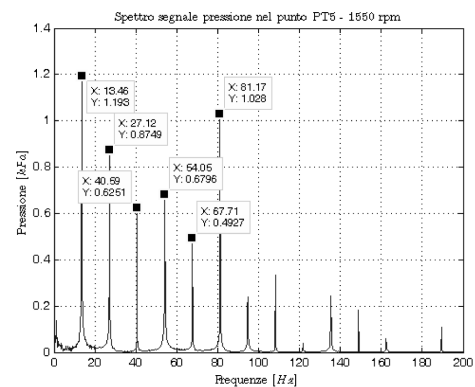


Figura 6.11: Andamento spettrale oscillazioni di pressione:  $R_{26,0}^{SBR-PT4-1550-PT5}$ .



Un'altra importante osservazione riguarda l'influenza del risonatore sulla dinamica complessiva dell'impianto: infatti l'inserimento del risonatore comporta una variazione delle ampiezze delle oscillazioni di pressione anche a monte del punto di collegamento, mostrando anche in queste posizioni un buon indice di abbattimento. Questo risultato si discosta da quello ottenuto mediante il modello analitico (par. 4.4) dove, a monte del punto di collegamento del risonatore, vi è invece un incremento del livello delle oscillazioni di pressione, in corrispondenza della frequenza  $f_R$ , rispetto al caso in cui non è presente il risonatore. In riferimento a questa discrepanza tra i dati sperimentali e analitici va considerato che, il modello acustico presentato in precedenza si limita a rappresentare il solo condotto in cui sono presenti le oscillazioni di pressione, prescindendo dal fatto che queste oscillazioni vengono generate da un sistema che presenta una propria dinamica. Le condizioni al contorno del modello utilizzato sono imposte agli estremi del condotto e non sull'effettiva sorgente delle oscillazioni, poiché questa non è modellata. Questa approssimazione porta ad una valutazione parziale ma sufficiente degli effetti che si verificano all'interno del condotto in presenza del dispositivo.

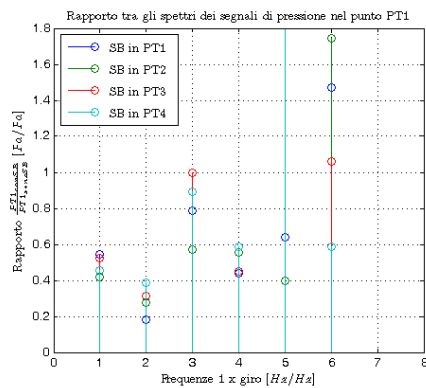


Figura 6.12: *Rapporto tra gli spettri dei segnali di pressione in corrispondenza di PT1, con e senza risonatore side-branch  $R_{26.0}^{SBR}$ .*

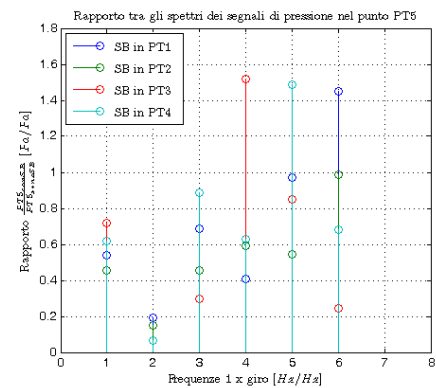


Figura 6.13: *Rapporto tra gli spettri dei segnali di pressione in corrispondenza di PT5, con e senza risonatore side-branch  $R_{26.0}^{SBR}$ .*

### 6.1.2 Progettazione dei risonatori side-branch a spirale

Dal progetto e dall'analisi sperimentale dei risonatori side-branch rettilinei è sorta la necessità di realizzare dei risonatori di forma più compatta al fine di poter valutare sperimentalmente l'efficienza di abbattimento in funzione della loro lunghezza e poter

raggiungere frequenze di abbattimento più basse. Per far fronte alla problematica relativa agli ingombri dei risonatori side-branch si sceglie di progettare e verificare il funzionamento di risonatori avvolti a spirale. In questo caso, tale progetto risulta di facile realizzazione grazie alle caratteristiche dell'impianto disponibile, dove il diametro del condotto principale è pari a  $d_{cnd} = 12.4 \text{ mm}$ , quindi avvolgibile con ridotti raggi di curvatura senza grosse difficoltà costruttive. Più problematico potrebbe essere il caso in cui il diametro del condotto principale dell'impianto è molto grande dove un risonatore side-branch risulta ingombrante e quindi difficilmente utilizzabile. Al fine di ridurre al minimo l'effetto geometrico di una eccessiva riduzione del raggio di curvatura dei risonatori side-branch a spirale si è limitato tale raggio a  $0.375 \text{ m}$ . Considerando il range di frequenze delle oscillazioni di pressione presenti nell'impianto e l'andamento della lunghezza del risonatore  $L_R$  in funzione della frequenza di risonanza  $f_R$ , riportato in figura 4.30, si è scelto di progettare e testare i risonatori riportati in tabella 6.2.

<b>Risonatore</b>	<b><math>L_R</math> [m]</b>	<b><math>f_R</math> [Hz]</b>
$R_{24.7}^{SBS}$	3.5	24.7
$R_{14.6}^{SBS}$	5.9	14.6
$R_{10.3}^{SBS}$	8.3	10.3
$R_{8.0}^{SBS}$	10.7	8.0
$R_{6.5}^{SBS}$	13.2	6.5
$R_{5.5}^{SBS}$	15.6	5.5

Tabella 6.2: *Risonatori side-branch a spirale realizzati.*

I risonatori side-branch a spirale vengono realizzati in modo modulare, incrementando la loro lunghezza mediante l'aggiunta di spire, come mostrato in figura 6.14. L'apice *SBS* relativo al nome del risonatore indica il tipo di risonatore (*Side Branch Spirale*), mentre il pedice indica la frequenza di risonanza del risonatore, ad esempio la dicitura  $R_{10.3}^{SBS}$  indica un risonatore side-branch a spirale con frequenza di risonanza  $f_R = 10.3 \text{ Hz}$ . I risonatori riportati in tabella 6.2 sono stati progettati in modo da ricoprire maggiormente il range di frequenze della componente 2x giro della velocità di rotazione dell'albero del compressore, ossia il range di frequenze  $7.5 \leq f_R \leq 26.7 \text{ Hz}$ .

Data la bassa influenza del punto di collegamento dei risonatori lungo il condotto principale dell'impianto sull'attenuazione delle oscillazioni di pressione, verificata mediante il risonatore side-branch rettilineo (par. 6.1.1), i risonatori a spirale vengono

testati in corrispondenza di un solo punto di collegamento. In particolare è stato scelto di collegare i risonatori side-branch a spirale in corrispondenza del punto  $PT3$ , ossia alla distanza di circa  $2.5\text{ m}$  dalla flangia di uscita del compressore. Sperimentalmente le oscillazioni di pressione sono state rilevate in tutti i punti di misura a disposizione, tuttavia i risultati ottenuti sui singoli punti mostrano mediamente gli stessi rapporti delle ampiezze delle oscillazioni tra il sistema costituito dal solo impianto e il sistema con il dispositivo risonante ad esso collegato: di seguito si riportano le misure della pressione dinamica in corrispondenza dei soli punti di misura  $PT1$  e  $PT10$  al fine di mostrare l'effetto di abbattimento a monte e a valle del punto di collegamento del risonatore. Per la posizione e la distanza dei punti di misura si rimanda al paragrafo 5.2.2.

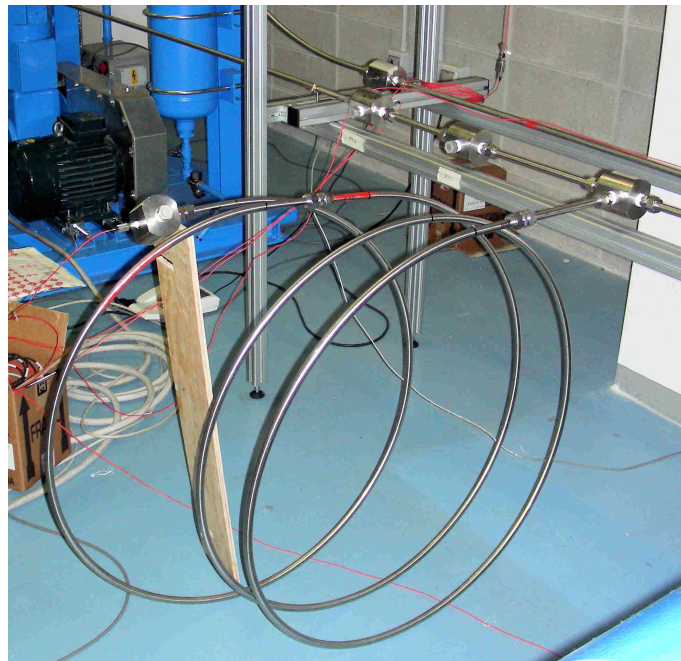


Figura 6.14: Risonatore side-branch a spirale  $R_{10.3}^{SBS}$ .

Per valutare l'abbattimento del risonatore  $R_{24.7}^{SBS}$  è necessario disporre nel condotto di un'oscillazione di pressione alla frequenza di circa  $25\text{ Hz}$  quindi, secondo la relazione 6.1 esposta nel paragrafo precedente, il numero di giri del motore deve essere imposto pari a  $750\text{ RPM}$ , a cui corrisponde una velocità di rotazione dell'albero del compressore di  $1500\text{ RPM}$ . In corrispondenza di questo regime di rotazione, e in assenza del dispositivo risonante, si ottiene l'andamento nel dominio delle frequenze riportato nelle figure 6.15 e 6.16 riferite ai punti di misura presi in considerazione per l'analisi. Al medesimo

regime di rotazione del motore è stato poi valutato l'effetto di abbattimento legato all'inserimento del risonatore  $R_{24,7}^{SBS}$  nel punto PT3; i risultati ottenuti, relativamente ai punti di misura considerati sono riportati nelle figure 6.17 e 6.18.

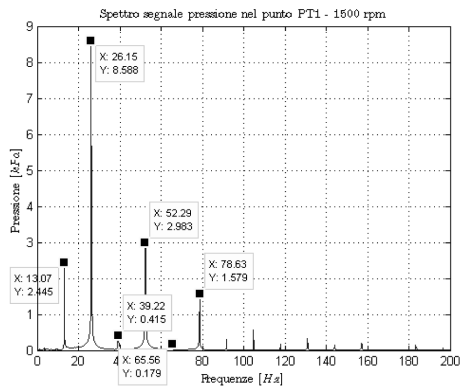


Figura 6.15: *Andamento spettrale oscillazioni di pressione: SS-1500-PT1.*

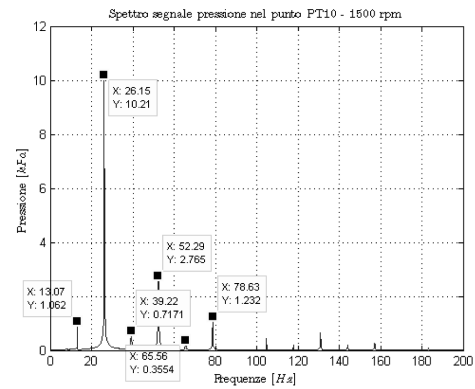


Figura 6.16: *Andamento spettrale oscillazioni di pressione: SS-1500-PT10.*

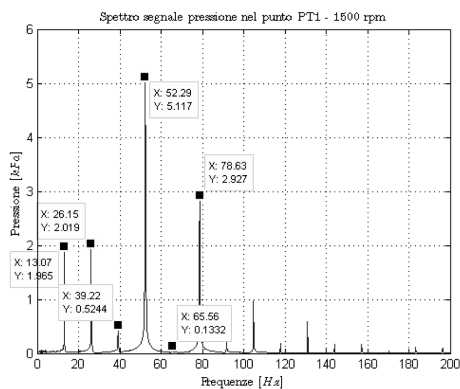


Figura 6.17: *Andamento spettrale oscillazioni di pressione:  $R_{24,7}^{SBS}$ -PT3-1500-PT1.*

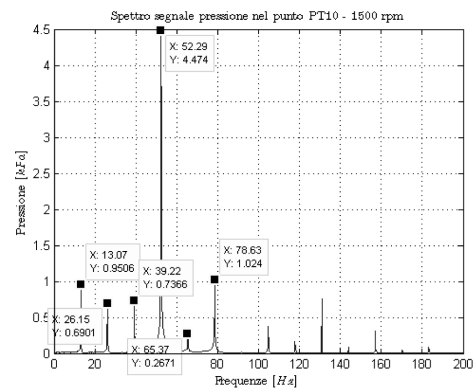


Figura 6.18: *Andamento spettrale oscillazioni di pressione:  $R_{24,7}^{SBS}$ -PT3-1500-PT10.*

Dalle figure 6.15 e 6.17 si nota come il risonatore side-branch a spirale sia in grado di apportare sostanziali abbattimenti anche a monte del punto di collegamento del risonatore infatti, considerando la componente spettrale alla frequenza di 26.15 Hz, questa passa dall'ampiezza di 8.6 kPa in assenza del risonatore, all'ampiezza ridotta di 2.0 kPa in presenza nell'impianto del risonatore  $R_{24,7}^{SBS}$ . Questo risultato, come già evidenziato in precedenza (par. 6.1.1), si discosta da quello ottenuto mediante il modello analitico (par. 4.4) dove, a monte del punto di collegamento del risonatore, vi è invece un incremento del livello delle oscillazioni di pressione, in corrispondenza della

frequenza  $f_R$ , rispetto al caso in cui non è presente il risonatore. Per la spiegazione di questa discrepanza tra i dati sperimentali e quelli ottenuti con i modelli matematici si rimanda al paragrafo 6.1.1. Le figure 6.16 e 6.18 mostrano invece l'effetto dell'inserimento del risonatore, rispetto al caso in cui questo non è presente nell'impianto, nel punto di misura PT10 a valle di quello di collegamento del risonatore e in particolare alla distanza di circa 15.8 m dal punto di collegamento del risonatore nel condotto principale. In corrispondenza di questo punto di misura si evidenzia l'elevata efficienza di abbattimento del risonatore  $R_{24.7}^{SBS}$  alla frequenza di 26.15 Hz, quindi nel range prossimo a  $f_R = 24.7$  Hz, mentre si nota un incremento dell'oscillazione di pressione alla frequenza di 52.29 Hz. Si ricorda infatti, secondo quanto mostrato nel modello analitico (par. 4.4.2), che il risonatore side-branch abbatte le oscillazioni di pressione in un range molto prossimo a  $f_R$ , mentre tende ad incrementare il livello delle oscillazioni in corrispondenza di alcune particolari frequenze. È interessante valutare, nell'intero range di funzionamento dell'impianto ( $450 \leq RPM_{motore} \leq 1600$  RPM), l'effetto dell'inserimento del risonatore  $R_{24.7}^{SBS}$ : per valutare questo andamento vengono visualizzati su un unico grafico gli spettri del segnale di pressione dinamica rilevati in un particolare punto di misura in assenza e in presenza del risonatore e al variare del numero di giri del motore del compressore. In particolare la figura 6.19 mostra gli spettri sovrapposti, ottenuti al variare del numero di giri del motore, in assenza di dispositivi risonanti nel sistema e in corrispondenza del punto di misura PT1, mentre la figura 6.21 mostra la sovrapposizione degli spettri ottenuti collegando in PT3 il risonatore  $R_{24.7}^{SBS}$  e rilevando le oscillazioni di pressione in PT1.

La stessa analisi viene effettuata anche nel punto di misura PT10, a valle del punto di collegamento del risonatore (figure 6.20 e 6.22). Grazie a queste analisi viene messa in evidenza la banda in cui il risonatore side-branch  $R_{24.7}^{SBS}$  apporta benefici relativamente al range di funzionamento del compressore; in particolare dalle figure 6.20 e 6.22 emerge un effetto benefico sull'abbattimento delle oscillazioni di pressione nel range di frequenze compreso tra 25 e 35 Hz. L'area di influenza del risonatore è mostrata nelle figure 6.21 e 6.22 con un'area tratteggiata posta sulle figure.

Dalla figura 6.23 alla figura 6.27 vengono mostrati i risultati ottenuti ripetendo le analisi viste finora per tutti i risonatori side-branch riportati in tabella 6.2: per sintetizzare la trattazione si riportano di seguito solamente i risultati complessivi per il solo punto di misura PT10, mediante la visualizzazione spettrale sovrapposta vista per il risonatore  $R_{24.7}^{SBS}$ .

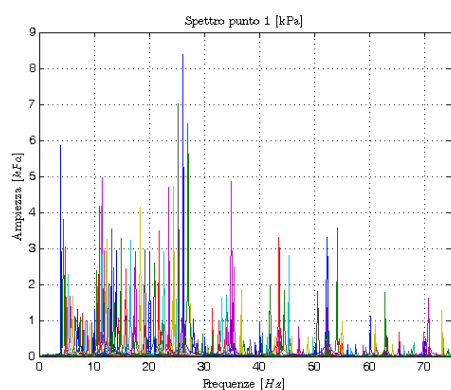


Figura 6.19: Sovrapposizione spettri oscillazioni di pressione: SS-PT1.

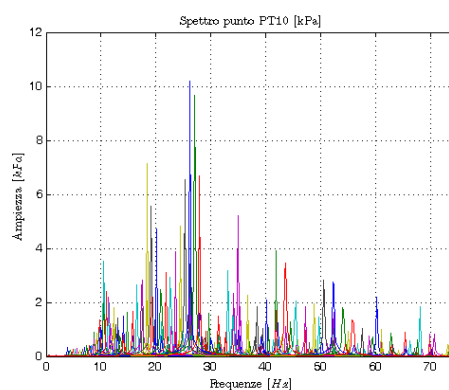


Figura 6.20: Sovrapposizione spettri oscillazioni di pressione: SS-PT10.

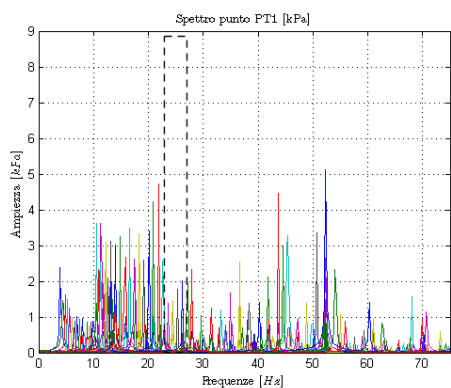


Figura 6.21: Sovrapposizione spettri oscillazioni di pressione:  $R_{24,7}^{SBS}$ -PT3-PT1.

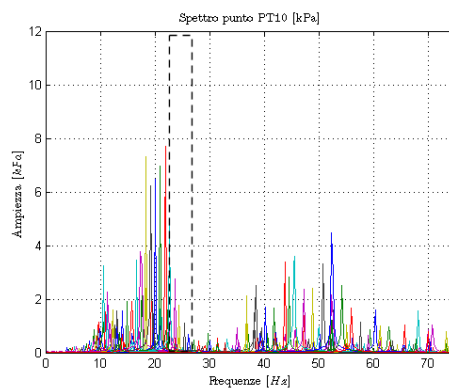


Figura 6.22: Sovrapposizione spettri oscillazioni di pressione:  $R_{24,7}^{SBS}$ -PT3-PT10.

I test effettuati hanno evidenziato un buon comportamento di questi dispositivi anche alle basse frequenze ( $5.5 \div 10.3 \text{ Hz}$ ): in questi casi tuttavia (figg. 6.25-6.27), l'abbattimento ottenibile è piuttosto ridotto poiché questo impianto non riesce a fornire sufficiente energia alle basse frequenze. Nelle figure dalla 6.22 alla 6.27 è evidente, per ciascun tipo di risonatore testato, un abbattimento delle oscillazioni in una banda piuttosto ristretta e centrata in corrispondenza della frequenza  $f_R$  di risonanza del risonatore. Gli effetti di amplificazione sono sempre piuttosto limitati e tali da rendere applicabile questa tipologia di risonatori nei casi in cui è richiesto un abbattimento concentrato ad una particolare frequenza di funzionamento dell'impianto. Dalle dimensioni di questi dispositivi risulta potenzialmente complessa l'adattabilità e l'utilizzo di un solo risonatore di questo tipo per far fronte a grandi variazioni di frequenza delle oscillazioni di pressione dinamica presenti nel condotto principale.

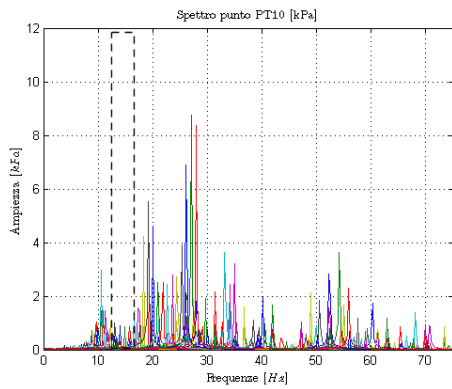


Figura 6.23: Sovrapposizione spettri oscillazioni di pressione:  $R_{14,6}^{SBS}$ -PT3-PT10.

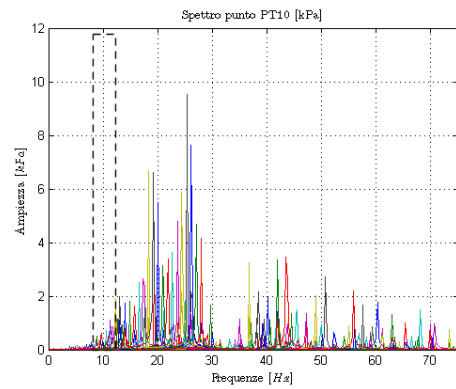


Figura 6.24: Sovrapposizione spettri oscillazioni di pressione:  $R_{10,3}^{SBS}$ -PT3-PT10.

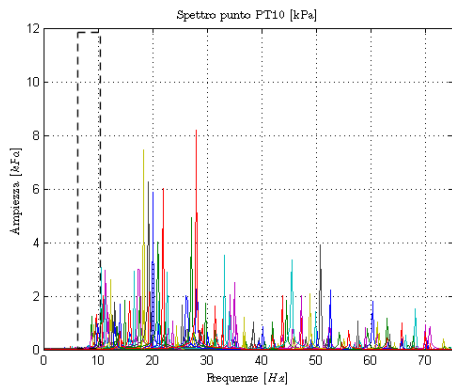


Figura 6.25: Sovrapposizione spettri oscillazioni di pressione:  $R_{8,0}^{SBS}$ -PT3-PT10.

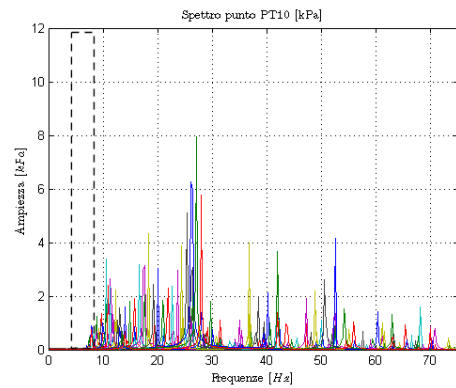


Figura 6.26: Sovrapposizione spettri oscillazioni di pressione:  $R_{6,5}^{SBS}$ -PT3-PT10.

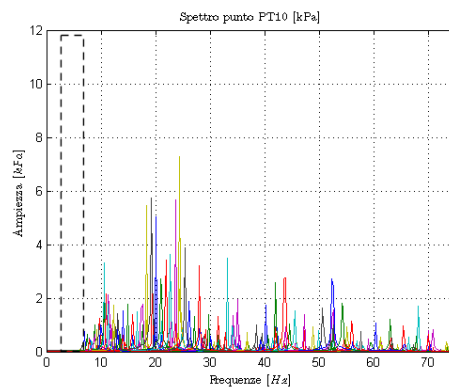


Figura 6.27: Sovrapposizione spettri oscillazioni di pressione:  $R_{5,5}^{SBS}$ -PT3-PT10.

### 6.1.3 Progettazione dei risonatori di Helmholtz

Di seguito viene mostrato il progetto e l'analisi sperimentale dei risonatori di Helmholtz a geometria fissa allo scopo di valutare le loro performance nell'abbattimento delle oscillazioni di pressione e confrontare queste con quelle ottenute con i risonatori side-branch esposti nei paragrafi precedenti. Considerando i risultati ottenuti dal modello analitico, presentato nel paragrafo 4.4.1, si sceglie da subito di realizzare dei risonatori di Helmholtz a geometria cilindrica per semplificarne la loro costruzione meccanica e con un diametro del collo  $D_C$  pari a quello del condotto principale  $d_{cnd}$  al fine di massimizzare le performance di abbattimento.

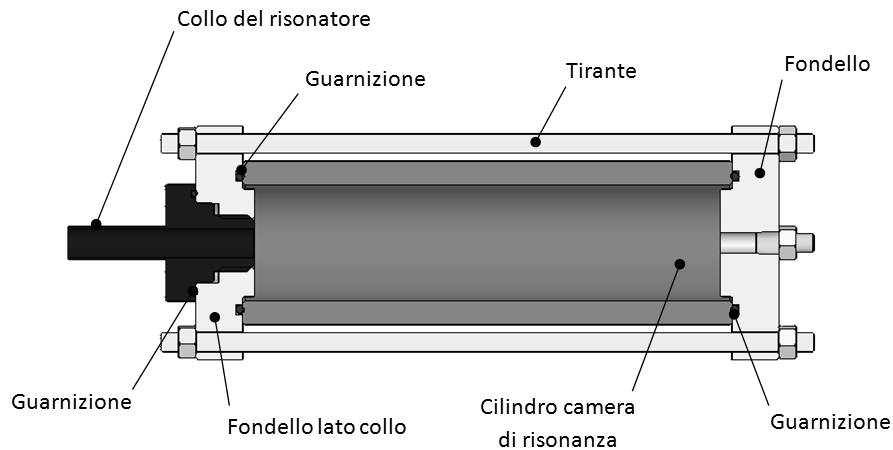


Figura 6.28: Sezione del risonatore di Helmholtz cilindrico a geometria fissa.

Nella figura 6.28 si riporta la sezione di uno dei risonatori progettati; esso risulta costituito dalle seguenti parti di cui vengono brevemente definiti i criteri di progettazione e scelta dei materiali:

- Collo del risonatore: realizzato utilizzando un tubo con le medesime caratteristiche di quelle del condotto principale.
- Cilindro camera di risonanza: realizzati mediante tubi in alluminio di differenti diametri che verranno definiti in seguito nelle diverse fase di progettazione e test. Il progetto della camera di risonanza viene eseguito considerando quest'ultima come un recipiente in pressione soggetto alle forze legate alla presenza dell'aria all'interno della camera alla pressione statica di 4 bar.



- Fondelli: ottenuti a partire da tondi pieni in alluminio e opportunamente progettati al fine di favorire l'alloggiamento delle guarnizioni di tenuta. È necessario infatti realizzare i risonatori garantendo un'adeguata tenuta tra il cilindro della camera di risonanza e i fondelli della stessa e tra la camera di risonanza e il collo del risonatore.
- Tiranti: ottenuti utilizzando tondi di acciaio Fe340. Lo scopo dei tiranti è quello di garantire la chiusura del cilindro della camera di risonanza tra i due fondelli e garantire un'adeguata pressione sulle guarnizioni di tenuta. I tiranti sono stati dimensionati considerando gli sforzi di trazione cui risultano soggetti al fine di limitare eccessivi allungamenti e quindi l'eventuale mancanza di una sufficiente tenuta all'interno della camera di risonanza.
- Guarnizioni: elementi di tipo O-Ring disponibili commercialmente e scelti in funzione della geometria della scanalatura di inserimento.

La progettazione dei risonatori di Helmholtz si riconduce al dimensionamento della lunghezza del collo del risonatore  $L_C$ , del diametro  $D_R$  della camera di risonanza e della sua lunghezza  $L_R$ . Nella figura 6.29 viene mostrato uno dei risonatori realizzati, dove vengono messe in evidenza le grandezze geometriche caratteristiche del risonatore di Helmholtz.

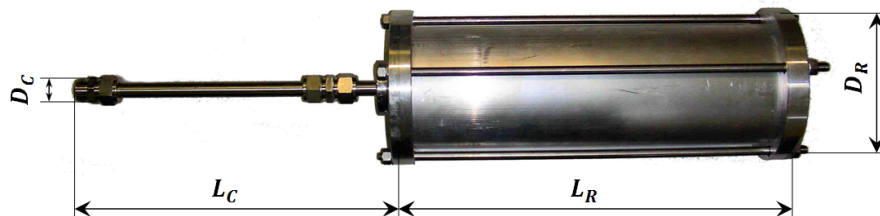


Figura 6.29: *Risonatore di Helmholtz cilindrico a geometria fissa: grandezze caratteristiche.*

Il dominio di ciascuna variabile geometrica del risonatore è limitato infatti dagli effetti di dissipazione e dalle non linearità presenti nel sistema reale, elementi che risultano difficilmente indagabili per mezzo di analisi puramente analitiche o numeriche. Il progetto e l'analisi dei risonatori di Helmholtz a geometria fissa viene suddiviso in diverse fasi progettuali dove, in ciascuna di queste, vengono considerate e analizzate differenti caratteristiche proprie del risonatore allo scopo di valutare variazioni nella performance di abbattimento.

### Prima fase progettuale

In questa fase si sceglie di realizzare dei risonatori con caratteristiche prossime a quelle disponibili in letteratura, ossia con lunghezze del collo  $L_C$  molto inferiori rispetto a quelle della camera di risonanza  $L_R$ : in particolare si vogliono identificare eventuali limiti sul diametro  $D_R$  della camera di risonanza o eventualmente sul volume  $V_R$  della camera. Nella tabella 6.3 si riportano i quattro risonatori realizzati in questa fase e testati sull'impianto.

Risonatore	$D_C$ [mm]	$L_C$ [mm]	$D_R$ [mm]	$L_R$ [mm]	$f_R$ [Hz]
$R_{36.3}^H$	12.4	118	50	1000	36.3
$R_{25.7}^H$	12.4	118	50	2000	25.7
$R_{20.2}^H$	12.4	118	90	1000	20.2
$R_{14.3}^H$	12.4	118	90	2000	14.3

Tabella 6.3: Risonatori di Helmholtz realizzati: prima fase progettuale.

Data la bassa variabilità delle performance di abbattimento dei risonatori side-branch al variare della posizione di inserimento di questi lungo il condotto principale dell'impianto, è stato scelto di inserire i risonatori di Helmholtz in corrispondenza del punto PT3. Nella figura 6.30 viene mostrato lo schema relativo al collegamento del risonatore  $R_{36.3}^H$  nel punto PT3 del condotto principale.

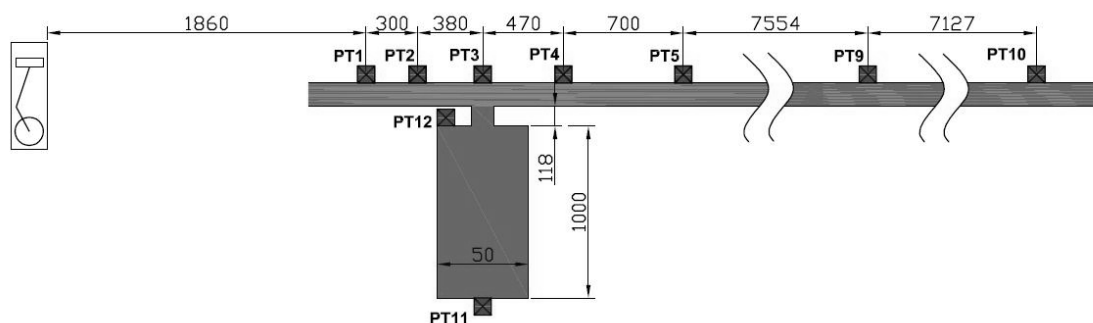


Figura 6.30: Risonatore di Helmholtz  $R_{36.3}^H$ : schema dell'impianto.

L'analisi delle performance dei risonatori di Helmholtz viene effettuata mediante la visualizzazione degli spettri sovrapposti, adottata in precedenza per lo studio dei risonatori side-branch. Sperimentalmente è emerso un comportamento di abbattimento simile nei diversi punti di misura sul condotto principale dell'impianto pertanto si

riportano i risultati ottenuti sul solo punto di misura PT4. Nella figura 6.31 si riporta l'andamento delle oscillazioni di pressione rilevate sull'impianto nel punto di misura PT4 in assenza di dispositivi risonanti: questo andamento viene confrontato con gli andamenti mostrati nelle figure dalla 6.32 alla 6.35, ottenute a seguito del collegamento nel punto PT3 dei risonatori riportati in tabella 6.3.

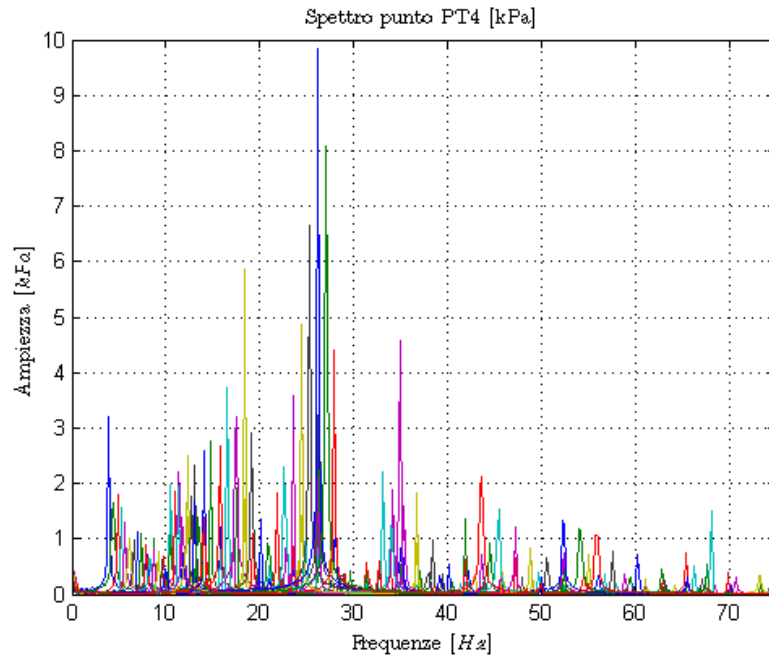


Figura 6.31: Sovrapposizione spettri oscillazioni di pressione: SS-PT4.

Dall'analisi delle figure 6.32 e 6.33 si evidenziano le elevate performance di abbattimento dei risonatori  $R_{36.3}^H$  e  $R_{25.7}^H$  in corrispondenza della loro frequenza di progetto  $f_R$  e un abbattimento generalizzato in tutto il campo di frequenze analizzato, mentre i risonatori  $R_{20.2}^H$  e  $R_{14.3}^H$  non mostrano un abbattimento marcato in corrispondenza di  $f_R$  ma solo l'abbattimento generalizzato (figure 6.34 e 6.35). Dai risultati sperimentali sembra emergere inoltre un limite sul rapporto dei diametri  $\frac{D_C}{D_R}$ .

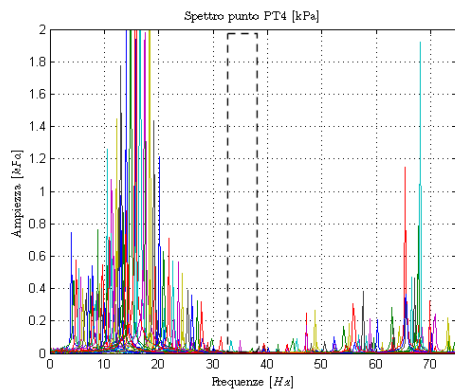


Figura 6.32: Sovrapposizione spettri oscillazioni di pressione:  $R_{36.3}^H$ -PT3-PT4.

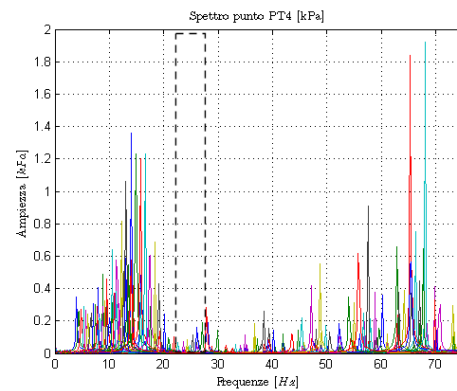


Figura 6.33: Sovrapposizione spettri oscillazioni di pressione:  $R_{25.7}^H$ -PT3-PT4.

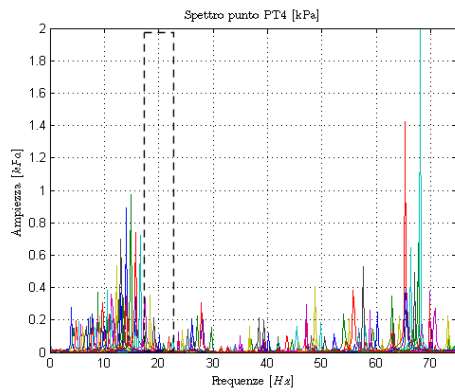


Figura 6.34: Sovrapposizione spettri oscillazioni di pressione:  $R_{20.2}^H$ -PT3-PT4.

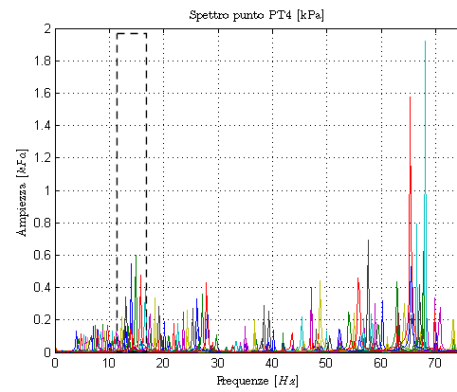


Figura 6.35: Sovrapposizione spettri oscillazioni di pressione:  $R_{14.3}^H$ -PT3-PT4.

## Seconda fase progettuale

Al fine di indagare meglio la presenza di una limitazione sul rapporto dei diametri  $\frac{D_C}{D_R}$ , si effettua il progetto di due risonatori con medesimo collo e medesimo volume della camera di risonanza ma con diametri  $D_R$  differenti. Nella tabella 6.4 si riportano le caratteristiche geometriche di questi risonatori.

Nelle figure 6.36 e 6.37 si confrontano le performance di abbattimento dei due risonatori riportati in tabella 6.4: è evidente che questi due risonatori hanno un comportamento del tutto equivalente, quindi la limitazione sul rapporto  $\frac{D_C}{D_R}$  è dettata solamente da un aumento eccessivo delle perdite localizzate e la mancanza di un abbattimento localizzato per i risonatori  $R_{20.2}^H$  e  $R_{14.3}^H$  è causato da un eccessivo volume  $V_R$  della camera di risonanza. L'abbattimento ridotto di risonatori con volume troppo elevato

è causato quindi dalla mancanza di sufficiente energia da parte dell'onda di pressione presente nel condotto principale dell'impianto: il volume massimo è dettato dalla massima ampiezza dell'oscillazione di pressione che eccita il risonatore, ossia l'ampiezza dell'onda di pressione che deve essere attenuata.

Risonatore	$D_C$ [mm]	$L_C$ [mm]	$D_R$ [mm]	$L_R$ [mm]	$f_R$ [Hz]
$R_{36.3}^H$	12.4	118	50	1000	36.3
$R_{36.3-2}^H$	12.4	118	90	310	36.3

Tabella 6.4: *Risonatori di Helmholtz realizzati: seconda fase progettuale.*

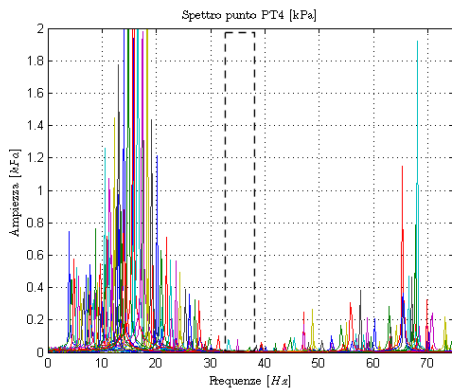


Figura 6.36: *Sovrapposizione spettri oscillazioni di pressione:  $R_{36.3}^H$ -PT3-PT4.*

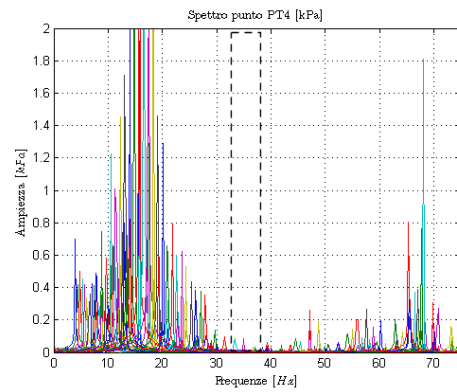


Figura 6.37: *Sovrapposizione spettri oscillazioni di pressione:  $R_{36.3-2}^H$ -PT3-PT4.*

Il limite sul volume massimo eccitabile dall'onda di pressione presente nel condotto principale si riconduce quindi, una volta fissato il diametro  $D_R$  della camera di risonanza, ad un limite della lunghezza massima  $L_R$  del risonatore.

### Terza fase progettuale

I risonatori progettati e testati in queste due prime fasi progettuali portano a dispositivi con frequenze di risonanza  $f_R$  troppo elevate rispetto al campo di frequenze di funzionamento dell'impianto. Dato che i risonatori side-branch non hanno evidenziato particolari limiti all'aumentare della loro lunghezza  $L_R$ , vengono progettati dei risonatori di Helmholtz con lunghezze del collo più elevate in modo da ridurre a parità di volume  $V_R$  della camera di risonanza la frequenza di progetto  $f_R$  di questi dispositi-

vi: nella tabella 6.5 si riportano i risonatori di Helmholtz progettati nella terza fase progettuale.

Risonatore	$D_C$ [mm]	$L_C$ [mm]	$D_R$ [mm]	$L_R$ [mm]	$f_R$ [Hz]
$R_{26.8}^H$	12.4	233	50	1000	26.8
$R_{18.8}^H$	12.4	493	50	1000	18.8
$R_{12.4}^H$	12.4	1168	50	1000	12.4
$R_{11.0}^H$	12.4	1500	50	1000	11.0

Tabella 6.5: Risonatori di Helmholtz realizzati: terza fase progettuale.

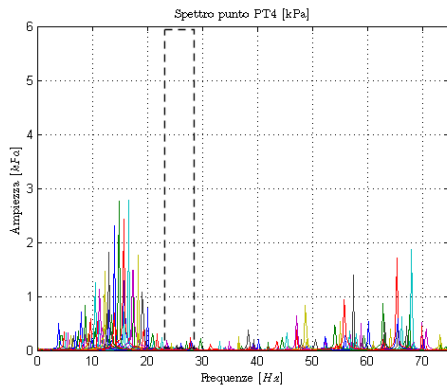


Figura 6.38: Sovrapposizione spettri oscillazioni di pressione:  $R_{26.8}^H$ -PT3-PT4.

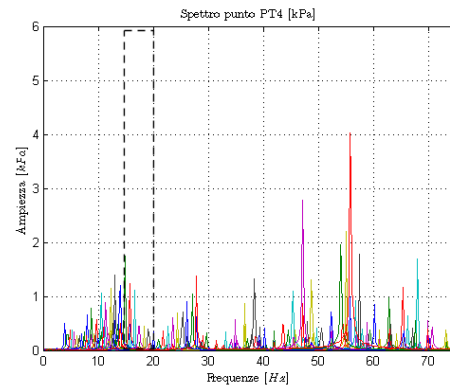


Figura 6.39: Sovrapposizione spettri oscillazioni di pressione:  $R_{18.8}^H$ -PT3-PT4.

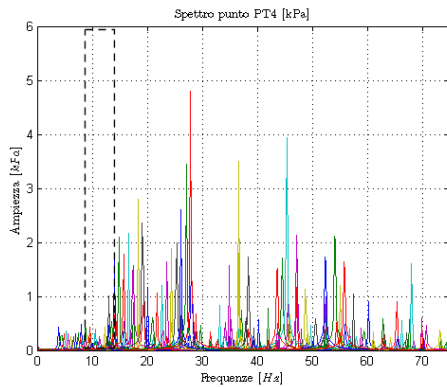


Figura 6.40: Sovrapposizione spettri oscillazioni di pressione:  $R_{12.4}^H$ -PT3-PT4.

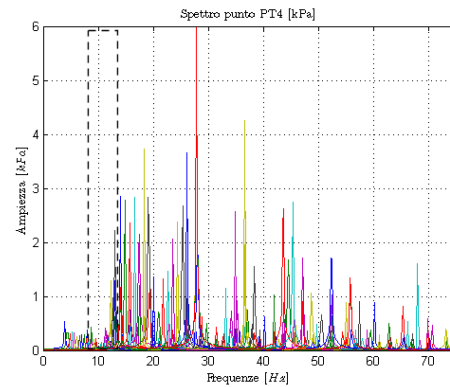


Figura 6.41: Sovrapposizione spettri oscillazioni di pressione:  $R_{11.0}^H$ -PT3-PT4.

Nelle figure dalla 6.38 alla 6.41 si mostrano gli andamenti delle ampiezze delle oscillazioni di pressione, rilevate nel punto PT4, in presenza dei risonatori di Helmholtz

riportati in tabella 6.5. Confrontando questi andamenti con l'andamento in assenza dei dispositivi risonanti (fig. 6.31), si evidenziano delle ottime performance di abbattimento per tutti e quattro i risonatori progettati nella terza fase, infatti l'abbattimento generalizzato è ridotto rispetto ai risonatori precedenti con volumi  $V_R$  più grandi (figure 6.34 e 6.35) e risulta più evidente l'abbattimento concentrato attorno alla frequenza di risonanza  $f_R$  del risonatore anche per le basse frequenze di risonanza (12.4 e 11.0 Hz).

#### Quarta fase progettuale

Date le elevate performance di abbattimento dei risonatori di Helmholtz evidenziate nelle analisi sperimentali e grazie alle loro caratteristiche geometriche, questi dispositivi si prestano maggiormente, rispetto ai risonatori side-branch, ad essere attuati. In questo modo è possibile poter variare alcune loro dimensioni geometriche e rendere variabile la loro frequenza di risonanza, in funzione della frequenza dell'oscillazione di pressione che si intende abbattere. La caratteristica geometrica più facilmente modificabile per mezzo di un sistema di attuazione è rappresentata sicuramente dalla lunghezza  $L_R$  della camera di risonanza. A tal fine vengono progettati e analizzati altri quattro risonatori con medesima lunghezza del collo  $L_C$  e medesimo diametro della camera di risonanza  $D_R$ . Nella tabella 6.6 si riportano le caratteristiche geometriche di questi risonatori: la quarta fase progettuale ha visto la realizzazione di risonatori con una lunghezza  $L_R$  della camera di risonanza pari al 20, 50 e 80 % di quella del risonatore con volume  $V_R$  pari a quello limite evidenziato nei risultati sperimentali precedenti.

Risonatore	$D_C$ [mm]	$L_C$ [mm]	$D_R$ [mm]	$L_R$ [mm]	$f_R$ [Hz]
$R_{24.6}^H$	12.4	1500	50	200	24.6
$R_{15.5}^H$	12.4	1500	50	500	15.5
$R_{12.3}^H$	12.4	1500	50	800	12.3
$R_{11.0}^H$	12.4	1500	50	1000	11.0

Tabella 6.6: *Risonatori di Helmholtz realizzati: quarta fase progettuale.*

Nelle figure dalla 6.42 alla 6.45 si mostrano gli andamenti delle ampiezze delle oscillazioni di pressione, rilevate nel punto PT4, in presenza dei risonatori di Helmholtz riportati in tabella 6.6.

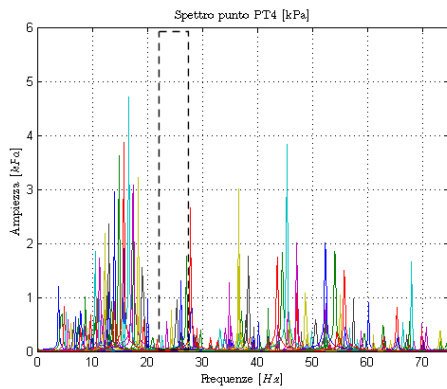


Figura 6.42: Sovrapposizione spettri oscillazioni di pressione:  $R_{24,6}^H$ -PT3-PT4.

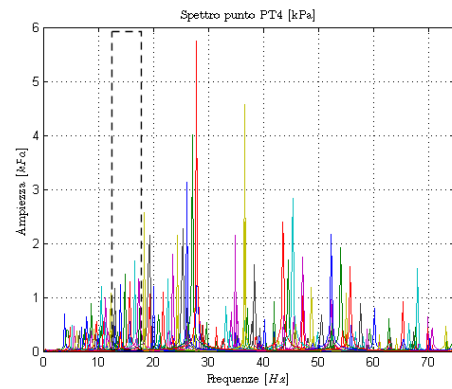


Figura 6.43: Sovrapposizione spettri oscillazioni di pressione:  $R_{15,5}^H$ -PT3-PT4.

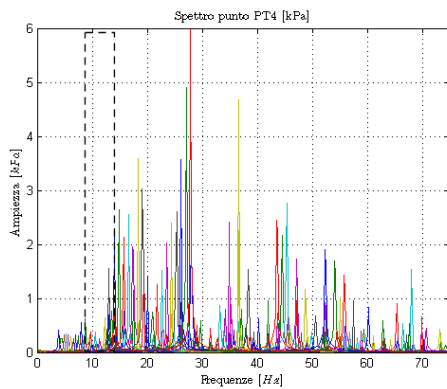


Figura 6.44: Sovrapposizione spettri oscillazioni di pressione:  $R_{12,3}^H$ -PT3-PT4.

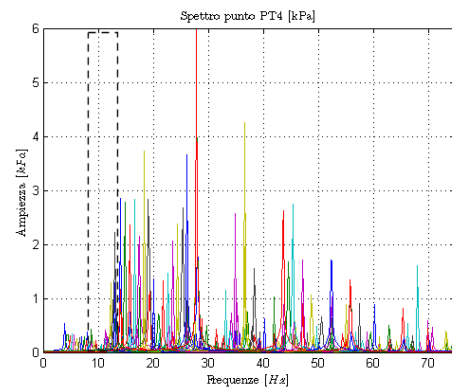


Figura 6.45: Sovrapposizione spettri oscillazioni di pressione:  $R_{11,0}^H$ -PT3-PT4.

Da questa ultima fase progettuale si evidenzia la possibilità di realizzare un risonatore con una lunghezza  $L_R$  della camera di risonanza variabile in funzione della frequenza  $f_R$  dell'oscillazione di pressione che si intende abbattere: nel paragrafo 6.2 viene presentato un risonatore di Helmholtz a geometria variabile progettato tenendo in considerazione i risultati ottenuti dall'analisi sperimentale dei risonatori a geometria fissa presentati nel presente paragrafo.



## Conclusioni

Dai risultati sperimentali ottenuti vengono delineati i criteri per la progettazione di un risonatore di Helmholtz a geometria variabile in grado di adattare la sua geometria alla frequenza dell'oscillazione di pressione presente nel funzionamento dell'impianto.

Dalle analisi sperimentali è possibile affermare quanto segue:

- Esiste un diametro del collo ottimale tale da massimizzare l'energia in ingresso al risonatore:

$$D_C = d_{cmd}. \quad (6.5)$$

- Esiste un limite al rapporto  $\frac{D_C}{D_R}$  dettato da un eccessivo aumento delle perdite localizzate che si hanno nella propagazione dell'onda di pressione tra il collo e la camera di risonanza del risonatore. Secondo i risultati sperimentali ottenuti in precedenza viene scelto un rapporto pari a:

$$\frac{D_C}{D_R} = \frac{1}{4}. \quad (6.6)$$

- Esiste un limite sulla lunghezza  $L_R$  massima della camera di risonanza dettato dal massimo volume in grado di essere eccitato dall'impianto di test. Il massimo volume  $V_R$  eccitabile dall'impianto limita la minima frequenza dell'oscillazione di pressione in grado di essere attenuata dal risonatore di Helmholtz.
- La minima frequenza di risonanza dell'oscillazione di pressione, in grado di essere abbattuta con questo dispositivo, può essere regolata agendo sulla lunghezza  $L_C$  del collo del risonatore.
- La massima frequenza di risonanza dell'oscillazione di pressione, in grado di essere abbattuta con questo dispositivo, è data dalla massima riduzione del volume  $V_R$  ossia dalla più piccola lunghezza  $L_R$  ottenibile.

## 6.2 Risonatore di Helmholtz a geometria variabile

A partire dai risultati sperimentali ottenuti con i risonatori di Helmholtz a geometria fissa si procede con la realizzazione di un risonatore di Helmholtz a geometria variabile in grado di coprire il più ampio range di frequenze di funzionamento del compressore dell'impianto di test. Considerando il range di frequenze delle oscillazioni di pressione nel funzionamento del compressore  $7.5 \leq f_R \leq 26.7 \text{ Hz}$  e le osservazioni appena esposte, vengono determinate le caratteristiche geometriche del risonatore a geometria variabile:

- Diametro del collo  $D_C$  del risonatore pari al diametro del condotto principale al fine di massimizzare l'energia in ingresso e in uscita dal collo del risonatore e massimizzare quindi gli abbattimenti ottenibili:  $D_C = d_{cnd} = 12.4 \text{ mm}$ .
- Diametro della camera di risonanza  $D_R$  dettato dal limite sulle perdite localizzate e pari quindi a  $D_R = 4D_C \simeq 50.0 \text{ mm}$ .
- Considerando il massimo volume in grado di essere eccitato dal sistema, ottenuto dalle indagini sperimentali viste nel paragrafo 6.1.3, la massima lunghezza  $L_R$  della camera di risonanza, con un diametro della camera  $D_R = 50.0 \text{ mm}$ , è pari a  $L_{Rmax} = 1000.0 \text{ mm}$ .
- La lunghezza del collo viene scelta in modo tale da consentire al risonatore di abbattere la minima frequenza presente nel funzionamento dell'impianto. Si consideri a tal fine la figura 6.46, dove viene mostrato l'andamento della frequenza di risonanza  $f_R$  del risonatore in funzione della lunghezza del collo per due diversi risonatori: la curva blu indica un risonatore con lunghezza della camera di risonanza pari a quella dettata dal volume limite ( $L_{Rmax} = 1000.0 \text{ mm}$ ), mentre la curva rossa può essere ottenuta utilizzando un risonatore con lunghezza della camera di risonanza pari a quella minima in grado di essere raggiunta con il dispositivo a geometria variabile; si ipotizza in questa fase un volume minimo della camera di risonanza ottenibile con una lunghezza pari a  $L_{Rmin} = 200.0 \text{ mm}$ . Dato il range di frequenze di funzionamento del compressore, quindi il range di frequenze delle oscillazioni di pressione da abbattere nell'impianto, la lunghezza del collo  $L_C$  viene scelta in modo da centrare il campo di frequenze di risonanza del risonatore a geometria variabile con il campo di frequenze delle oscillazioni di pressione presenti nel condotto principale al variare della velocità di rotazione dell'albero del compressore, come indicato in figura 6.46. Centrando i due

range di frequenze è possibile scegliere una lunghezza del collo del risonatore a geometria variabile pari a  $L_C = 1500.0 \text{ mm}$ .

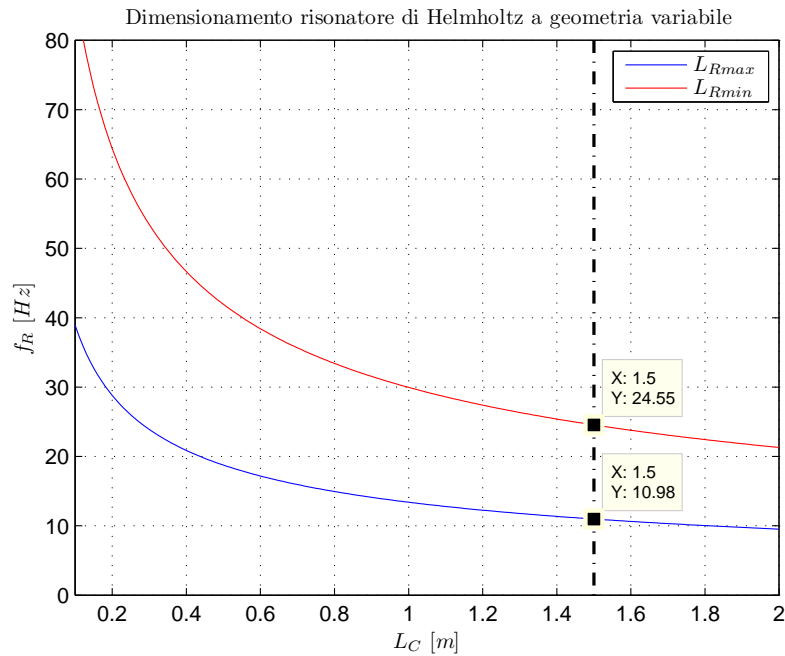


Figura 6.46: Dimensionamento risonatore Helmholtz a geometria variabile.

Nella tabella 6.7 vengono riassunte le caratteristiche geometriche del risonatore di Helmholtz a geometria variabile di cui verrà esposta la progettazione meccanica nel paragrafo 6.2.1.

Caratteristica	Valore [mm]
$D_C$	12.4
$L_C$	1500.0
$D_R$	50.0
$L_R$	200.0 ÷ 1000.0

Tabella 6.7: Caratteristiche del risonatore di Helmholtz a geometria variabile.

Nella figura 6.47 si riporta una schema del risonatore di Helmholtz a geometria variabile collegato al condotto principale.

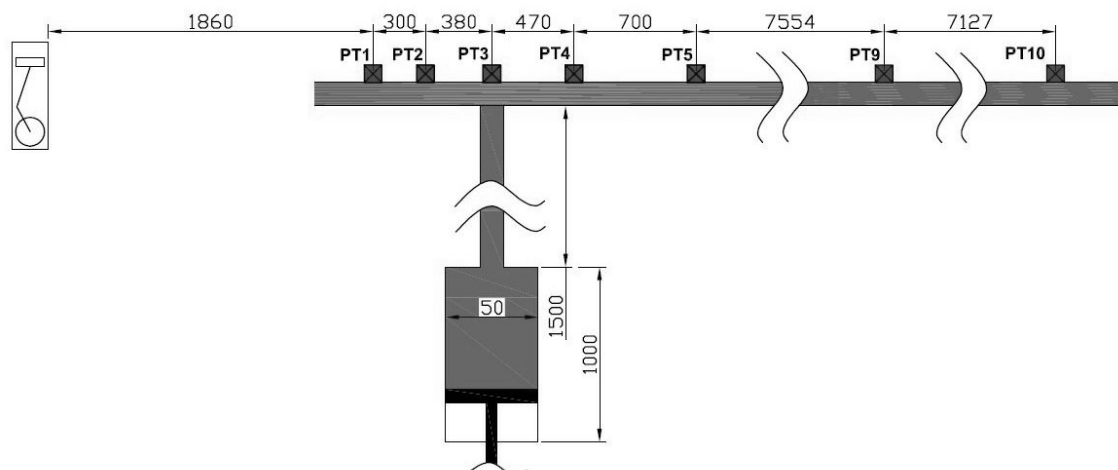


Figura 6.47: Schema del risonatore di Helmholtz a geometria variabile collegato al condotto principale.

### 6.2.1 Progettazione meccanica del dispositivo

La realizzazione del risonatore di Helmholtz a geometria variabile è stata effettuata utilizzando un cilindro pneumatico disponibile commercialmente, opportunamente modificato in modo da consentire il collegamento del collo. La variazione della posizione del pistone è stata ottenuta mediante l'utilizzo di un sistema attivo costituito da un motore brushless, e da una trasmissione con vite e guide lineari a ricircolo di sfere. La scelta del cilindro pneumatico commerciale è stata dettata dalla necessità di realizzare in breve tempo un risonatore di Helmholtz a geometria variabile che garantisca un'adeguata tenuta per l'aria alla pressione statica di 4 *bar*, tra il cilindro e lo stantuffo. Sperimentalmente è emerso infatti che la presenza di luci tra il cilindro e lo stantuffo causano un eccessivo peggioramento delle performance di abbattimento del risonatore, che deve quindi essere realizzato garantendo una adeguata tenuta sullo stantuffo. Nella figura 6.48 viene mostrato il cilindro Festo modello DNG-50-1000-PPV utilizzato per la realizzazione della camera di risonanza del dispositivo. Per garantire il collegamento del collo è stato modificato il fondello del cilindro indicato col numero 1 in figura 6.48. Il collo è stato realizzato utilizzando un tubo con le stesse caratteristiche del condotto principale e dei colli utilizzati nei risonatori a geometria fissa.

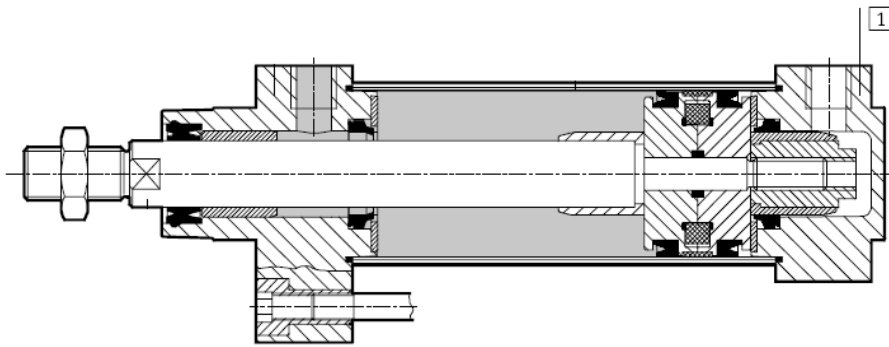


Figura 6.48: *Cilindro pneumatico Festo DNG-50-1000-PPV: vista in sezione.*

Per le caratteristiche costruttive di questo risonatore è possibile scendere al di sotto del volume minimo scelto durante la progettazione e posizionare lo stantuffo del cilindro al di sotto di  $200.0\text{ mm}$ . Questo potrebbe consentire di estendere il range di funzionamento del risonatore a frequenze di risonanza  $f_R \geq 24.55\text{ Hz}$ : tale caratteristica verrà valutata durante la fase di indagine sperimentale (par. 6.4). Nella figura 6.49 si riporta una foto del risonatore di Helmholtz a geometria variabile collegato al condotto principale dell'impianto e realizzato secondo quanto esposto nel presente paragrafo.

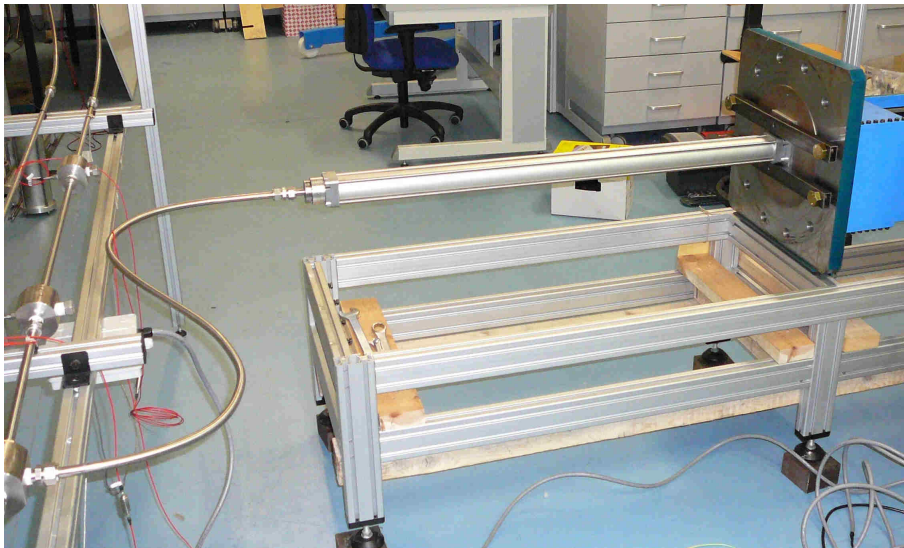


Figura 6.49: *Foto del risonatore di Helmholtz a geometria variabile realizzato.*

Per la movimentazione dello stantuffo è stata progettata una trasmissione con guide lineari e vite a ricircolo di sfere per garantire un corretto allineamento e posizionamento dello stantuffo lungo l'asse principale del cilindro, nonché garantire la trasmissione di

un'adeguata forza in grado di mantenere in una posizione prefissata lo stantuffo durante il funzionamento del compressore (appendice F.1). Considerando un possibile utilizzo del meccanismo di trasmissione per testare, in futuri sviluppi del progetto di tesi, risonatori con geometrie differenti e indagare dettagliatamente eventuali limitazioni sul diametro massimo della camera di risonanza si è proceduto a sovra-dimensionare il sistema di trasmissione considerando un risonatore a geometria variabile con diametro massimo dello stantuffo pari a  $D_{Rmax} = 300 \text{ mm}$  e lunghezza pari a  $L_{Rmax} = 1 \text{ m}$ .

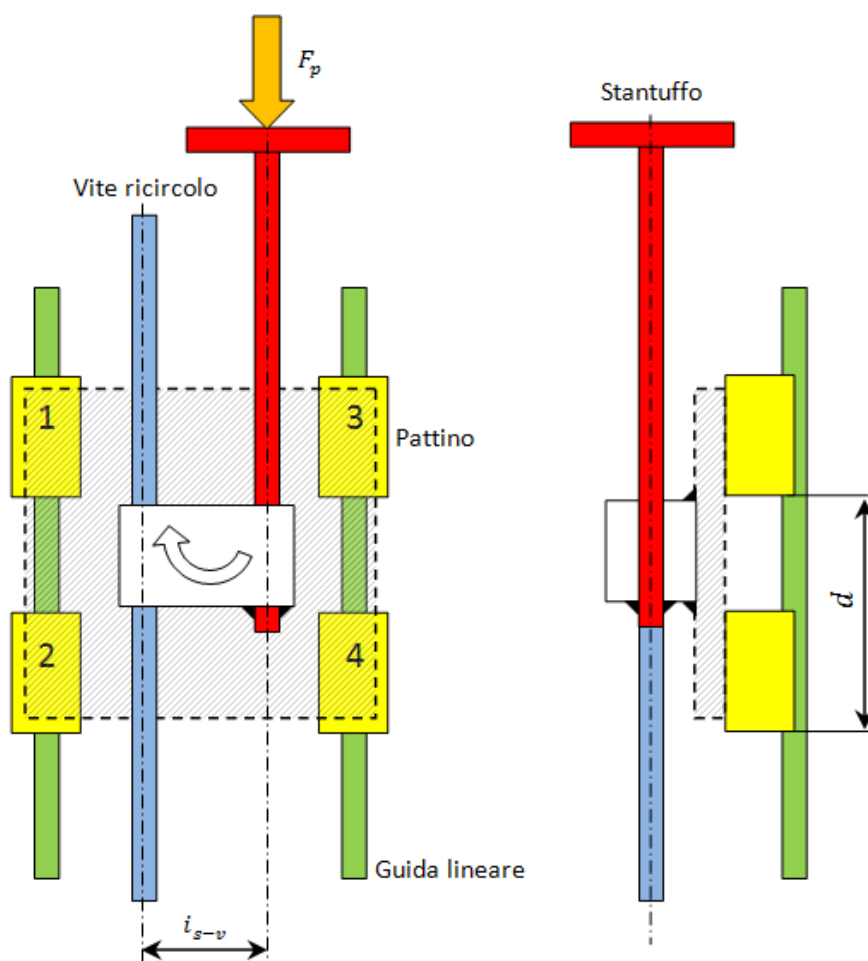


Figura 6.50: Schema trasmissione per risonatore di Helmholtz a geometria variabile.

Il progetto del meccanismo di trasmissione viene inoltre effettuato considerando la pressione massima, pari a  $p_0 = 4 \text{ bar}$ , realizzabile dal compressore dell'impianto di test. Noti il rapporto di compressione massimo e il diametro dello stantuffo del risonatore più grande collegabile al sistema di trasmissione, la forza  $F_p$  agente su quest'ultimo

risulta pari a:

$$F_p = p_0 \pi \frac{D_{Rmax}^2}{4} = 28274 \text{ N}, \quad (6.7)$$

approssimata quindi a  $F_p = 30000 \text{ N}$ . Nella figura 6.50 si riporta lo schema della trasmissione che si intende realizzare in cui vengono messe in evidenza le principali dimensioni geometriche per il dimensionamento secondo la procedura riportata nel datasheet del costruttore delle guide e delle viti a ricircolo di sfere. La progettazione e la scelta dei pattini delle guide lineari viene effettuata in funzione della durata (vita utile in  $km$ ) secondo quanto riportato nel datasheet del produttore. Come produttore delle guide lineari e delle viti a ricircolo è stata scelta la casa costruttrice *HI-WIN* il cui riferimento al datasheet è disponibile in bibliografia [32]. Data la particolare configurazione del sistema di trasmissione, tra lo stantuffo del cilindro e la vite a ricircolo è presente un interasse  $i_{s-v}$ , scelto pari a  $110 \text{ mm}$ . Esso comporta la presenza di un carico agente sui pattini delle guide lineari che devono essere in grado di sopportare tale sforzo (fig. 6.50). Il carico dinamico  $C_{dyn-pattini}$  applicato ai pattini risulta pari a:

$$C_{dyn-pattini} = \frac{F_p i_{s-v}}{2d} = 7500 \text{ N}, \quad (6.8)$$

dove  $d$  è la distanza dei pattini lungo l'asse dello stantuffo. A seguito del carico applicato è necessario determinare la durata  $D_{pattini}$  delle guide lineari, secondo la relazione fornita dal produttore:

$$D_{pattini} = 50 \left( \frac{C_{pattino}}{C_{dyn-pattini}} \right)^3, \quad (6.9)$$

dove  $C_{pattino}$  è il carico dinamico relativo alla tipologia di pattini scelti. Scegliendo di utilizzare pattini del tipo *HGH 30 CA*, il valore di  $C_{pattino}$  è pari a  $38740 \text{ N}$ , la durata dei pattini scelti risulta  $D_{pattini} = 6900 \text{ km}$  e quindi sufficientemente ampia per le prove che si intendono effettuare con il risonatore a geometria variabile. Anche la vite a ricircolo è stata scelta secondo il procedimento di verifica della durata indicata dal costruttore [33]. Considerando un carico applicato pari a  $F_p$ , il carico dinamico agente viene ottenuto secondo la seguente relazione:

$$C_{dyn-vite} = f_p F_p, \quad (6.10)$$

dove il coefficiente  $f_p$  indica il tipo di funzionamento cui è sottoposta la vite a ricircolo; considerando un funzionamento della vite in assenza di impatti viene scelto un coefficiente  $f_p = 1.1$ , secondo quanto indicato nel datasheet delle viti a ricircolo *HI-WIN*.

Il carico dinamico agente sulla vite risulta pari a  $C_{dyn-vite} = 31102 N$ . Viene scelta la vite a ricircolo *HI-WIN FSV 50-10B3* con carico dinamico resistente  $C_{vite}$  pari a  $8394 kgf$  e passo della vite pari a  $p_{vite} = 10 mm/giro$ ; la durata  $D_{vite}$  della vite scelta risulta pari a:

$$D_{vite} = \left( 9.8 \frac{C_{vite}}{C_{dyn-vite}} \right)^3 10^6, = 1.85 \cdot 10^7 \quad (6.11)$$

espressa in numero di giri della vite a ricircolo in tutto il suo utilizzo. Nella scelta del sistema di attuazione (par. 6.2.2) occorrerà verificare se il passo della vite scelta è sufficiente a garantire un buon funzionamento dell'attuatore, altrimenti occorre procedere alla scelta di una nuova vite a ricircolo con un passo differente. A causa della lunghezza elevata della vite, che deve essere almeno pari ad  $1 m$  al fine di garantire il posizionamento di uno stantuffo altrettanto lungo, è necessario verificare la vite alla deflessione assiale determinando il carico di buckling ottenuto come [33]:

$$F_k = 0.540720 \left( \frac{N_f d_r^4}{L_t^2} \right), \quad (6.12)$$

dove  $N_f$  è un fattore che dipende dal tipo di montaggio scelto,  $d_r$  in  $mm$  è il diametro di base della vite a ricircolo e  $L_t$  è la distanza espressa in  $mm$  tra i supporti dei cuscinetti della vite a ricircolo. Da un primo dimensionamento si a:

- Coefficiente  $N_f$  pari a 0.5 avendo scelto un montaggio della vite di tipo *fixed-supported*. La figura F.2 nell'appendice F.1 riporta i principali elementi della trasmissione realizzata dove vengono messi in evidenza i supporti della vite a ricircolo.
- La determinazione del diametro di base  $d_r$  della vite tiene conto della dimensione effettiva delle sfere contenute. Per la vite a ricircolo scelta il diametro delle sfere è pari a  $6.35 mm$ , pertanto il diametro di base viene determinato sottraendo il diametro di una sfera al diametro nominale della vite:  $d_r = 50 - 6.35 = 43.65 mm$ .
- La distanza tra i supporti viene scelta pari a  $L_t = 1000 mm$ .

Considerando i valori appena definiti il carico di buckling risulta pari a  $F_k = 36956 kgf$ ; essendo superiore alla forza  $F_p = 30000 N$  del carico applicato si ritiene che la vite sia correttamente dimensionata. Per il dimensionamento corretto della vite occorre infine verificare che la velocità critica di rotazione della vite sia superiore alla velocità massima di funzionamento della stessa, al fine di evitare l'utilizzo della vite in corrispondenza



di velocità tali da eccitare la sua prima frequenza naturale. La determinazione della velocità critica della vite viene effettuata secondo la relazione fornita dal costruttore:

$$N_c = 2.7110^8 \left( \frac{M_f d_r^4}{L_t^2} \right), \quad (6.13)$$

dove  $M_f$  è un fattore dipendente dal tipo di montaggio scelto, che nel caso in oggetto risulta pari a 0.692 (montaggio di tipo *fixed-supported*). Con i dati a disposizione il valore della velocità critica risulta pari a  $N_c = 8185 \text{ RPM}$ : durante il posizionamento dello stantuffo è necessario evitare di raggiungere velocità di rotazioni della vite superiori a tale valore.

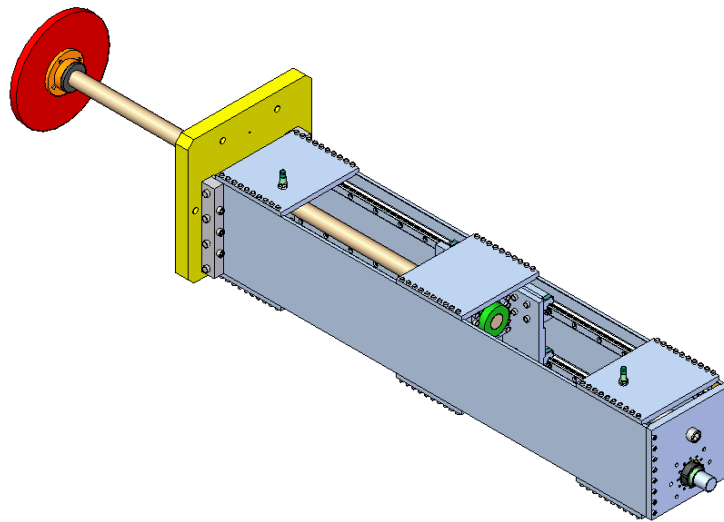


Figura 6.51: *Assieme complessivo del risonatore di Helmholtz a geometria variabile.*

La scelta delle guide lineari e della vite a ricircolo permette di proseguire nella progettazione del sistema di movimentazione e realizzare il disegno di assieme del risonatore a geometria variabile e dell'intero sistema di movimentazione. Nella figura 6.51 si riporta il disegno complessivo di assieme del risonatore e del sistema di trasmissione realizzato, mentre nella figura 6.52 vengono messi in evidenza i particolari della trasmissione dimensionati in questo paragrafo. La scatola di contenimento della trasmissione viene realizzata in Fe 340 e viene dimensionata per garantire un'adeguata resistenza meccanica per gli sforzi cui risulta soggetta. Dalla figura 6.51 risulta evidente la particolare struttura di contenimento del sistema di trasmissione: questa struttura, studiata appositamente, consente di racchiudere le forze all'interno del sistema di trasmissione evitando che le forze in gioco vengano trasmesse a terra.

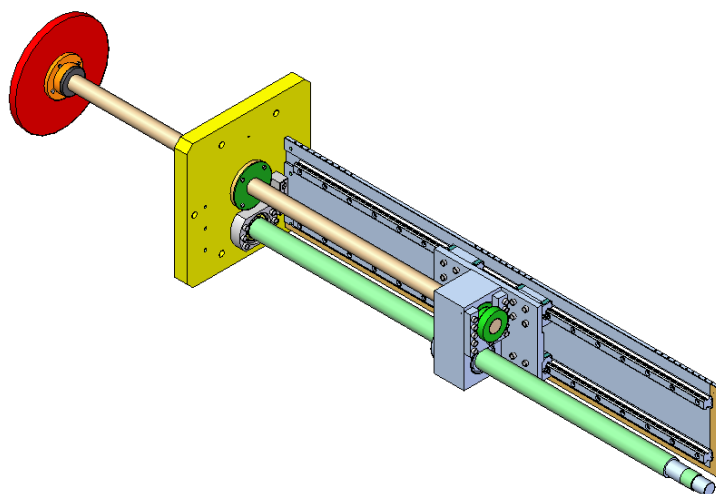


Figura 6.52: *Asieme complessivo del risonatore di Helmholtz a geometria variabile: vista di alcuni particolari della trasmissione.*

Nella figura 6.53 viene mostrato il trasferimento delle forze, indicato con le frecce rosse, che si racchiude completamente dallo stantuffo alla vite a ricircolo; queste forze vengono trasferite alla scatola di contenimento e alle piastre di sostegno della vite a ricircolo di cui si deve opportunamente verificare la loro resistenza meccanica. L'unica zona di contatto tra il sistema di trasmissione e la terra è costituita dal piede di appoggio, che consente di sostenere l'intero sistema di trasmissione.

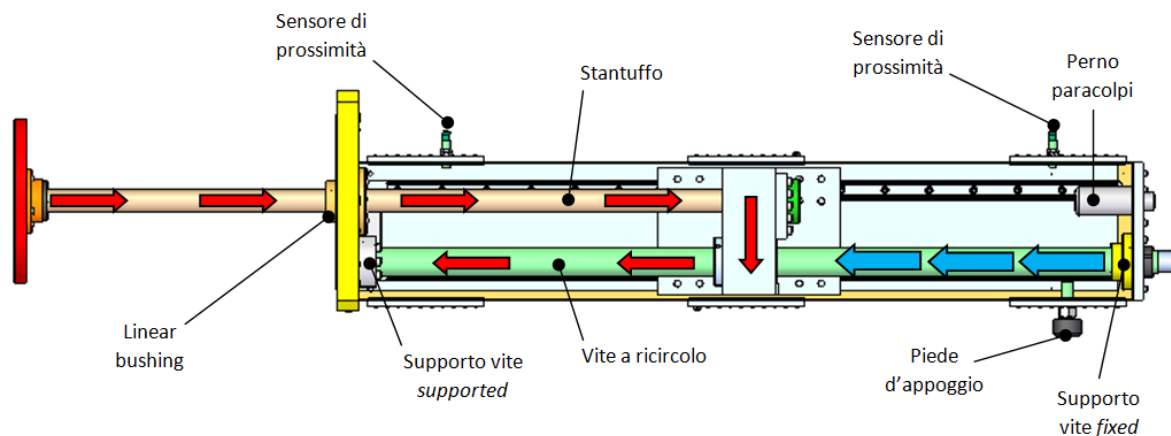


Figura 6.53: *Schema trasmissione risonatore di Helmholtz a geometria variabile: trasmissione delle forze.*

Per racchiudere completamente le forze all'interno della scatola di contenimento della trasmissione occorre procedere ad una verifica della resistenza meccanica di alcune parti della scatola soggette alle forze in questione: data la complessità della geometria di queste parti queste verifiche sono state effettuate attraverso delle analisi ad elementi finiti (*FEM*) realizzate utilizzando Comsol Multiphysics. La figura 6.56 riporta il risultato della verifica degli sforzi eseguita sulla piastra di *supporto fixed* della vite a ricircolo di sfere (mostrata in figura 6.53). Essendo un *supporto fixed*, tutte le forze cui risulta soggetta la vite a ricircolo vengono trasmesse a questa piastra che deve essere opportunamente verificata agli sforzi di flessione derivanti dall'applicazione del carico. Nell'elenco seguente vengono mostrati in dettaglio i dati utilizzati nell'analisi ad elementi finiti del particolare in questione:

- La figura 6.54 mostra in blu le parti della piastra soggette ad un vincolo di non spostamento, infatti queste parti sono rigidamente vincolate al resto del telaio di supporto della trasmissione.
- In verde (fig. 6.54) viene indicata la corona circolare su cui sono state applicate le forze che vengono trasmesse dalla vite a ricircolo. L'area  $A_{cc} = 3805 \text{ mm}^2$  della corona circolare risulta soggetta alla forza  $F_p = 30000 \text{ N}$ , quindi il carico  $S_{cc}$  applicato sul modello ad elementi finiti risulta pari a:

$$S_{cc} = \frac{F_p}{A_{cc}} \simeq 8 \text{ MPa}. \quad (6.14)$$

- Nella figura 6.55 viene mostrata la mesh utilizzata per la discretizzazione della piastra. In particolare è stata utilizzata una mesh con un totale di 5033 elementi tetraedrici, con un maggiore affinamento in corrispondenza della zona di applicazione del carico al fine di ottenere un risultato più preciso nella zona di interesse.

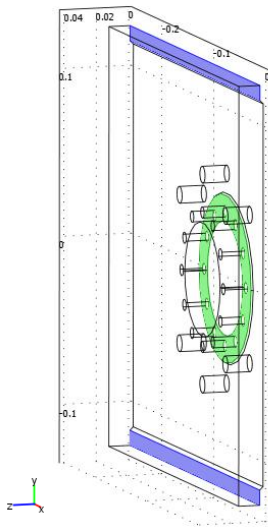


Figura 6.54: *Analisi FEM resistenza meccanica piastra supporto vite a ricircolo: applicazione delle condizioni al contorno e dei carichi.*

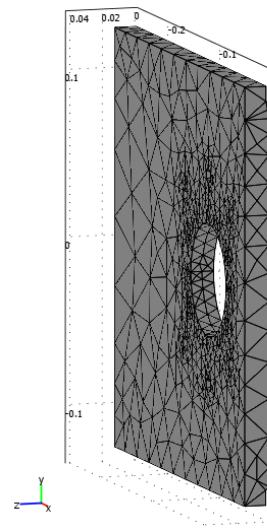


Figura 6.55: *Analisi FEM resistenza meccanica piastra supporto vite a ricircolo: mesh analisi FEM.*

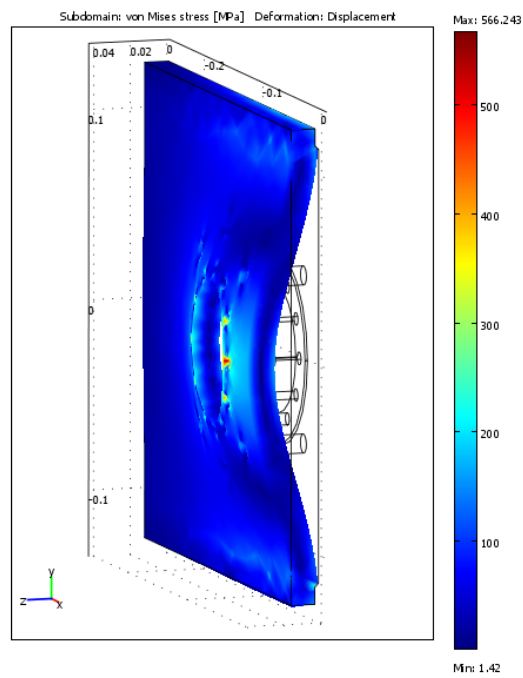


Figura 6.56: *Analisi FEM resistenza meccanica piastra supporto vite a ricircolo: sforzi di von Mises e deformazioni.*

### 6.2.2 Scelta degli attuatori

Definito il sistema di trasmissione con i suoi principali elementi è necessario scegliere adeguatamente l'attuatore in grado di movimentare lo stantuffo del risonatore nelle normali condizioni di funzionamento del compressore, considerando i carichi effettivamente agenti sulla superficie dello stantuffo. Come tipologia di attuatore per la movimentazione del sistema è stato scelto di utilizzare un motore brushless del produttore *Parker S.B.C.* I dati a disposizione per la scelta del motore elettrico sono i seguenti:

- Passo della vite a ricircolo  $p_{vite} = 10 \text{ mm/giro}$ .
- Velocità lineare massima di movimentazione dello stantuffo pari a  $v_{lin} = 20 \text{ mm/s}$ .  
La scelta di questo valore è dettato principalmente da due obiettivi: in primo luogo si vuole garantire il posizionamento dello stantuffo in condizioni di sicurezza, ossia garantendo una non eccessiva variazione della pressione statica a seguito di una rapida variazione della posizione dello stantuffo, e in secondo luogo si vuole poter posizionare dello stantuffo in un tempo ragionevole.
- Carico agente sullo stantuffo pari a  $F_p = 30000 \text{ N}$ .

Considerando il carico agente sullo stantuffo è possibile determinare la coppia agente sulla vite a ricircolo al fine di garantire una certa posizione dello stantuffo:

$$C_p = \frac{F_p p_{vite}}{2\pi} = 45 \text{ Nm}. \quad (6.15)$$

Dato l'elevato valore di coppia necessario a mantenere in posizione lo stantuffo occorre inserire tra la vite a ricircolo e il motore un ulteriore sistema di riduzione in grado di ridurre la coppia richiesta al motore, quindi ridurre le dimensioni e i costi di quest'ultimo. Volendo utilizzare un motore brushless di dimensioni contenute avente una coppia nominale massima di  $2 \text{ Nm}$ , si sceglie di utilizzare un riduttore epicicloidale con rapporto di riduzione  $\tau_{rid} = 1/28$ . Considerando tale rapporto di riduzione la coppia  $C_{mot}$  richiesta al motore è pari a:

$$C_{mot} = C_p \tau_{rid} = 1.6 \text{ Nm}. \quad (6.16)$$

Occorre anche determinare il numero di giri nominali di funzionamento del motore che vengono calcolati a partire dalla velocità  $v_{lin}$ :

$$N_{mot} = 60 v_{lin} \frac{1}{p_{vite} \tau_{rid}} = 3360 \text{ RPM}. \quad (6.17)$$

Dal datasheet del produttore di motori brushless viene scelto un motore che soddisfa i valori di coppia e numero di giri appena determinati, in particolare viene scelto il motore con alimentazione trifase prodotto da *Parker S.B.C.*, modello *SMBA-82-56-03-8-14-S-2D-64-400* (fig. 6.57). Per motivi di sicurezza è stata scelta la versione del motore con freno integrato in modo da garantire, anche in caso di mancata energia alimentazione, il mantenimento della posizione dello stantuffo. Parallelamente alla scelta del motore è stato selezionato l'azionamento elettrico prodotto da *Parker S.B.C.*, modello *SPD5N* (figura 6.58), per il controllo del motore. Tale azionamento verrà inserito all'interno del quadro elettrico di controllo del risonatore (par. 6.2.3) e verrà comandato da un opportuno sistema di controllo che consentirà di fornire il segnale di comando al motore in funzione della posizione che dovrà essere raggiunta dallo stantuffo del risonatore.



Figura 6.57: Motore brushless Parker S.B.C. SMBA-82-56-03-8-14-S-2D-64-400.



Figura 6.58: Azionamento elettrico Parker S.B.C. SPD5N.

Il riduttore epicicloidale scelto che soddisfa i requisiti di progettazione è prodotto da *Alpha Wittenstein*, modello *SP+075 MF*,  $i=28$ . È inoltre necessario garantire tra l'albero del riduttore e la vite a ricircolo un opportuno disaccoppiamento in grado di consentire eventuali disallineamenti di montaggio tra i due componenti: a tal fine è stato scelto di inserire tra il riduttore e la vite il giunto elastico di tipo *ROTEX-GS28 AL-H 98Sh A-GS 2,5-22 2,5-35*, prodotto da *KTR*, in grado di trasmettere la coppia  $C_p$  dal riduttore alla vite a ricircolo della trasmissione. Nella figura 6.59 viene mostrato in dettaglio la trasmissione del moto dal motore alla vite a ricircolo della trasmissione dimensionata nel paragrafo 6.2.1.

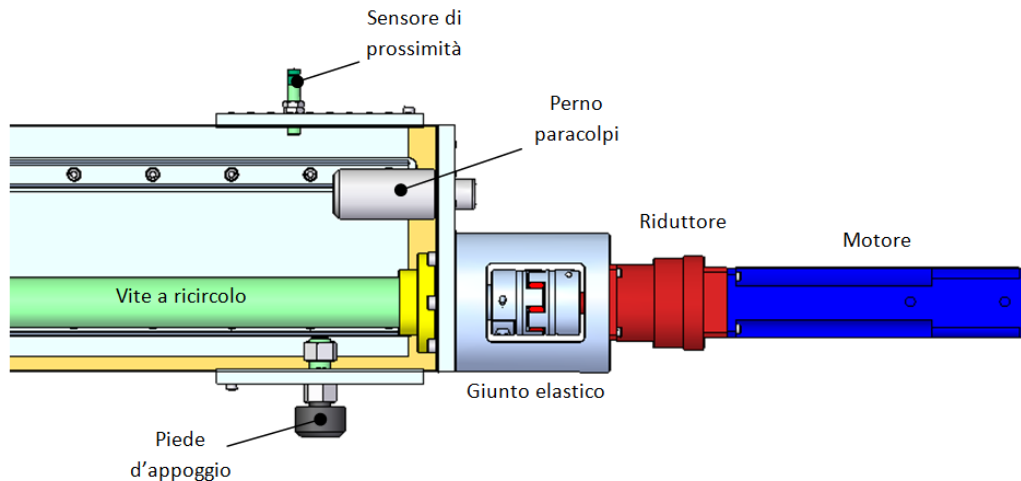


Figura 6.59: *Trasmissione del moto dal motore alla vite a ricircolo.*

### 6.2.3 Progettazione del circuito elettrico

Il controllo del motore brushless che movimentata il risonatore avviene tramite l'azionamento SPD5N, opportunamente comandato dal sistema di controllo. È necessario progettare e realizzare un opportuno circuito elettrico che consenta di integrare il sistema di controllo del risonatore con il circuito elettrico del compressore e con l'azionamento del motore del risonatore, principalmente per garantire la sicurezza dell'impianto, integrando il circuito di emergenza presente sul compressore con la nuova parte elettromeccanica che si intende attuare, e secondariamente per avere una interfaccia tra il controllore in cui è implementato il codice software di controllo, l'azionamento del motore e i sensori presenti sull'impianto.

Il controllore scelto per la gestione del sistema è costituito dal controllore NI PXI, di cui all'appendice B è riportata la scheda tecnica. Dato che il sistema è modulare, per questa applicazione viene allestito nella seguente configurazione:

- NI PXI 6123: scheda di input analogici e digitali.
- NI PXI 6602: scheda counter.
- NI PXI 6733: scheda di output analogici e digitali.

Con la scheda NI PXI 6123 verranno rilevate le misure relative a quattro sensori di pressione posti sul circuito. Con la scheda NI PXI 6602 verranno rilevati i segnali degli

encoder posti sul compressore, per la misura della velocità di rotazione del compressore, e sul motore che aziona il risonatore, per la misura della posizione dello stelo del risonatore. La scheda NI PXI 6733 è necessaria per fornire le uscite del sistema di controllo sia analogiche come ad esempio il segnale di comando dell'azionamento, sia digitali, come i comandi di servizio per il circuito di emergenza.

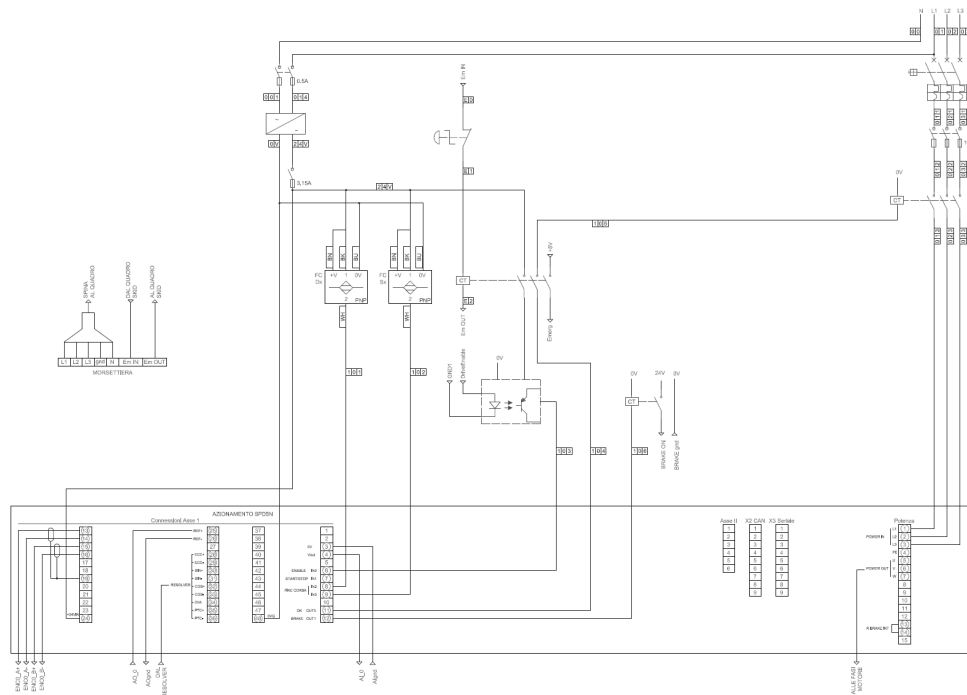


Figura 6.60: *Schema elettrico comando risonatore attivo: collegamento all'azionamento.*

Nella figura 6.60 è mostrato lo schema elettrico del circuito che consente l'interfacciamento dell'azionamento al controllore e al circuito elettrico del compressore. Nel circuito si possono notare la presenza del circuito di emergenza, denominato con *EmIn-EmOut*, la presenza dei sensori di prossimità posti in corrispondenza dei finecorsa, e i disaccoppiatori optoelettronici che consentono l'interfacciamento dei segnali digitali dal sistema di controllo al circuito di comando del motore.

Nella figura 6.61 è mostrato lo schema elettrico di collegamento al controllore: in particolare è rappresentato il controllore e i collegamenti dalle sue schede al circuito da controllare; inoltre è mostrata la catena di misura dei sensori di pressione che prevede la rilevazione della pressione con i sensori della serie 2200C1 e degli amplificatori di carica della serie 4751B1.



Il segnale deve essere opportunamente filtrato prima di essere acquisito, per evitare problemi di aliasing, e ciò viene effettuato inserendo nella catena di misura l'amplificatore NEXUS 2693-0S4. Accanto alla rilevazione dei segnali di pressione dinamica, si è ritenuto opportuno inserire anche due termocoppie per la rilevazione della temperatura del fluido a monte e a valle dell'impianto di test, con lo scopo di monitorare l'andamento di tale temperatura per verificare che essa rimanga entro certi limiti. Il segnale delle termocoppie deve essere condizionato per poter essere acquisito dal sistema di controllo e questo viene svolto da un opportuno circuito inserito all'interno del quadro di comando del risonatore. Anche i segnali degli encoder, essendo forniti in logica *open-collector*, necessitano di un circuito di condizionamento di tipo *line receiver*, svolto da un unico circuito integrato prodotto da Texas Instruments, modello AM26LS32ACN.

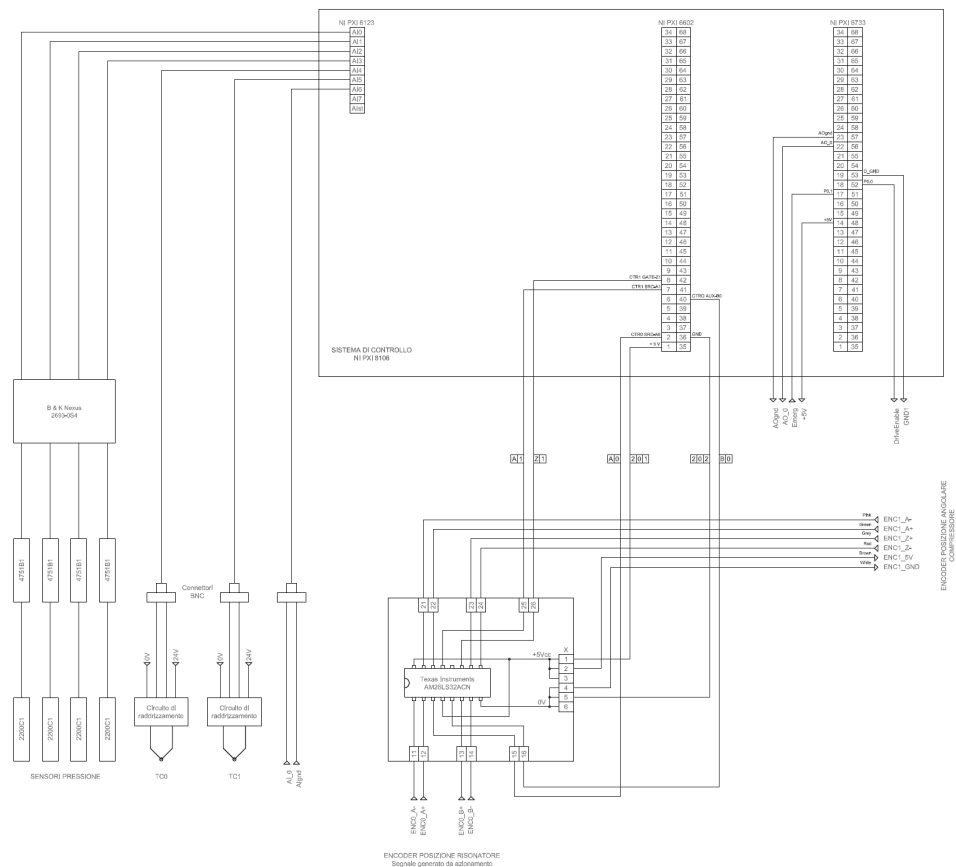


Figura 6.61: Schema elettrico comando risonatore attivo: collegamento al controllore.

### 6.3 Sistema di controllo

Al fine di realizzare un risonatore di Helmholtz attivo in grado di adattarsi automaticamente alle condizioni di funzionamento dell'impianto per l'attenuazione di una particolare oscillazione di pressione è necessario realizzare un sistema di controllo che, in funzione di alcune misure effettuate sul sistema, sia in grado di comandare l'attuatore che muove lo stantuffo del risonatore di Helmholtz a geometria variabile. L'energia necessaria al sistema di controllo è quella dovuta al solo posizionamento dello stantuffo, poiché l'abbattimento dell'oscillazione di pressione viene effettuato interamente dal dispositivo risonante. L'abbattimento è realizzato da un dispositivo passivo, le cui caratteristiche possono essere modificate dal sistema di controllo che adatta la geometria, quindi la frequenza di risonanza  $f_R$  del risonatore, in funzione della frequenza dell'oscillazione di pressione che si intende abbattere. La figura 6.62 mostra una foto dell'intero sistema di abbattimento delle oscillazioni di pressione costituito dal risonatore di Helmholtz a geometria variabile (par. 6.2.1), dal sistema di trasmissione (par. 6.2.1), dal motore brushless (par. 6.2.2) e dal quadro elettrico di controllo (par. 6.2.3).

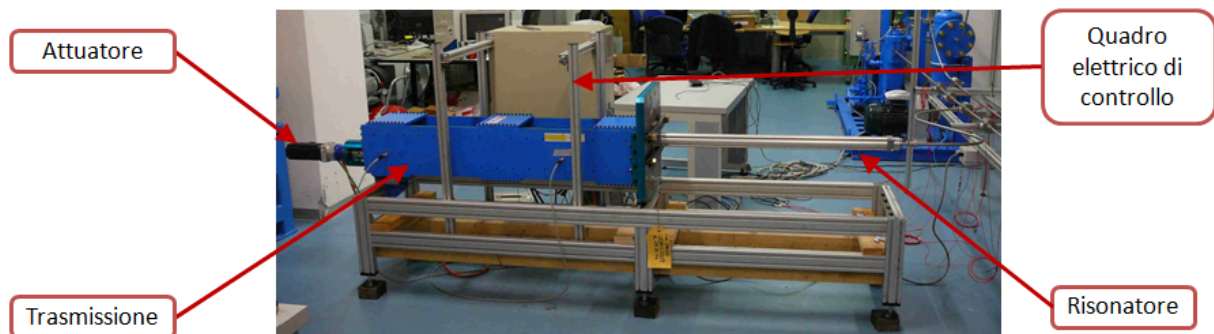


Figura 6.62: *Sistema di abbattimento indiretto delle oscillazioni di pressione.*

Per realizzare un buon sistema di controllo occorre definire in modo adeguato tutte le operazioni che devono essere svolte dallo stesso, a tal fine la figura 6.63 riporta lo schema concettuale su cui si basa l'algoritmo implementato nel sistema di controllo per realizzare l'abbattimento desiderato delle oscillazioni di pressione.

Nel seguente elenco vengono riassunte le principali operazioni svolte dal sistema di controllo:

1. L'utente seleziona dal quadro elettrico del compressore la velocità del motore al fine di regolare la portata di aria nel condotto di mandata del compressore.

2. Il sistema di controllo rileva la velocità del motore del compressore e determina la frequenza corrispondente dell'oscillazione di pressione: dalle analisi effettuate sul compressore industriale (paragrafi 5.1.4 e 5.2.2) è emerso infatti che le pulsazioni di pressione si hanno in corrispondenza dei multipli fondamentali della velocità di rotazione del compressore.
3. A partire dalla misura della frequenza dell'oscillazione di pressione un algoritmo determina il *setpoint* di posizione da fornire allo stantuffo del risonatore (par. 6.3.2).
4. Noto il *setpoint* di posizione, un anello di controllo in *feedback* sulla posizione dello stantuffo garantisce l'esatto posizionamento di quest'ultimo e quindi, la geometria del risonatore di Helmholtz viene modificata in modo da tarare la sua frequenza di risonanza  $f_R$  a quella dell'oscillazione di pressione presente nel condotto principale.
5. Il sistema di controllo fornisce il *setpoint* di posizione dello stantuffo all'azionamento del motore che a sua volta fornisce la potenza necessaria a muovere il motore del risonatore.
6. Il segnale dell'encoder del motore del risonatore viene inviato all'anello di controllo che verifica il raggiungimento del *setpoint* di posizione secondo un regolatore *PID*.
7. La trasmissione, realizzata in precedenza, è dotata di due sensori di prossimità che garantiscono il funzionamento in sicurezza del risonatore. Durante il funzionamento del sistema di controllo, l'azionamento del motore verifica l'eventuale raggiungimento di uno dei due sensori al fine di togliere potenza al motore ed evitare impatti che possono comportare un danneggiamento della trasmissione meccanica. Gli stessi sensori vengono utilizzati nella prima fase di accensione del sistema di controllo per favorire l'azzeramento della posizione dello stantuffo del risonatore. Se l'energia del sistema in movimento è elevata e vengono superati abbondantemente i finecorsa elettrici, dei finecorsa meccanici (fig. 6.59) bloccano la corsa dello stantuffo al fine di ridurre i danni provocati da un eventuale malfunzionamento del sistema di controllo.

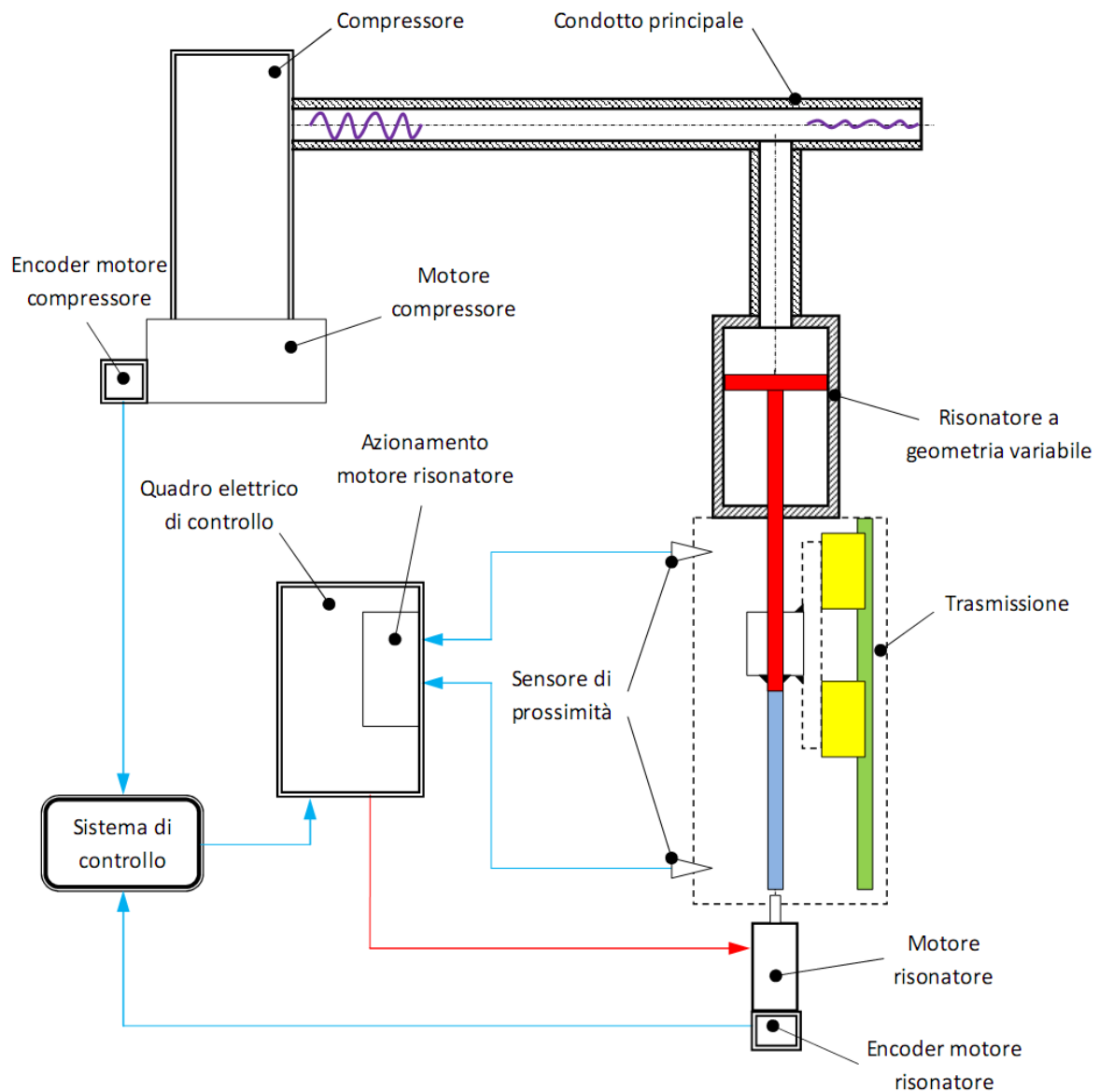


Figura 6.63: Schema concettuale del sistema di controllo per l'abbattimento delle oscillazioni di pressione con il risonatore di Helmholtz a geometria variabile.

### 6.3.1 Schema di controllo

Secondo le funzionalità del sistema di controllo, descritte nel paragrafo 6.3, lo schema di controllo può essere suddiviso, come illustrato in figura 6.64, in due parti fondamentali:

- La prima parte fondamentale dello schema di controllo, evidenziata in rosso nella figura, è quella che garantisce il corretto posizionamento dello stantuffo del

risonatore di Helmholtz a geometria variabile. Il controllo della posizione dello stantuffo, quindi del raggiungimento del setpoint desiderato, è basato su un algoritmo che realizza un classico regolatore per il controllo del moto.

- La seconda parte dello schema di controllo, evidenziata in giallo, implementa un algoritmo che determina il setpoint da fornire allo stantuffo al fine di abbattere una certa componente in frequenza delle oscillazioni di pressione presenti nel condotto principale dell'impianto di test. Il compito di questa seconda parte dello schema di controllo è quello di determinare il setpoint di posizione e fornirlo al primo algoritmo che determina l'azione di comando da fornire al motore del risonatore.

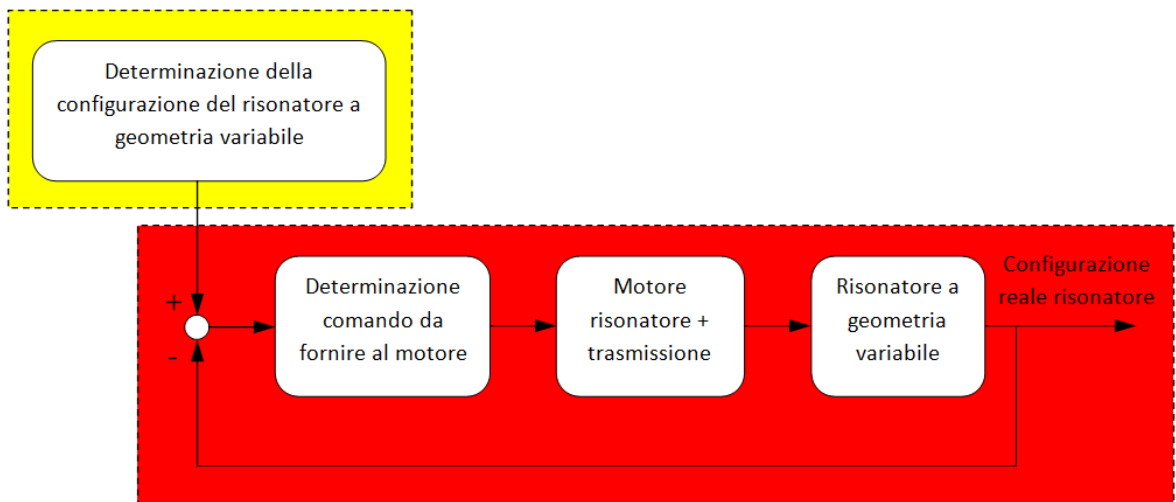


Figura 6.64: *Schema di controllo del risonatore a geometria variabile.*

L'implementazione del codice di controllo avviene secondo lo schema a blocchi riportato in figura 6.65 che realizza tutte le funzionalità del sistema di controllo esposte nel paragrafo 6.3; nella figura vengono in particolare evidenziati i blocchi relativi alle due parti fondamentali del sistema di controllo i cui relativi algoritmi verranno discussi nel paragrafo 6.3.2. Per il codice di controllo, realizzato in ambiente *ANSI C*, si faccia riferimento all'appendice G. Le modalità di implementazione e di utilizzo del codice di controllo sviluppato sono le medesime illustrate per il controllo attivo del rumore (par. 3.2.2).

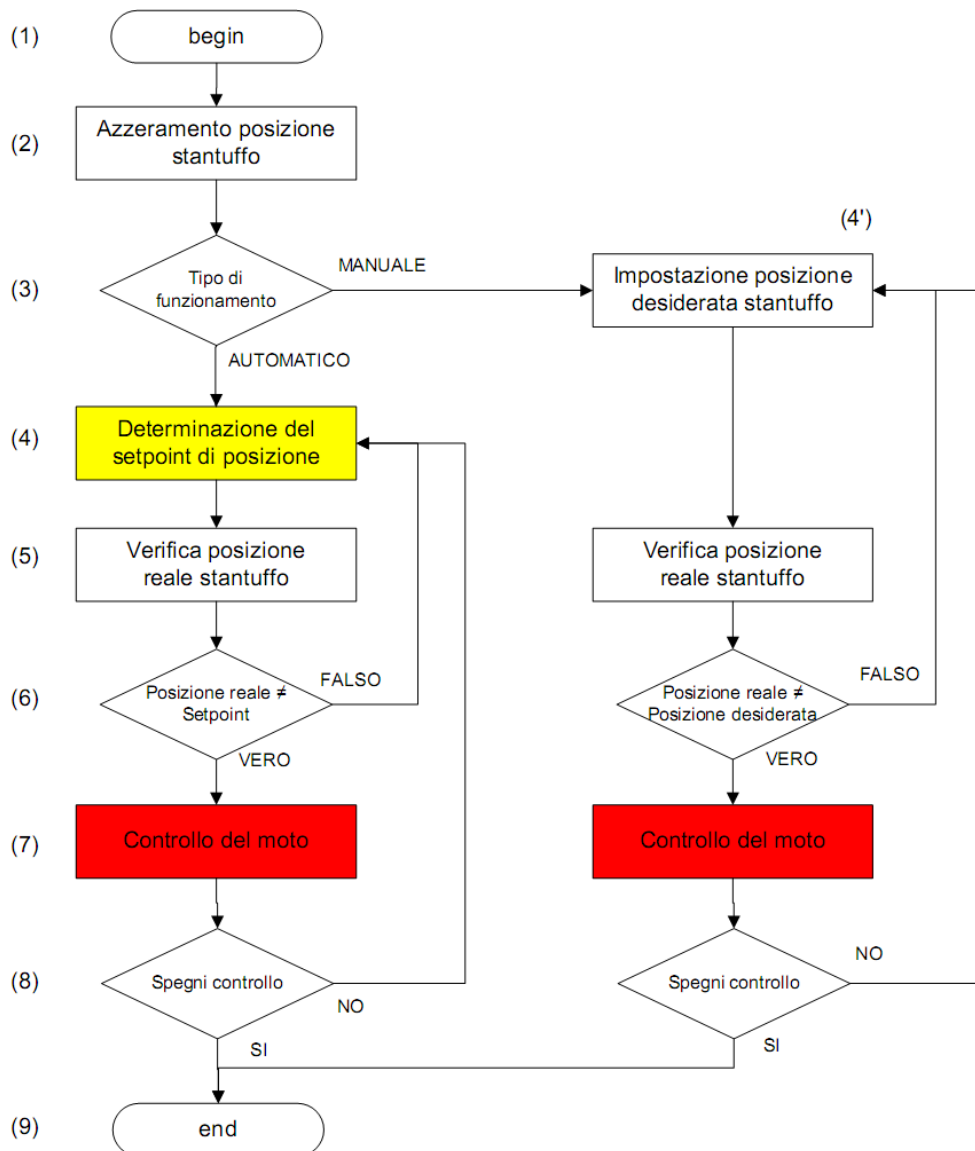


Figura 6.65: Schema a blocchi del sistema di controllo per l'abbattimento delle oscillazioni di pressione mediante il risonatore di Helmholtz a geometria variabile.

L'interfaccia utente LabView, mostrata in figura 6.69, è stata creata allo scopo di consentire all'utente il comando in modalità manuale o automatica del risonatore a geometria variabile (fig. 6.65). Il funzionamento in modalità manuale consente di effettuare le operazioni necessarie per la caratterizzazione del risonatore, come la determinazione della frequenza propria  $f_R$  del risonatore in funzione dell'effettiva posizione dello stantuffo. Il funzionamento automatico sfrutta il sensore per rilevare la velocità del motore e quindi la frequenza dell'oscillazione di pressione, utilizzando poi il codice

di controllo del moto per muovere automaticamente lo stantuffo nella posizione che consente di ridurre tale oscillazione. L'interfaccia utente consente inoltre di tenere sotto controllo eventuali condizioni di *fault* che potrebbero verificarsi nell'esecuzione del sistema di controllo e procedere con l'arresto sicuro della macchina.

L'implementazione reale dello schema di controllo prevede la caratterizzazione dell'intero sistema in termini di masse, inerzie, e frequenze di risonanza del risonatore a geometria variabile al fine di realizzare un modello che consente di determinare gli algoritmi e le costanti di controllo.

### 6.3.2 Algoritmo di controllo

Come mostrato nel paragrafo precedente il sistema di controllo è costituito principalmente da due algoritmi: il primo, denominato "Controllo del moto", consente di posizionare lo stantuffo del risonatore in corrispondenza di una posizione desiderata, mentre il secondo algoritmo, denominato "Determinazione del setpoint di posizione", determina la posizione desiderata in funzione delle condizioni di funzionamento reali del compressore.

#### Controllo del moto

L'algoritmo di controllo del moto è basato su un regolatore *PID* (regolatore Proporzionale Derivativo Integrale): noto il setpoint di posizione, determinato dall'algoritmo "Determinazione del setpoint di posizione" che verrà mostrato in seguito, l'algoritmo di controllo del moto determina l'azione di comando da fornire al motore del risonatore al fine di portare nella posizione desiderata lo stantuffo del risonatore. A partire dalla scrittura dell'equazione di moto per il modello semplificato del sistema rappresentato in figura 6.66, vengono determinati i guadagni da assegnare al regolatore [34, 35]. Nella tabella 6.8 si riportano le caratteristiche principali del sistema.

Mediante la scrittura di un bilancio energetico delle forme di energia associate al funzionamento del modello rappresentato in figura 6.66, si perviene all'equazione di moto del sistema:

$$m^* \ddot{x} = \tau_t C_{mot}, \quad (6.18)$$

dove  $\tau_t$  è il rapporto di trasmissione totale ottenuto come:

$$\tau_t = \frac{2\pi}{\tau_{rid} p_{vite}}, \quad (6.19)$$

e  $m^*$  è la massa generalizzata del sistema:

$$m^* = m_{tot} + (J_{mot} + J_{rid})\tau_t^2 + (J_{vite} + J_{tex})\left(\frac{2\pi}{p_{vite}}\right)^2. \quad (6.20)$$

Parametro	Valore	Unità di misura	Descrizione
$m_{tot}$	35.3	kg	Massa mobile complessiva
$J_{vite}$	0.0059	kgm <sup>2</sup>	Inerzia vite a ricircolo
$J_{mot}$	0.183 10 <sup>-3</sup>	kgm <sup>2</sup>	Inerzia vite a ricircolo
$J_{rid}$	0.180 10 <sup>-4</sup>	kgm <sup>2</sup>	Inerzia riduttore
$J_{tex}$	1.090 10 <sup>-4</sup>	kgm <sup>2</sup>	Inerzia giunto elastico
$k_{mot}$	0.96	Nm/A	Costante di coppia motore
$\tau_{rid}$	1/28		Rapporto di trasmissione riduttore
$p_{vite}$	10.0	mm/giro	Passo vite a ricircolo

Tabella 6.8: Parametri fisici del risonatore a geometria variabile.

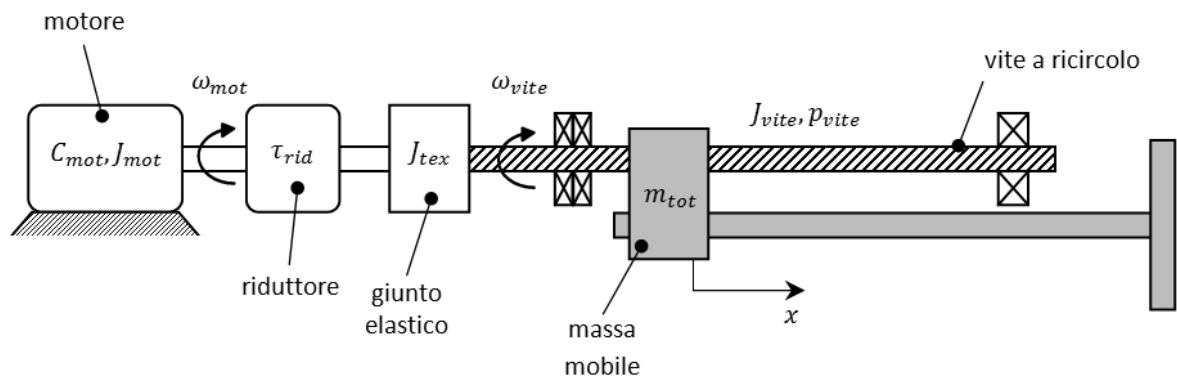


Figura 6.66: Modello semplificato del risonatore di Helmholtz a geometria variabile con trasmissione.

Nota l'equazione di moto del sistema e applicando a questa la trasformata di Laplace si perviene all'equazione 6.21, che esprime la funzione di trasferimento  $M(s)$  del sistema meccanico tra l'azione di comando  $C_{mot}$  del motore e la risposta in posizione  $X(s)$  dello stantuffo, ossia di quanto si sposta lo stantuffo a seguito dell'applicazione di una coppia motrice da parte del motore del risonatore.

$$M(s) = \frac{X(s)}{C_{mot}(s)} = \frac{\tau_t}{m^*s^2}. \quad (6.21)$$



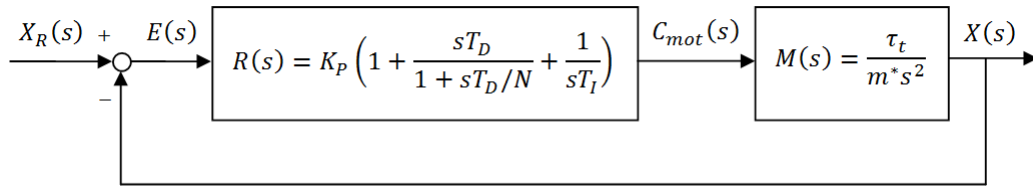


Figura 6.67: Schema di controllo con regolatore PID.

L'azione di controllo, utilizzando un regolatore *PID* (fig. 6.67), può essere scritta come [35]:

$$C_{mot}(s) = R(s)E(s) = K_P \left( 1 + \frac{sT_D}{1 + sT_D/N} + \frac{1}{sT_I} \right) (X_R(s) - X(s)), \quad (6.22)$$

dove:

- $R(s)$  è la funzione di trasferimento del regolatore *PID* [35].
- $E(s) = X_R(s) - X(s)$  è l'errore in posizione ottenuto come differenza tra il valore del setpoint desiderato e la posizione reale del sistema.
- $X_R(s)$  è il setpoint di posizione, ossia la posizione desiderata che si vuole far assumere allo stantuffo del risonatore. Il setpoint  $X_R(s)$ , a seconda del funzionamento in manuale o automatico del sistema di controllo, può essere o la posizione dello stantuffo scelta dall'utente o la posizione determinata dall'algoritmo "Determinazione del setpoint di posizione".
- $K_P$  è il guadagno proporzionale all'errore di posizione  $E(s)$ .
- $T_D$  è il tempo derivativo ottenuto come il rapporto tra il guadagno derivativo  $K_D$  e quello proporzionale  $K_P$ .
- $T_I$  è il tempo integrale ottenuto come il rapporto tra il guadagno proporzionale  $K_P$  e il guadagno dell'azione integrale  $K_I$ .
- $N$  è la posizione del polo di un filtro passa-basso del I ordine per limitare il livello di amplificazione del rumore associato all'azione di derivazione del segnale di errore  $E(s)$ .

A partire dall'analisi dei poli del sistema non controllato  $M(s)$  i guadagni del regolatore vengono tarati in modo tale da garantire al sistema controllato un'adeguata banda passante e un buon comportamento a regime ( $E(s) \simeq 0$ ) e buone prestazioni dinamiche in assenza di sovraelongazioni sulla posizione  $X(s)$  dello stantuffo. Il sistema non controllato è un sistema labile costituito solamente da una massa soggetta ad una forza e i poli di questo sistema sono due poli reali nulli e coincidenti. Per garantire un'adeguata stabilità del sistema si introduce il regolatore  $PID$  analizzando il luogo delle radici riportato in figura 6.68.

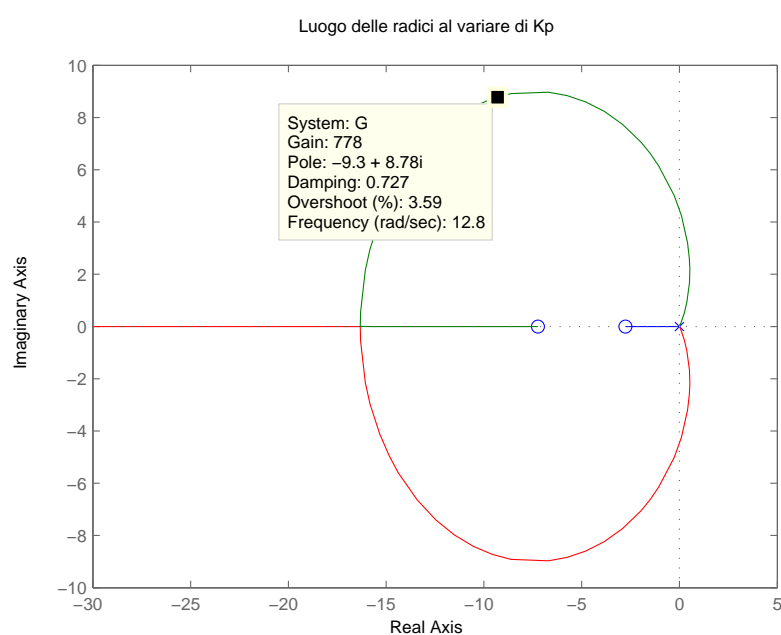


Figura 6.68: *Luogo delle radici del sistema controllato con regolatore PID al variare del guadagno proporzionale  $K_P$ .*

Dall'analisi del luogo della radici si sceglie il valore di  $K_P = 777$  tale da massimizzare lo smorzamento adimensionale<sup>1</sup> del sistema, che si ha in corrispondenza di  $h = 0.707$  e gli altri guadagni del regolatore tali da garantire un adeguato margine di fase e di guadagno. Nella tabella 6.9 vengono riportati i guadagni utilizzati per il regolatore  $PID$  progettato.

<sup>1</sup>Lo smorzamento adimensionale di un sistema ad un grado di libertà, costituito da una massa  $m$  connessa a terra da una molla di rigidità  $k$  e uno smorzatore di coefficiente  $r$ , viene definito come  $h = \frac{r}{r_c}$ , dove  $r_c = 2m\omega_0$  e  $\omega_0 = \sqrt{\frac{k}{m}}$ . Con  $h = 0.707$  la sovraelongazione della risposta del sistema è minima, e si garantisce anche un adeguato tempo di risposta del sistema.

Parametro	Valore
$K_P$	777.0
$T_D$	0.1
$T_I$	0.5
$N$	125.7

Tabella 6.9: *Parametri del regolatore PID per il controllo del moto del risonatore di Helmholtz a geometria variabile.*

### Determinazione del setpoint di posizione

La frequenza dell'oscillazione di pressione viene ottenuta indirettamente mediante un encoder che rileva la velocità di rotazione dell'albero del compressore. La prima fase per lo sviluppo dell'algoritmo di "Determinazione del setpoint di posizione" prevede la determinazione della frequenza di risonanza del risonatore in funzione della posizione dello stantuffo. A tal fine, mediante lo sviluppo di un opportuno codice di controllo e il pannello utente (fig. 6.69), con la procedura di funzionamento in manuale l'utente è in grado di scegliere a piacere la posizione dello stantuffo e valutare la frequenza di risonanza del risonatore con lo stantuffo nella posizione desiderata. Tale procedura viene effettuata secondo le seguenti fasi:

1. Viene imposta grazie all'inverter la velocità di rotazione del motore del compressore  $RPM_{motore}$ . La velocità di rotazione viene fatta variare nel range di funzionamento del compressore con incrementi di  $50 RPM$ , da  $450 RPM$  a  $1600 RPM$ .
2. Per ogni velocità di rotazione impostata mediante l'inverter, grazie al pannello utente è possibile variare la posizione dello stantuffo e valutare in tempo reale lo spettro del segnale di pressione in un punto del condotto principale. Quando il picco in corrispondenza della componente in frequenza  $2x$  giro (multiplo doppio della velocità di rotazione dell'albero del compressore) raggiunge il minimo, si ritiene che lo stantuffo abbia raggiunto una posizione tale da rendere ottima la performance di abbattimento del risonatore per quella frequenza di rotazione. In corrispondenza di questa posizione dello stantuffo la frequenza di risonanza del risonatore è pari a quella del picco che viene ridotto.
3. Per ogni variazione della velocità di rotazione del motore, vengono salvati la posizione dello stantuffo e la frequenza del picco che viene ridotto. Una volta coperto

tutto il range di velocità di funzionamento del compressore, viene mappato l'andamento della frequenza di risonanza del risonatore in funzione della posizione dello stantuffo e questo andamento viene confrontato con quello atteso dalla relazione analitica 4.110 del risonatore di Helmholtz:

$$f_R = \frac{c}{2\pi} \frac{D_C}{D_R} \sqrt{\frac{1}{(L_{Ceff} + \frac{16}{3\pi} D_C) L_R}}. \quad (6.23)$$

Quest'ultima relazione mette in evidenza l'utilizzo della correzione (relazione 4.98) sulla lunghezza del collo per considerare eventuali fenomeni non lineari. Nella figura 6.70 vengono riportati i risultati ottenuti mediante questa procedura di caratterizzazione del risonatore di Helmholtz a geometria variabile.

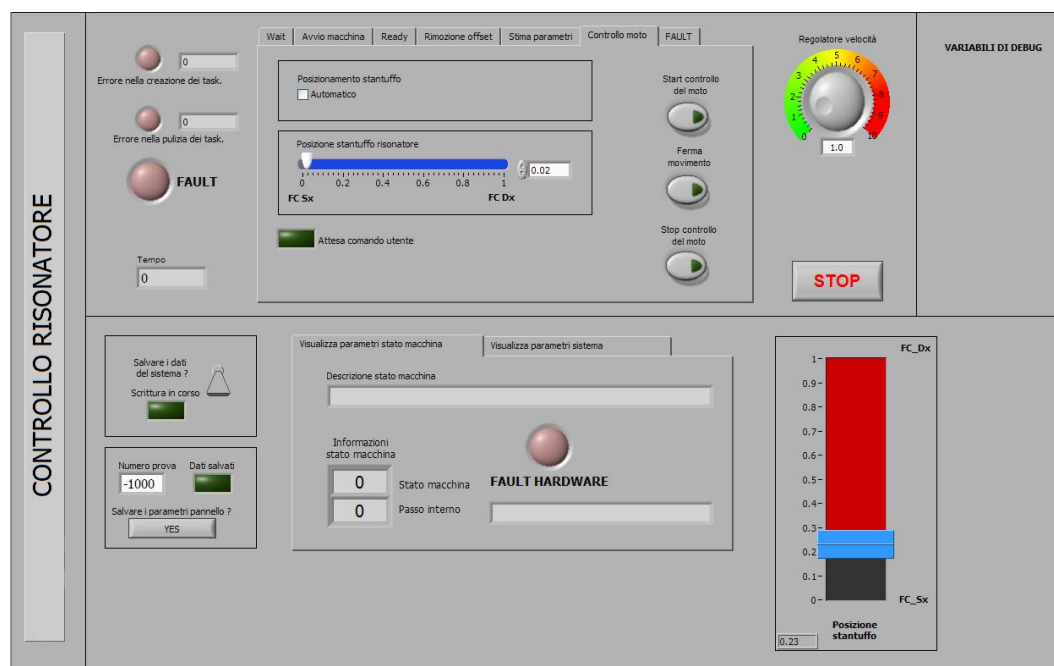


Figura 6.69: Pannello utente per il controllo del sistema di controllo del risonatore di Helmholtz a geometria variabile.

Analizzando la figura 6.70 si nota che la frequenza di risonanza ottenuta sperimentalmente ha un andamento molto simile a quello atteso dalla relazione analitica ma che non si sovrappone perfettamente a quest'ultima. Questa leggera differenza tra i risultati può essere dovuta alla presenza di un volume, tra la camera di risonanza e il collo del risonatore, trascurato nella determinazione della frequenza  $f_R$  con la relazione analitica e in particolare nel calcolo del volume complessivo (o della lunghezza  $L_R$ ) della

camera di risonanza. Inoltre, nella relazione analitica, si è tenuto conto della correzione sulla lunghezza del collo che probabilmente si discosta da quella effettivamente corretta per questo impianto. Al fine di compensare questi effetti sulla relazione analitica del risonatore è stato sufficiente aggiungere a quest'ultima una correzione sulla lunghezza della camera di risonanza pari a  $L_{Rc} = 35 \text{ mm}$ . Considerando la correzione sulla lunghezza della camera di risonanza la curva relativa alla  $f_R$  teorica in figura 6.71 trasla verso sinistra fino a sovrapporsi ai dati rilevati sperimentalmente.

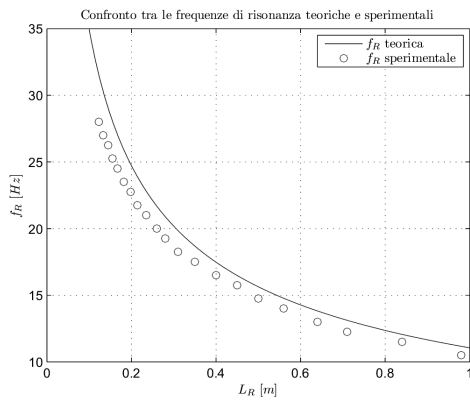


Figura 6.70: *Caratterizzazione sperimentale del risonatore di Helmholtz a geometria variabile.*

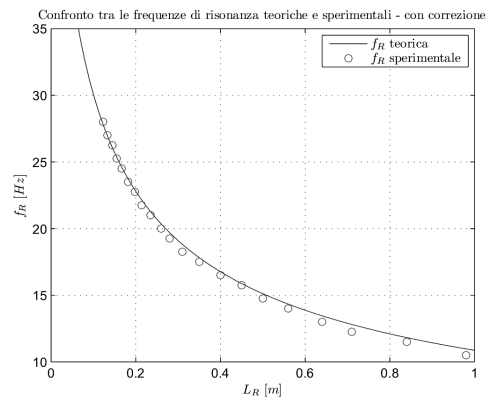


Figura 6.71: *Caratterizzazione sperimentale del risonatore di Helmholtz a geometria variabile, con correzione.*

Nota quindi la velocità di rotazione dell'albero del compressore (o la velocità angolare del motore), ovvero la frequenza di risonanza della componente  $2x$  giro da attenuare, la posizione dello stantuffo  $L_R$ , ossia il setpoint  $X_R$  da assegnare al regolatore per il controllo del moto, viene determinato dall'algoritmo di controllo mediante la seguente relazione:

$$L_R = \left( \frac{c}{2\pi} \frac{D_C}{D_R} \right) \frac{(L_{Ceff} + \frac{16}{3\pi} D_C)}{f_R^{2x}} - L_{Rc} = X_R, \quad (6.24)$$

dove  $f_R^{2x}$  è data da:

$$f_R^{2x} = \frac{\kappa_{2x} RPM_{motore}}{60} = \frac{RPM_{motore}}{30}. \quad (6.25)$$

Il valore di setpoint  $X_R = L_R$  determinato da questo algoritmo viene fornito all'algoritmo di controllo del moto che provvede a determinare la coppia motrice  $C_{mot}$  tale da portare lo stantuffo nella posizione desiderata.

## 6.4 Risultati sperimentali

Le analisi delle performance del risonatore di Helmholtz a geometria variabile sono state effettuate collegando il risonatore nel punto PT3 (fig. 6.47), valutando le oscillazioni di pressione in punti di misura a valle e a monte di quello di collegamento del risonatore e confrontando le ampiezze di queste oscillazioni con quelle presenti nell'impianto in assenza di dispositivi risonanti.

Le analisi vengono effettuate variando la velocità di rotazione del motore nel range di funzionamento del compressore  $450 \leq RPM_{motore} \leq 1600 RPM$  e misurando di volta in volta la storia temporale delle oscillazioni di pressione nei punti di misura di interesse.

Il risonatore di Helmholtz viene comandato in modo automatico dal sistema di controllo che rileva la velocità di rotazione del motore e comanda il motore elettrico per il posizionamento dello stantuffo del cilindro pneumatico. La posizione dello stantuffo viene determinata dall'algoritmo di controllo in modo da regolare il volume della camera di risonanza e far coincidere la frequenza di risonanza del risonatore  $f_R$  alla componente 2x giro dell'oscillazione di pressione corrispondente alla velocità impostata sul motore. Utilizzando il sistema di controllo in modalità automatica viene posizionato lo stantuffo del cilindro al fine di realizzare, in tutto il range di frequenze di funzionamento del compressore, il massimo abbattimento possibile: con tale procedura vengono effettuate le analisi mostrate nei grafici nelle figure dalla 6.72 alla 6.79.

Dalle analisi sui punti di misura a valle di quello di inserimento del risonatore (figure dalla 6.72 alla 6.77) si notano le elevate performance di abbattimento di questo risonatore in tutto il range di frequenze di funzionamento dell'impianto: in particolare, data la particolare configurazione costruttiva di questo risonatore, è stato possibile incrementare il range di frequenze di abbattimento del risonatore potendo posizionare lo stantuffo del cilindro pneumatico fino alla posizione di  $L_R = 100.0 mm$  che permette di abbattere oscillazioni di pressione fino alla frequenza di  $34.71 Hz$ .

Elevati abbattimenti possono essere ottenuti anche nei punti a monte di quello di inserimento del risonatore, come mostrato nelle figure 6.78 e 6.79, relative al punto di misura PT1: questo risultato era atteso considerando i risultati ottenuti nelle analisi delle performance dei risonatori di Helmholtz a geometria fissa.

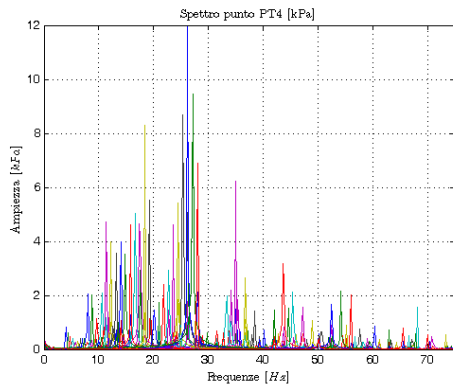


Figura 6.72: Sovrapposizione spettri oscillazioni di pressione: SS-PT4.

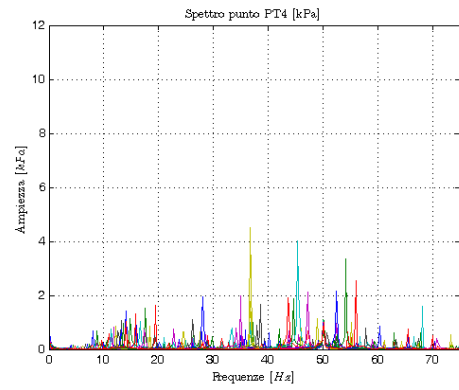


Figura 6.73: Sovrapposizione spettri oscillazioni di pressione:  $R_{VAR}^H$ -PT3-PT4.

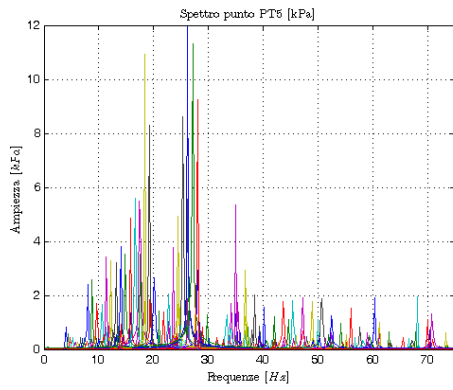


Figura 6.74: Sovrapposizione spettri oscillazioni di pressione: SS-PT5.

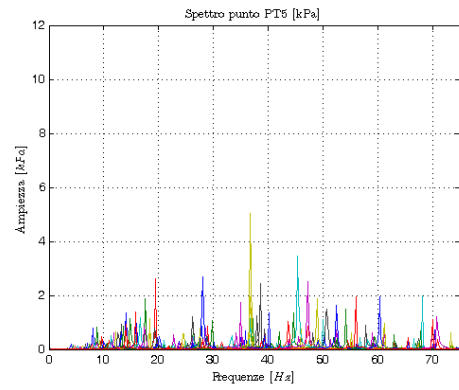


Figura 6.75: Sovrapposizione spettri oscillazioni di pressione:  $R_{VAR}^H$ -PT3-PT5.

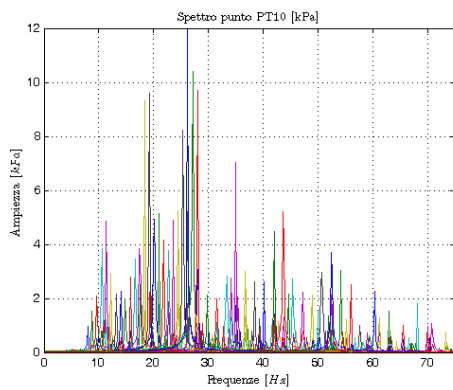


Figura 6.76: Sovrapposizione spettri oscillazioni di pressione: SS-PT10.

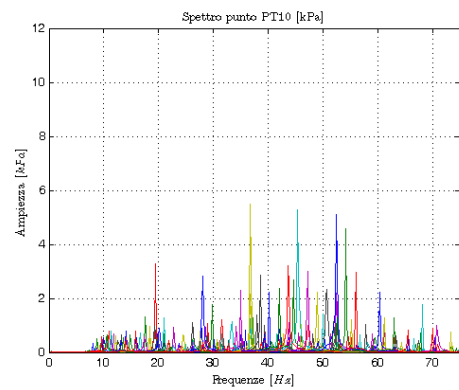


Figura 6.77: Sovrapposizione spettri oscillazioni di pressione:  $R_{VAR}^H$ -PT3-PT10.

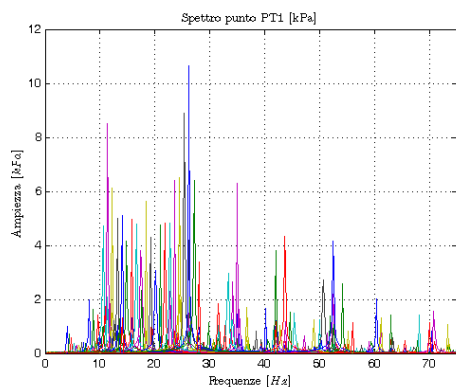


Figura 6.78: Sovrapposizione spettri oscillazioni di pressione: SS-PT4.

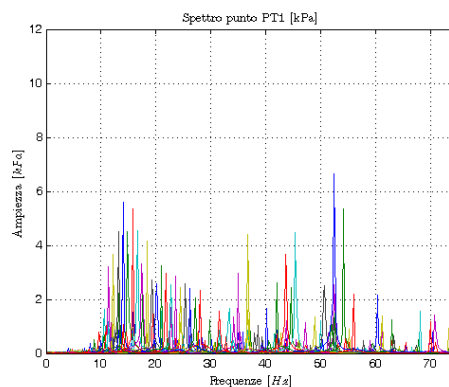


Figura 6.79: Sovrapposizione spettri oscillazioni di pressione:  $R_{\text{VAR}}^H$ -PT3-PT1.

Nelle figure 6.80 e 6.81 vengono mostrati i grafici riepilogativi degli abbattimenti ottenuti nei punti di misura PT4 e PT10.

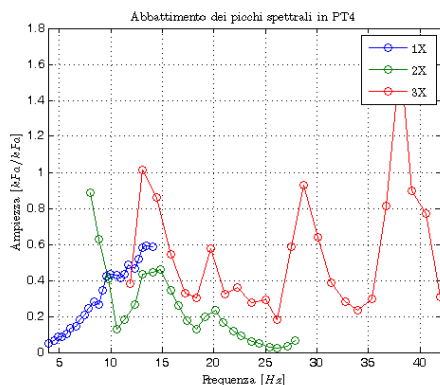


Figura 6.80: Grafico riepilogativo abbattimento risonatore di Helmholtz a geometria variabile nel punto di misura PT4.

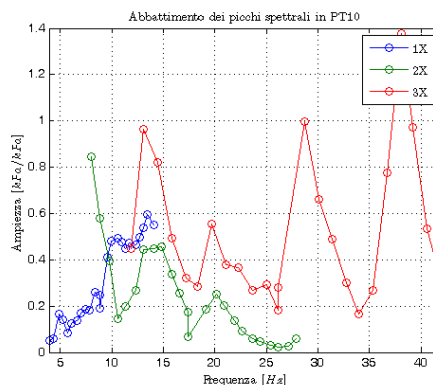


Figura 6.81: Grafico riepilogativo abbattimento risonatore di Helmholtz a geometria variabile nel punto di misura PT10.

Questi grafici vengono ottenuti rapportando le ampiezze dei picchi spettrali con e senza risonatore delle frequenze corrispondenti ai primi tre multipli interi della velocità di rotazione del compressore. Essi evidenziano le capacità di questo dispositivo di attenuare non solo la componente in frequenza delle oscillazione per la quale è stato progettato, ma anche di ottenere elevati abbattimenti per le componenti 1x giro e 3x giro della velocità di rotazione dell'albero del compressore, sebbene queste componenti non siano direttamente controllate. Ovviamente le maggiori performance si hanno in



corrispondenza della frequenza  $f_R$  di progetto del risonatore che si ricorda esser stato tarato sulla componente in frequenza  $2x$  giro della velocità di rotazione del compressore.

Il tempo di ricerca della posizione dello stantuffo da parte dell'algoritmo di controllo è estremamente ridotto grazie alle elevate caratteristiche prestazionali del sistema di controllo. Sperimentalmente si osserva che il sistema di controllo determina e garantisce il posizionamento dello stantuffo del cilindro nella posizione desiderata ancor prima che le oscillazioni di pressione, alla frequenza corrispondente alle condizioni di funzionamento impostate sul motore, raggiungano le condizioni di regime. Ciò permette quindi di mantenere sempre ridotte le massime ampiezze delle oscillazioni di pressione che si avrebbero in assenza del dispositivo risonante collegato all'impianto.



# CONCLUSIONI

In questo lavoro di tesi sono stati sviluppati due apparati sperimentali per la valutazione delle performance dei sistemi di controllo per l'abbattimento delle oscillazioni di pressione. Lo sviluppo di questi apparati, basati sull'utilizzo di sistemi di controllo attivi e passivi, è stato preceduto da una dettagliata analisi teorica e numerica della propagazione monodimensionale delle onde e dei meccanismi che regolano l'abbattimento delle oscillazioni di pressione. Queste analisi hanno consentito di realizzare degli apparati sperimentali che hanno dimostrato delle elevate performance di abbattimento in un ampio campo di frequenze. Il primo apparato, progettato per il test dei sistemi di controllo attivo del rumore, ha consentito di ottenere abbattimenti all'interno del campo delle frequenze del suono udibili e in grado di essere generate da un comune altoparlante acustico. Questo apparato ha consentito lo studio degli algoritmi di controllo attivo del rumore basati sulla minimizzazione di un funzionale, definito in base alle caratteristiche del rumore. Le prestazioni di tale dispositivo hanno portato alla possibilità di abbattere rumori stazionari multi-frequenza con un numero di iterazioni ridotto ed in un range di frequenze da  $250\text{ Hz}$  a  $1300\text{ Hz}$ . All'interno del campo di frequenze controllato l'abbattimento medio ottenuto è pari a  $12\text{ dB}$ , valutato sulla singola armonica. L'utilizzo di questa metodologia non sempre è applicabile, poiché in alcuni casi le prestazioni ottenute sono inferiori a quelle ottenibili da altri metodi. In impianti dove vi è trasporto di massa o pressioni elevate rende complicato l'utilizzo di queste tecnologie di abbattimento. Per far fronte a questi limiti, il secondo apparato utilizza dispositivi passivi di tipo risonante che, pur essendo generalmente progettati per l'abbattimento di rumore con frequenze piuttosto elevate, sono stati realizzati con particolari configurazioni geometriche che consentono di abbattere oscillazioni di pressione al di sotto del campo di frequenze in cui vengono generalmente utilizzati. Il progetto di queste particolari configurazioni geometriche è stato accompagnato dallo studio teo-

rico e numerico delle equazioni che governano la dinamica di questi dispositivi quando vengono collegati ad un condotto di un impianto ove sono presenti delle oscillazioni di pressione. Queste analisi, affiancate dalla verifica sperimentale del comportamento di questi sistemi, hanno consentito di stabilire i criteri di buon progetto. Tutto ciò, unito all'interesse personale per gli argomenti trattati, ha dato vita alla progettazione e allo sviluppo sperimentale di un risonatore di Helmholtz a geometria variabile: il range di frequenze di funzionamento di questo dispositivo è ben al di sotto di quello ottenibile da un comune risonatore di Helmholtz, pervenendo di fatto ad abbattimenti di oscillazioni di pressione con frequenze minime prossime a  $6 \text{ Hz}$  e massime superiori a  $36 \text{ Hz}$ .

Naturalmente, prima di intraprendere lo sviluppo degli apparati sperimentali, nel **Primo Capitolo** viene presentata la teoria che governa la propagazione monodimensionale delle onde ed i meccanismi che regolano l'abbattimento. Al fine di delineare efficacemente i criteri di abbattimento e presentare le tecniche che verranno utilizzate negli apparati sperimentali realizzati, vengono classificate le metodologie per l'abbattimento del rumore con particolare riferimento ai dispositivi passivi e attivi, noti in letteratura, evidenziandone pregi e difetti.

Il **Secondo Capitolo** ha visto la definizione delle tecniche e la presentazione degli elementi costituenti un sistema di controllo attivo del rumore, mettendo in evidenza le loro caratteristiche in funzione delle performance del sistema di controllo che si intende progettare. In seguito sono stati presentati gli algoritmi di controllo che saranno implementati nell'apparato sperimentale per il controllo attivo del rumore. La definizione del sistema di controllo e degli algoritmi è stata effettuata previa applicazione della teoria di propagazione delle onde, presentata nel primo capitolo, ad un condotto dove vi è una propagazione monodimensionale delle onde. La geometria dell'ambiente sonoro scelta nello sviluppo di questo modello matematico ha portato alla definizione delle caratteristiche geometriche dell'apparato sperimentale.

Grazie allo studio dell'ambiente di propagazione delle onde di pressione, nel **Terzo Capitolo** viene progettato l'apparato sperimentale per il test degli algoritmi di controllo attivo del rumore implementati nel secondo capitolo. Il progetto ha riguardato il dimensionamento dell'ambiente acustico, la scelta della geometria del condotto che consente di realizzare il controllo attivo, e infine la scelta dei sensori, degli attuatori e della piattaforma hardware e software per l'esecuzione dei codici di controllo. Una volta eseguita la progettazione e la realizzazione dell'apparato sperimentale si è proceduto ad effettuare la caratterizzazione dell'impianto, e a valutare e confrontare le performan-

ce di abbattimento del rumore ottenibili mediante gli algoritmi presentati nel secondo capitolo.

Il **Quarto Capitolo** sposta l'attenzione sul secondo apparato sperimentale. Lo studio di un apparato per il controllo indiretto dell'abbattimento di oscillazioni di pressione nasce dalla necessità di coprire il campo di frequenze dove le tecniche di controllo attivo, mostrate nel primo apparato sperimentale, possono risultare inadeguate. Lo sviluppo questo apparato avviene considerando l'opportunità di realizzare degli abbattimenti delle oscillazioni di pressione sfruttando le tecniche di abbattimento passivo basate sull'utilizzo di dispositivi risonanti di tipo Helmholtz e side-branch. La teoria di propagazione vista nel primo capitolo viene rielaborata al fine di sviluppare un metodo sistematico per la scrittura delle equazioni che governano la propagazione delle onde in condotti in cui vi sono dei dispositivi risonanti ad essi collegati. Il modello analitico realizzato è stato quindi impiegato per l'analisi del comportamento dinamico e dell'abbattimento delle oscillazioni di pressione ottenibili con le due tipologie di dispositivi. L'analisi effettuata al variare delle caratteristiche geometriche dei risonatori consente di indagare i range di frequenze di funzionamento e i criteri di buona progettazione. I risultati così ottenuti sono stati successivamente validati grazie ad una analisi numerica, mediante lo studio del sistema con una modellazione ad elementi finiti.

Al fine di realizzare un apparato sperimentale che più si avvicini al comportamento reale di un impianto industriale, nel **Quinto Capitolo** vengono presentati i dati ottenuti da una campagna di misura sperimentale relativa all'andamento delle oscillazioni di pressione all'interno di un impianto costituito da un compressore industriale. A partire dalle misure effettuate si è progettato l'apparato sperimentale per il test dei dispositivi risonanti presentati nel quarto capitolo.

Nel **Sesto Capitolo** sono stati progettati alcuni dispositivi risonanti di tipo side-branch ed Helmholtz a geometria fissa, allo scopo di determinare sperimentalmente le loro performance di abbattimento quando collegati all'impianto di test. L'analisi sperimentale è stata effettuata per stabilire quali caratteristiche geometriche di questi dispositivi possano essere modificate da un sistema di attuazione, al fine di adattare alcuni parametri geometrici alla reale oscillazione presente all'interno dei condotti in cui vengono inseriti. La scelta del dispositivo a geometria variabile è ricaduta sui risonatori di Helmholtz cilindrici: tale scelta ha portato alla progettazione di un risonatore di Helmholtz con la lunghezza della camera di risonanza variabile. Un'indagine sperimentale ha permesso di rilevare le ottime performance di abbattimento di que-

sto dispositivo che rispettano da una parte i risultati ottenuti dal modello analitico e, dall'altra confermano la bontà dei criteri di progettazione utilizzati.

Il naturale proseguimento del lavoro vede la determinazione dei limiti fisici entro i quali i risonatori di Helmholtz permettono di ottenere buoni abbattimenti delle oscillazioni di pressione: in questo senso dovranno essere progettati e testati dei risonatori di Helmholtz in cui siano fatte variare le loro grandezze geometriche allo scopo di determinare eventuali limiti dettati dalle non linearità e dalla presenza di eccessive perdite di carico. Un altro importante fattore che influenza il funzionamento dei dispositivi risonanti è rappresentato dall'energia presente nel sistema: a tal fine, le analisi condotte sull'impianto di test dovranno ricondursi all'impianto alimentato dal compressore industriale. Disponendo in questo caso di un'energia delle oscillazioni di pressione molto più ampia di quella fornita dall'impianto di test, sarà possibile valutare l'efficacia di abbattimento dei dispositivi risonanti a frequenze inferiori rispetto al limite ottenuto nell'impianto di test, dove alle basse frequenze il sistema non è in grado di fornire sufficiente energia. L'utilizzo di questi dispositivi passivi in un compressore industriale permetterà di valutare i benefici del loro utilizzo, che possono risultare in un miglioramento dell'affidabilità, in un aumento dei rendimenti della macchina, nella riduzione degli interventi di manutenzione e nella riduzione degli ingombri della macchina. Questi benefici porterebbero conseguentemente ad una riduzione dei costi associati al trasporto, alla manutenzione ed al funzionamento della macchina.

Parallelamente al controllo delle oscillazioni di pressione per mezzo dei dispositivi risonanti, gli algoritmi per il controllo attivo del rumore sviluppati in questo lavoro di tesi dovranno essere rielaborati al fine di permettere un abbattimento di rumori non stazionari e con una banda di frequenze più estesa. Il miglioramento di questa tecnologia consentirà la sua applicabilità nei dispositivi industriali dove la propagazione del rumore avviene principalmente lungo una direzione: impianti industriali dove si verifica tale condizione sono i ventilatori industriali. Negli sviluppi futuri di questo lavoro di tesi si dovranno cercare delle soluzioni a basso costo per l'implementazione delle tecniche di controllo attivo, valutando parallelamente l'eventuale degrado delle prestazioni. La tecnologia di controllo attivo del rumore pur avendo raggiunto oggi uno sviluppo particolarmente ampio, non trova un'applicazione altrettanto ampia nei dispositivi industriali a causa delle difficoltà teoriche nella modellazione e dei costi di applicazione di queste tecnologie che possono risultare troppo elevati.

## APPENDICE A

# Complementi analitici

### A.1 Punti di minimo di una funzione

Nella minimizzazione di una funzione si possono riscontrare principalmente due tipi di minimo, noti in letteratura come minimi locali e minimi globali. Data una generica funzione  $f : A \subset R^n \rightarrow R$  e un vettore  $\mathbf{x} \in A$ , lo sviluppo in serie di Taylor della funzione  $f$  attorno al punto  $\mathbf{x}_0$  risulta:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)\mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \dots, \quad (\text{A.1})$$

dove  $\nabla f(\mathbf{x}_0)$  indica il gradiente della funzione  $f$  calcolato nel punto  $\mathbf{x}_0$  e  $\mathbf{H}(\mathbf{x}_0)$  indica la matrice hessiana della funzione  $f$  calcolata in  $\mathbf{x}_0$ . È possibile dimostrare che  $\mathbf{x}_0$  è un punto di minimo locale se il gradiente in quel punto è nullo e l'hessiano è definito positivo. Poiché il calcolo dell'hessiano è molto complesso, soprattutto quando le dimensioni dello spazio in cui si lavora sono elevate, si semplifica la definizione precedente definendo il minimo locale come quel punto in cui il gradiente della funzione è nullo. Questa semplificazione porta allo sviluppo di algoritmi numerici per la ricerca del minimo molto semplici e veloci rispetto al caso in cui debba essere calcolato anche l'hessiano della funzione, ma c'è il rischio che un qualunque punto in cui  $\|\nabla f(\mathbf{x}_0)\|$  sia scambiato per un punto di minimo; un tipico esempio è rappresentato da una sella e dai punti di massimo locale dove il gradiente della funzione è nullo ma questi non sono punti di minimo. In questi punti il valore dell'hessiano non è strettamente positivo infatti nel caso del massimo locale l'hessiano è definito negativo.

Per la risoluzione del problema di minimo è possibile utilizzare il *metodo del gradiente*, che permette di trovare il minimo non vincolato di una funzione valutando in

punti successivi il valore del gradiente. Preso un punto iniziale  $\mathbf{x}_0$  si inizia un ciclo che termina solo quando il gradiente della funzione assume un valore equiparabile a 0. La posizione successiva dei punti in cui viene valutato il gradiente della funzione è determinata dalla direzione opposta a quella del gradiente stesso, secondo la relazione seguente:

$$d_k = -\nabla f(x_k), \quad (\text{A.2})$$

dove  $d_k$  è la direzione data dall'antigradiente, ossia dal gradiente cambiato di segno, della funzione  $f$  in  $x_k$ . Si ricorda che il gradiente definisce la massima variazione positiva della funzione. Questa scelta è giustificata dal fatto che la direzione dell'antigradiente (normalizzata) è la direzione che minimizza la derivata parziale in  $x_k$  tra tutte le direzioni che hanno norma euclidea unitaria, ossia è la soluzione del seguente problema:

$$\min \nabla f(x_k)^T d, \quad (\text{A.3})$$

$$\|d\|_2 = 1. \quad (\text{A.4})$$

Per la *disuguaglianza di Schwarz*, si ha che il prodotto scalare di due elementi è minore del prodotto delle loro norme:

$$|\nabla f(x_k)^T d| \leq \|d\|_2 \|\nabla f(x_k)\|_2, \quad (\text{A.5})$$

in cui il segno di uguaglianza vale se e solo se  $d = \lambda \nabla f(x_k)$ , con  $\lambda \in R$ . La soluzione del problema A.3 è data allora da:

$$d_k = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|_2}, \quad (\text{A.6})$$

che definisce la direzione dell'antigradiente in  $x_k$ . Tuttavia, occorre osservare che l'ottimalità locale della direzione  $-\nabla f(x_k)$  dipende dalla scelta della norma e che pertanto in un punto fissato  $x_k$  qualsiasi direzione di discesa può interpretarsi come una direzione di massima discesa in una norma opportuna.

Un'ipotesi fondamentale per la convergenza del metodo risiede nella continuità del gradiente: se questa è verificata, la direzione  $-\nabla f(x_k)$  è una direzione di discesa continua rispetto a  $x$ , che si annulla se e solo se  $x$  è un punto stazionario. Questa proprietà garantisce che, scelto un opportuno passo  $\alpha_k$ , è possibile stabilire un risultato di convergenza globale. Il passo  $\alpha_k$  oltre a garantire la convergenza del metodo stabilisce anche la velocità di convergenza del metodo stesso.



Di seguito vengono definiti gli step per l'implementazione numerica del metodo del gradiente:

1. Sia  $\epsilon > 0$  e si prenda un qualsiasi punto iniziale  $x_0 \in R$ .
2. Se  $\|\nabla f(x_k)\| < \epsilon$ , l'algoritmo termina con minimo locale  $x_{loc} = x_k$ .
3. Con  $d_k = -\nabla f(x_k)$  si calcola  $\alpha_k$  come:

$$\alpha_k = \min_{\alpha \geq 0} f(x_k + \alpha d_k). \quad (\text{A.7})$$

Tale passo garantisce la convergenza del metodo del gradiente. Analiticamente esistono diversi metodi per la determinazione del passo, in particolare questo può essere costante o, come in questo caso può dipendere dalla funzione da minimizzare.

4. Si aggiorna il valore di  $x_k$  come:

$$x_{k+1} = x_k + \alpha_k d_k, \quad \alpha_k > 0. \quad (\text{A.8})$$

5. Si calcola il nuovo passo di iterazione  $k = k + 1$  e si ritorna al punto 2).

## A.2 Analisi spettrale di un segnale

L'analisi spettrale è quella tecnica che consente di trarre importanti informazioni a partire da un segnale continuo, ad esempio il segnale acquisito con un comune microfono. Una prima importante grandezza che occorre fissare è la frequenza di campionamento del segnale acquisito dal microfono. La frequenza di campionamento scelta per l'acquisizione limita la frequenza massima rilevabile nello spettro senza commettere aliasing. Un altro parametro fondamentale per il calcolo corretto dello spettro del segnale riguarda il numero di punti  $N$  acquisiti all'interno della finestra di acquisizione. Il numero di punti è legato ovviamente alla lunghezza temporale  $T_{FIN}$  della finestra secondo la relazione:

$$T_{FIN} = N\Delta t, \quad (\text{A.9})$$

dove  $\Delta t$  è la risoluzione temporale del segnale, pari all'inverso della frequenza di campionamento del segnale, secondo la relazione:

$$\Delta t = \frac{1}{f_c}. \quad (\text{A.10})$$

Nella determinazione dello spettro è fondamentale il valore della risoluzione spettrale, che determina la minima distanza alla quale si trovano due righe spettrali per essere individuate come distinte. La risoluzione spettrale è legata al tempo totale di acquisizione del segnale secondo la relazione:

$$\Delta f = \frac{1}{T_{FIN}} = \frac{1}{N\Delta t} = \frac{f_c}{N}, \quad (\text{A.11})$$

dove risulta evidente che, a parità di punti acquisiti, la risoluzione spettrale è tanto minore quanto maggiore è la frequenza di campionamento, di conseguenza essa è limitata dalla capacità di memoria del controllore e dalla massima frequenza di campionamento della scheda.

Il segnale analogico di partenza  $x(t)$  viene campionato e successivamente convertito in digitale dal convertitore ADC della scheda di acquisizione del sistema PXI. Il campionamento di un numero finito  $N$  di campioni del segnale di partenza, da origine ad un segnale a tempo discreto  $x(n)$ . Il numero  $N$  di campioni acquisiti è selezionabile dalla scelta dell'operatore, in particolare dipende dalla lunghezza temporale e dalla frequenza di campionamento del record acquisito. La sequenza di campioni acquisita viene elaborata secondo la formula della trasformata discreta di Fourier (DFT):

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi n \frac{k}{N}}, \quad k = 0, 1, \dots, N-1, \quad (\text{A.12})$$

che fornisce lo spettro campionato del segnale  $x(t)$  in questione, ossia i valori della parte reale e immaginaria che tale spettro assume in corrispondenza di determinate frequenze. La relazione A.12 si ottiene dalla trasformata di Fourier, discretizzando il tempo  $t$ , ossia effettuando il campionamento del segnale, discretizzando  $f$  e sostituendo il simbolo di integrale con la sommatoria su  $N$ . Nel caso in cui  $N$  risulta essere una potenza di 2, è possibile effettuare il calcolo della trasformata in modo efficiente, riducendo la potenza di calcolo necessaria. Il calcolo della trasformata di Fourier richiede in generale  $O(N^2)$  operazioni, tuttavia se  $N$  è una potenza di 2, è possibile utilizzare l'algoritmo di Cooley-Tukey [17], riducendo drasticamente il numero delle operazioni a  $O(N \log_2 N)$ . Sfruttando l'algoritmo di Cooley-Tukey, i campioni della trasformata sono ottenuti con un'ordine diverso da quello "naturale"; questo ordine prende il nome di "bit-reversed order", secondo lo schema  $\{00, 10, 01, 11, \dots\}$  invece di  $\{00, 01, 10, 11, \dots\}$ .

Un problema tipico nel calcolo della DFT si riscontra nella dispersione spettrale, legato al fatto che il segnale temporale  $x(t)$  viene troncato per effetto della finestra di osservazione. Il troncamento fa sì che lo spettro che viene fuori dall'algoritmo di DFT non coinciderà mai con quello reale, infatti il troncamento corrisponde alla convoluzione dello spettro  $X(f)$  con lo spettro  $W(f)$  della finestra di acquisizione, per cui la DFT fornisce una versione campionata non di  $X(f)$  ma del prodotto  $X(f)W(f)$ . Se si considera un segnale  $x(t)$  composto da una sola armonica, la convoluzione comporta che l'energia di  $X(f)$ , inizialmente confinata in una precisa riga spettrale, risulti in realtà dispersa su una banda maggiore a causa dei lobi secondari di cui è composto lo spettro  $W(f)$  della finestra temporale. Questo fenomeno è noto come *leakage*, ossia la dispersione di energia su frequenze dove questa era inizialmente nulla; il leakage comporta una perdita di accuratezza dei risultati in quanto causa un'alterazione delle righe spettrali relative alle frequenze effettivamente presenti nel segnale. Il metodo migliore per minimizzare questo fenomeno consiste nell'utilizzare finestre di acquisizione che non presentino discontinuità agli estremi, ma abbiano una variazione più dolce, in modo che i lobi secondari del relativo spettro decrescano a zero più rapidamente rendendo minima l'energia dispersa.

### A.3 Differenti espressioni per l'equazione delle onde

Al fine di mostrare l'equivalenza delle trattazioni, relative alle espressioni analitiche dell'equazione delle onde, mostrate nei capitoli 1.2 e 4.2, si riportano di seguito i passaggi matematici per passare dall'equazione delle onde 4.15, in termini di spostamento di particella  $\xi$ :

$$\frac{\partial^2 \xi}{\partial t^2} + \bar{\gamma} \frac{\partial \xi}{\partial t} = c^2 \frac{\partial^2 \xi}{\partial x^2}, \quad (\text{A.13})$$

all'equazione delle onde 1.26, in termini di oscillazione di pressione dinamica  $p$ :

$$\frac{\partial^2 p}{\partial t^2} = c^2 \frac{\partial^2 p}{\partial x^2}. \quad (\text{A.14})$$

Si consideri la derivata rispetto alla coordinata  $x$  della relazione A.13:

$$\frac{\partial}{\partial x} \left( \frac{\partial^2 \xi}{\partial t^2} + \bar{\gamma} \frac{\partial \xi}{\partial t} \right) = \frac{\partial}{\partial x} \left( c^2 \frac{\partial^2 \xi}{\partial x^2} \right), \quad (\text{A.15})$$

scambiando l'ordine di differenziazione si ha:

$$\frac{\partial^2}{\partial t^2} \frac{\partial \xi}{\partial x} + \bar{\gamma} \frac{\partial}{\partial t} \frac{\partial \xi}{\partial x} = c^2 \frac{\partial^2}{\partial x^2} \frac{\partial \xi}{\partial x}. \quad (\text{A.16})$$

Sostituendo il legame tra la pressione  $p$  e la derivata spaziale dello spostamento di particella  $\frac{\partial \xi}{\partial x}$ , ottenuta a partire dalla relazione 4.5:

$$\frac{\partial \xi}{\partial x} = -\frac{p}{\rho_0 c^2}, \quad (\text{A.17})$$

nella relazione A.16, si ottiene:

$$\frac{\partial^2}{\partial t^2} \left( -\frac{p}{\rho_0 c^2} \right) + \bar{\gamma} \frac{\partial}{\partial t} \left( -\frac{p}{\rho_0 c^2} \right) = c^2 \frac{\partial^2}{\partial x^2} \left( -\frac{p}{\rho_0 c^2} \right), \quad (\text{A.18})$$

che semplificata può essere riscritta come:

$$\frac{\partial^2 p}{\partial t^2} + \bar{\gamma} \frac{\partial p}{\partial t} = c^2 \frac{\partial^2 p}{\partial x^2}. \quad (\text{A.19})$$

La relazione A.19 risulta formalmente identica, a meno del termine dissipativo, all'equazione delle onde A.14 pertanto, quanto appena mostrato consente di affermare l'equivalenza dei due approcci presentati.

## A.4 Stima della funzione di trasferimento di un sistema

Per effettuare il calcolo della funzione di trasferimento è possibile utilizzare diverse tecniche. Nel seguito si descriveranno alcune di queste, che differenziano i risultati ottenuti in fase di post-processing dei dati. Esse infatti presentano delle funzioni di stima della funzione di trasferimento partendo da dati già acquisiti.

Un primo metodo per effettuare il calcolo della funzione di trasferimento consiste nello sfruttare la definizione stessa di funzione di trasferimento, ovvero:

$$H(f) = \frac{Y(f)}{X(f)}, \quad (\text{A.20})$$

dove  $X(f)$  e  $Y(f)$  sono rispettivamente gli spettri del segnale in ingresso e del segnale in uscita. Modulo e fase vengono calcolati nel modo seguente:

$$\|H(f)\| = \frac{\|Y(f)\|}{\|X(f)\|}, \quad (\text{A.21})$$

$$\angle H(f) = \angle Y(f) - \angle X(f). \quad (\text{A.22})$$

Ovviamente, fornendo come ingresso una forzante sinusoidale di frequenza fissa e pari a  $\tilde{f}$  la funzione di trasferimento avrà valori sensati solamente per  $f = \tilde{f}$ , poiché sulle altre frequenze non sono presenti ingressi determinabili.

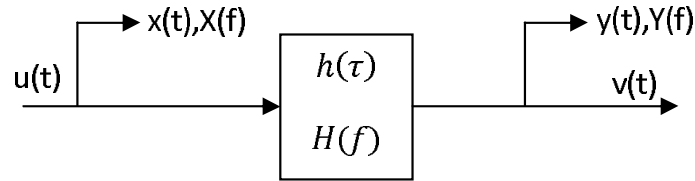


Figura A.1: Schema del sistema di cui si vuole eseguire la caratterizzazione in frequenza.

Un secondo metodo per effettuare il calcolo considera degli stimatori basati sul calcolo degli autospettri e crosspettri dei segnali. Ipotizzando che il sistema di cui si vuole ricavare la funzione di trasferimento sia stabile, lineare e a coefficienti costanti, si possono definire tre stimatori della funzione di trasferimento. Il sistema che si considera, mostrato in figura A.1, è costituito da un segnale di ingresso  $u(t)$  e dalla sua misura  $x(t)$ , dal segnale di uscita  $v(t)$  e dalla sua misura  $y(t)$ . Riscrivendo la relazione A.20 come:

$$Y(f) = H(f)X(f), \tag{A.23}$$

e premoltiplicando ambo i membri della A.23 per  $X(f)^*$ , che rappresenta lo spettro complesso coniugato dello spettro  $X(f)$  del segnale di ingresso, secondo la relazione:

$$X(f)^*Y(f) = H(f)X(f)^*X(f), \tag{A.24}$$

si ottiene lo stimatore  $H_1$  come:

$$H_1(f) = \frac{G_{xy}(f)}{G_{xx}(f)}, \tag{A.25}$$

dove  $G_{xy}(f)$  rappresenta il crosspettro tra i segnali di ingresso e di uscita del sistema e  $G_{xx}(f)$  rappresenta invece l'autospettro del segnale di ingresso.  $H_1$  stima perfettamente  $H$  solo se vengono stimate perfettamente le funzioni degli autospettri e crosspettri. Premoltiplicando ambi i membri della A.23 per  $Y(f)^*$ , che rappresenta lo spettro complesso coniugato dello spettro  $Y(f)$  del segnale di uscita, si ha:

$$Y(f)^*Y(f) = H(f)Y(f)^*X(f), \tag{A.26}$$

si ottiene in questo modo un nuovo stimatore  $H_2$ , dato dalla seguente relazione:

$$H_2(f) = \frac{G_{yy}(f)}{G_{yx}(f)}, \quad (\text{A.27})$$

dove  $G_{yy}(f)$  rappresenta l'autospettro del segnale di uscita.

Considerando la relazione A.23 è possibile premoltiplicare a sinistra per  $Y(f)^*$  e a destra per  $H(f)^*X(f)^*$ , avendo quindi:

$$Y(f)^*Y(f) = H(f)^*H(f)X(f)^*X(f). \quad (\text{A.28})$$

Si ottiene in questo modo lo stimatore  $H_a$  che possiede, rispetto ai primi due, delle caratteristiche differenti:

$$\|H_a(f)\|^2 = \frac{G_{yy}(f)}{G_{xx}(f)}. \quad (\text{A.29})$$

$H_a(f)$  è una grandezza reale, pertanto con questo stimatore non è possibile risalire allo sfasamento della funzione di trasferimento.

Ciascuno dei tre stimatori presentati ha dei vantaggi rispetto agli altri a seconda della situazione in cui viene effettuata la misura.

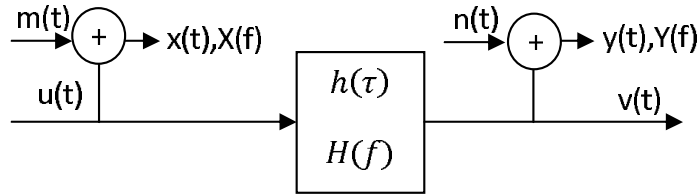


Figura A.2: Schema del sistema di cui si vuole eseguire la caratterizzazione in frequenza, affetto dalla presenza di rumore.

Se si considera un sistema affetto da rumore incorrelato, come quello mostrato in figura A.2, in cui si ha la presenza di rumore incorrelato sulla misura del segnale di ingresso  $m(t)$  o sulla misura del segnale di uscita  $n(t)$ , gli stimatori hanno comportamenti differenti in quanto le funzioni di autospettro e crosspettro dei segnali misurati risentono della presenza del rumore, come mostrato nelle equazioni A.30, A.31 e A.32:

$$G_{xx}(f) = G_{uu}(f) + G_{mm}(f) + G_{um}(f) + G_{mu}(f), \quad (\text{A.30})$$

$$G_{yy}(f) = G_{vv}(f) + G_{nn}(f) + G_{vn}(f) + G_{nv}(f), \quad (\text{A.31})$$

$$G_{xy}(f) = G_{uv}(f) + G_{un}(f) + G_{mv}(f) + G_{mn}(f). \quad (\text{A.32})$$

Nel caso nel sistema sia presente solo rumore sul segnale di uscita i valori dei tre stimatori della funzione di trasferimento sono i seguenti:

$$H_1(f) = \frac{G_{xy}(f)}{G_{xx}(f)}, \quad (\text{A.33})$$

$$H_2(f) = \frac{G_{yy}(f)}{G_{xy}(f)} = H(f) \left( 1 + \frac{G_{nn}(f)}{G_{vv}(f)} \right), \quad (\text{A.34})$$

$$\|H_a(f)\| = \frac{G_{yy}(f)}{G_{xx}(f)} = \|H(f)\|^2 \left( 1 + \frac{G_{nn}(f)}{G_{vv}(f)} \right). \quad (\text{A.35})$$

Si nota come le stime della funzione di trasferimento calcolate attraverso gli stimatori  $H_2(f)$  e  $H_a(f)$  siano affette da errori causati dalla presenza del rumore.

Viceversa, nel caso sia presente del rumore incorrelato solo sul segnale di ingresso, i valori dei tre stimatori sono i seguenti:

$$H_1(f) = \frac{G_{xy}(f)}{G_{xx}(f)} = H(f) \left( \frac{1}{1 + \frac{G_{mm}(f)}{G_{uu}(f)}} \right), \quad (\text{A.36})$$

$$H_2(f) = \frac{G_{yy}(f)}{G_{xy}(f)}, \quad (\text{A.37})$$

$$\|H_a(f)\| = \frac{G_{yy}(f)}{G_{xx}(f)} = \|H(f)\|^2 \left( \frac{1}{1 + \frac{G_{mm}(f)}{G_{uu}(f)}} \right). \quad (\text{A.38})$$

Utilizzando lo stimatore  $H_1(f)$  si commette un errore che tende a sottostimare il modulo della funzione di trasferimento, mentre la stima delle fasi rimane invariata.

Nel caso si abbia la presenza di rumore sia sugli ingressi che sulle uscite, con lo stimatore  $H_1(f)$  si ottiene una sottostima di  $H(f)$ , mentre con  $H_2(f)$  si ottiene una sovrastima di  $H(f)$ , è possibile dunque calcolare il limite superiore ed il limite inferiore all'interno dei quali giace il valore reale della funzione di trasferimento.





## APPENDICE B

# Sensori e apparecchiature

### B.1 Sistemi di acquisizione

Si riportano nella tabella B.1 i dati principali dell'hardware di acquisizione Bruel & Kjaer 3560-B.

Ingressi e uscite analogiche	5 canali di input 1 canale di output
Campionamento	Parallelo
Frequenza di analisi	24-bit: 460 $kHz$ (18 canali) 16-bit: 700 $kHz$ (27 canali)
Risoluzione ADC	24-bit
Voltaggio input	8 range: $7.071 mV_{peak} \div 7.07 V_{peak}$ 10 $dB$ ogni step
Condizionamento	Collegamento con amplificatori Delta-Tron, accelerometri ICP o preamplificatori per microfoni. Riconoscimento automatico dei sensori collegati al front-end.
Rispetto standard di sicurezza	Si
Integrato con software proprietario	Si

Tabella B.1: *Principali caratteristiche della piattaforma Bruel & Kjaer 3560B.*

Si riportano nella tabella B.2 i dati principali dell'hardware di acquisizione e controllo National Instruments PXI.

<b>Chassis</b>	<b>NI PXI-1031</b>
Numero slot	4
<b>Controller</b>	<b>NI PXI-8106</b>
Processore	Intel Core 2 Duo T7440 (2.16 GHz dual core)
Hard-disk	60 GB SATA (minimo)
Ram	DDR2 a 667 MHz, fino a 2 GB
<b>DAQ</b>	<b>NI PXI-6133</b>
Ingressi analogici	8 canali di input
Campionamento	Simultaneo
Frequenza di analisi	3 MHz a canale
Risoluzione ADC	14 bit
Tensione input	$\pm 10 V, \pm 5 V, \pm 2.5 V, \pm 1.25 V$
Condizionamento	/
<b>DAQ</b>	<b>NI PXI-6123</b>
Ingressi analogici	8
Campionamento	Simultaneo
Frequenza di analisi	500 kHz a canale
Risoluzione ADC	16 bit
Tensione input	$\pm 10 V, \pm 5 V, \pm 2.5 V, \pm 1.25 V$
Condizionamento	/
<b>Output analogico</b>	<b>NI PXI-6733</b>
Uscite analogiche	8 canali di output (tensione)
Aggiornamento	Simultaneo
Frequenza di analisi	740 kHz a canale
Risoluzione ADC	16 bit
Tensione output	$\pm 10 V, \pm AO EXT REF$
Condizionamento	/

Rispetto standard di sicurezza	Si
Integrato con software proprietario per il controllo	Si
<b>Counter-Timer</b>	<b>NI PXI-6602</b>
N° canali	8 canali 32-bit up/down counter/timer
Sorgente di clock	80 MHz
Canali digitali	32 canali I/O digitale (5 V TTL/CMOS)

Tabella B.2: *Principali caratteristiche della piattaforma NI PXI.*

## B.2 Amplificatori di segnale

Al fine di poter interpretare correttamente il segnale in uscita da un qualunque trasduttore, è necessario disporre di un segnale condizionato, ossia disporre di un segnale con ben determinate caratteristiche. Lo strumento che permette di effettuare il condizionamento e parallelamente l'amplificazione del segnale in uscita da un trasduttore è comunemente chiamato *Conditioning Amplifier*.

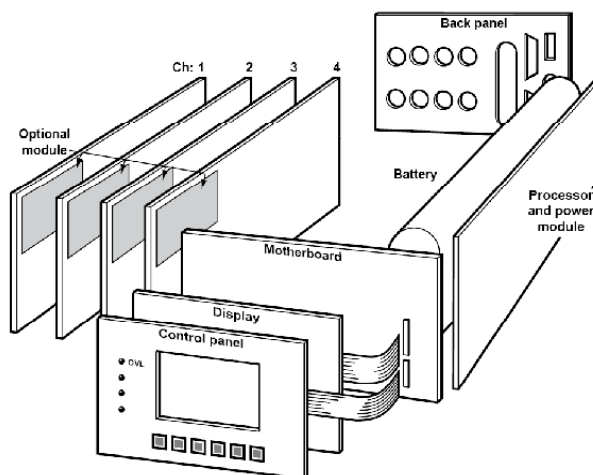


Figura B.1: *Configurazione interna dell'amplificatore NEXUS.*

Avendo deciso di utilizzare dei trasduttori Bruel & Kjaer, che richiedono un condizionamento del segnale di tipo DeltaTron, si è scelto un amplificatore del tipo *NEXUS*

*Conditioning Amplifier* [15]. Il condizionamento DeltaTron identifica i prodotti che utilizzano una corrente costante di alimentazione e che forniscono un segnale in uscita in modulazione di tensione sulla linea elettrica della stessa alimentazione. Ogni amplificatore della categoria NEXUS può contenere fino a quattro canali indipendenti con un'ampia banda passante e differenti filtri come ad esempio filtri passa-banda. Questi amplificatori possono essere configurati con differenti moduli di ingressi con un differente numero di canali a seconda delle esigenze. Ogni canale (1, 2, 3, e 4 nella figura B.1) è costituito da un modulo comune, un modulo di ingresso sostituibile, un modulo opzionale e un modulo di uscita. Il modulo comune contiene i filtri, i parametri di guadagno e le funzioni di calibrazione. I moduli di ingresso possono essere scelti a seconda dei trasduttori che si intendono utilizzare. L'amplificatore utilizzato per l'esecuzione delle prove è identificato con il seguente codice: 2693 – A – 0S4. Esso contiene quattro moduli di ingresso di tipo DeltaTron; la scelta di questa configurazione è stata dettata dalla necessità di avere un riconoscimento certo del tipo di trasduttore, in particolare questo si ritiene utile con le schede di acquisizione del sistema PXI prodotto da National Instruments dove, essendo un sistema general-purpose, non si ha il condizionamento del segnale. Di seguito si riportano le caratteristiche principali dell'amplificatore scelto:

- 4 moduli di ingresso.
- 4 uscite: a seconda della lunghezza del cavo è determinata la frequenza della linea, con un cavo di 20 m si arriva alla frequenza di 100 kHz, con un cavo di 1000 m si arriva a 2 kHz.
- Vengono utilizzati connettori BNC, e può essere scelta un'uscita come single ended oppure floating.
- Alimentazione AC/DC.
- Uscita AC.
- Detenzione errori nella configurazione hardware, viene ad esempio rilevato un errato collegamento dei trasduttori.
- Detenzione sovraccarichi di corrente/tensione: viene utilizzata per rilevare eccessive correnti di alimentazione nei microfoni preamplificati; viene quindi indicato un sovraccarico che sarebbe difficilmente rilevabile, specialmente quando si hanno lunghi cavi e segnali con elevato contenuto armonico.

- Peak Meter: monitoraggio dei valori di picco istantanei per tutti i canali.
- Alimentazione in corrente a 4 o 10 mA.
- Impedenza ingresso 1 M $\Omega$  - 100 pF.
- Range frequenza: 0.1 Hz - 100 kHz (guadagno < 60 dB).
- Connessioni LEMO a 7 contatti per la connessione di microfoni preamplificati. Possono essere utilizzati ingressi differenziali (floating inputs) per garantire la massima protezione dai disturbi elettromagnetici (EMI protection).
- Accurato guadagno di controllo: è possibile avere un'uscita compensata in funzione del guadagno; per tutti gli intervalli di guadagno e per ogni tipologia di filtro adottato viene fatta una compensazione automatica del guadagno, ciò è reso possibile grazie a test e calibrazioni effettuate da Bruel & Kjaer. Nella figura B.2 viene mostrato il rumore in uscita misurato in bande a 22 kHz in funzione degli intervalli di guadagno impostati dal costruttore.
- Possibilità utilizzo batteria per la portabilità.
- Supporto TEDS (Transducer Electronic Data Sheet).

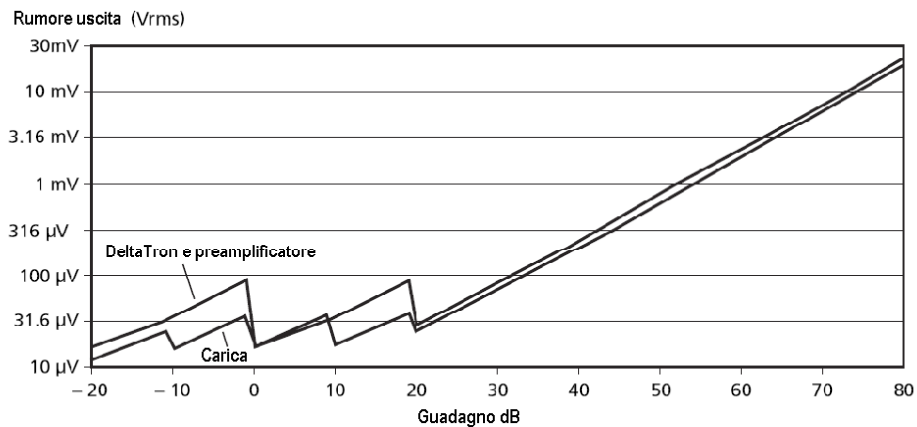


Figura B.2: Rumore in uscita in funzione del guadagno scelto dall'utente.

## B.3 Microfoni

Per lo studio si sono utilizzati due microfoni prodotti da Bruel & Kjaer, modello 4944. I parametri principali dei microfoni acquisiti sono i seguenti:

- Dimensione:  $1/4''$ .
- Accuratezza:  $1.0 \text{ mV/Pa}$ .
- Range frequenze:  $4 \text{ Hz} - 70 \text{ kHz}$ .
- Range dinamico:  $30 - 170 \text{ dB(A)}$ .
- Temperatura di utilizzo:  $-40^\circ - 150^\circ$ .

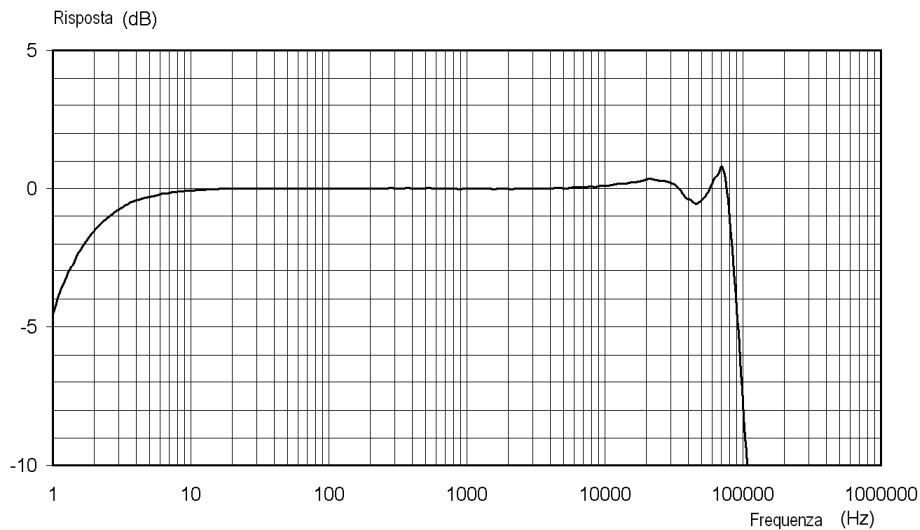


Figura B.3: *Curva risposta in frequenza microfono 4944.*

La figura B.3 mostra la curva di risposta in frequenza di questi microfoni; è possibile notare come la banda sia piatta in un grande intervallo di frequenze evidenziando la qualità costruttiva di questi sensori.

## B.4 Attuatori acustici

In riferimento agli attuatori acustici scelti nel paragrafo 3.2, si riportano le principali caratteristiche degli attuatori acustici.

- Full-range “Visaton R 10 S 8 Ohm”:

- Potenza nominale: 20  $W$ ,
- Potenza picco: 30  $W$ ,
- Impedenza: 8  $\Omega$ ,
- Range frequenze: 100  $Hz$  - 13  $kHz$ ,
- $f_c$  (frequenza di risonanza): 160  $Hz$ ,
- SPL: 90  $dB$ .
- Massimo spostamento del cono: 3  $mm$
- Induzione magnetica: 0.95  $Tesla$
- Flusso magnetico: 135  $\mu Weber$
- Full-range “Visaton FRS 8 8 Ohm”
  - Potenza nominale: 30  $W$  (6000  $Hz$ )
  - Potenza picco: 50  $W$  (6000  $Hz$ )
  - Impedenza: 8  $\Omega$
  - Range frequenze: 100  $Hz$  - 20  $kHz$
  - $f_c$  (frequenza di risonanza): 130  $Hz$
  - SPL: 82  $dB$
  - Massimo spostamento del cono: 5  $mm$
  - Induzione magnetica: 0.8  $Tesla$
  - Flusso magnetico: 200  $\mu Weber$

## B.5 Trasduttori di pressione

Per la rilevazione delle oscillazioni di pressione dinamica nell'impianto industriale (par. 5.1.2) e nell'impianto di test realizzato (par. 5.2), sono stati utilizzati otto trasduttori di pressione, quattro con un basso fondoscala e quattro con un fondoscala più elevato. La scelta di utilizzare due diverse tipologie di trasduttori è dettata dal fatto che, prima di effettuare la campagna di misura sull'impianto industriale, non si era a conoscenza dei massimi livelli delle oscillazioni di pressione presenti (in particolare le oscillazioni di pressione presenti nell'ultimo stadio di compressione del compressore industriale). Ovviamente la scelta più opportuna è quella di utilizzare dei sensori con fondoscala

prossimo alla massima ampiezza delle oscillazioni al fine di ottenere una migliore risoluzione nella misura. Data la necessità di rilevare le oscillazioni di pressione anche nei condotti di mandata del compressore prossimi al cilindro dove il fluido raggiunge elevate temperature superiori anche a  $100^{\circ} C$ , è stato scelto di utilizzare dei trasduttori di pressione con uscita in carica (*Coulomb*) e di realizzare lo stadio di amplificazione per mezzo di amplificatori in linea posizionati ad una certa distanza dal sensore. In questo modo l'elettronica e quindi la misura risulta meno influenzata dalle elevate temperature in gioco. Si è scelto di utilizzare sensori e amplificatori di carica della casa costruttrice *Dytran Instruments Incorporated*. Nell'elenco seguente si riportano le caratteristiche dei trasduttori e dei relativi amplificatori di carica utilizzati:

- Trasduttore con basso fondoscala: *Dytran 2200C1*.
  - Sensibilità:  $1 pC/PSI$ .
  - Fondoscala:  $\pm 100 PSI (\pm 7 bar)$ .
  - Frequenza di risonanza:  $250 kHz$ .
  - Temperatura di funzionamento:  $-70 \div +260^{\circ} C (-100 \div +500^{\circ} F)$ .
  - Coefficiente di sensibilità termica:  $0.01 \%/^{\circ} F$ .
  
- Trasduttore con elevato fondoscala: *Dytran 2200C4*.
  - Sensibilità:  $1 pC/PSI$ .
  - Fondoscala:  $\pm 1000 PSI (\pm 70 bar)$ .
  - Frequenza di risonanza:  $250 kHz$ .
  - Temperatura di funzionamento:  $-70 \div +260^{\circ} C (-100 \div +500^{\circ} F)$ .
  - Coefficiente di sensibilità termica:  $0.01 \%/^{\circ} F$ .
  
- Amplificatore di carica *Dytran 4751B1* per sensori *Dytran 2200C1*.
  - Sensibilità:  $50 mV/pC$ .
  - Fondoscala:  $\pm 100 pC (\pm 5V)$ .
  - Temperatura di funzionamento:  $-45 \div +90^{\circ} C (-50 \div +188^{\circ} F)$ .
  - Coefficiente di sensibilità termica:  $0.02 \%/^{\circ} F$ .
  - Tensione di alimentazione:  $+18 \div +30 V (2 \div 20 mA)$ .



- Amplificatore di carica *Dytran 4751B3* per sensori *Dytran 2200C4*.
  - Sensibilità:  $1 \text{ mV/pC}$ .
  - Fondoscala:  $\pm 5000 \text{ pC}$  ( $\pm 5 \text{ V}$ ).
  - Temperatura di funzionamento:  $-45 \div +90 \text{ }^\circ\text{C}$  ( $-50 \div +188 \text{ }^\circ\text{F}$ ).
  - Coefficiente di sensibilità termica:  $0.02 \text{ } \%/^\circ\text{F}$ .
  - Tensione di alimentazione:  $+18 \div +30 \text{ V}$  ( $2 \div 20 \text{ mA}$ ).



## APPENDICE C

# Script di MATLAB utilizzati per il sistema ANC

### C.1 Script MatLab per la mappatura dell'RMS della pressione

Nel presente paragrafo viene riportato lo script di MatLab per l'analisi dell'andamento del valore quadratico medio della pressione acustica rilevata dal microfono di errore in funzione dei parametri di controllo  $A_{cont}$  e  $\phi_{cont}$ .

```
%% Dati segnale di rumore
A_rum=1; f_rum=300; phi_rum=deg2rad(60);

%% Dati segnale di controllo
A_cont=[-2:0.2:2]; f_cont=f_rum; phi_cont=deg2rad([0:5:360]);

%% Dati sull'acquisizione dei punti
freq=300; fc=50000; dt=1/fc; Tacq=0.1; t=[0:dt:Tacq];

%% Calcolo dei segnali
sig_rum=A_rum*sin(2*pi*f_rum*t+phi_rum);

k=1; h= waitbar(0,'Attendere, risoluzione in
corso...', 'Name', 'CALCOLO DELL''RMS');
for i=1:length(A_cont)
    for j=1:length(phi_cont)
        sig_cont=A_cont(i)*sin(2*pi*f_cont*t+phi_cont(j));
        ris(k,:)=sig_rum+sig_cont;
        RMS(i,j)=sqrt(sum(ris(k,:).^2)/length(t));
        k=k+1;
    end
    waitbar(i/length(A_cont));
```

```

end close(h)

phi = phi_cont; amp = A_cont;

figure,surf(rad2deg(phi),amp,RMS) title('Andamento
$RMS$', 'Interpreter', 'Latex' ) ylabel('$A_{cont}$
$[Pa]$', 'Interpreter', 'Latex') xlabel('$\phi_{cont}$ $[\^{\circ}]$
$', 'Interpreter', 'Latex') zlabel('$RMS$
$[Pa]$', 'Interpreter', 'Latex') axis([0 360 -1.5 1.5 0 1.5])

figure, [C,h]=contour(rad2deg(phi),amp,RMS);
set(h, 'ShowText', 'on', 'LevelList', [0:0.1:5], 'TextStep', 10, 'TextList', [0:0.1:0.5
0.6:0.5:5]) title('Andamento $RMS$', 'Interpreter', 'Latex')
ylabel('$A_{cont}$ $[Pa]$', 'Interpreter', 'Latex')
xlabel('$\phi_{cont}$ $[\^{\circ}]$', 'Interpreter', 'Latex')

```

## C.2 Script MatLab per la mappatura dello spettro della pressione

Nel presente paragrafo viene riportato lo script di MatLab per l'analisi dell'andamento dell'ampiezza del picco spettrale del segnale rilevato dal microfono di errore in funzione dei parametri di controllo  $A_{cont}$  e  $\phi_{cont}$ .

```

%% Dati segnale di rumore
A_rum=0.5; f_rum=1000; phi_rum=deg2rad(0);

%% Dati segnale di controllo
A_cont=[-1.5:0.1:1.5]; f_cont=f_rum; phi_cont=deg2rad([0:5:360]);

%% Dati sull'acquisizione dei punti
fc=50000; dt=1/fc; T=0.1; df=1/T; n_punti_acq=T*fc;
n_punti_sp=2^(floor(log(n_punti_acq)/log(2))); Tacq=n_punti_sp*dt;
df_sp=1/Tacq; tempo=[0:dt:(n_punti_sp-1)*dt];
frequenze=0:df_sp:(n_punti_sp/2)*df_sp;
indice_freq=max(find(frequenze<=f_rum));

%% Calcolo dei segnali
% Segnale di rumore
sig_rum=A_rum*sin(2*pi*f_rum*tempo+phi_rum);

k=1; att= waitbar(0,'Attendere, risoluzione in
corso...', 'Name', 'CALCOLO DEGLI SPETTRI'); for i=1:length(A_cont)
    for j=1:length(phi_cont)
        sig_cont(k,:)=A_cont(i)*sin(2*pi*f_cont*tempo+phi_cont(j));
        sig_tot(k,:)=sig_rum+sig_cont(k,:);
        % Calcolo spettro segnale
    end
end

```

```

    spettro_sig=fft(sig_tot(k,:));
    % Calcolo modulo spettro
    modulo_spettro=[spettro_sig(1)/n_punti_sp; 2/n_punti_sp*...
        abs(spettro_sig(2:(n_punti_sp/2+1)))];
    val_mod(i,j)=max(modulo_spettro(indice_freq-2:indice_freq+2));
    k=k+1;
end
waitbar(i/length(A_cont))
end close (att)

phi = phi_cont; amp = A_cont;

figure,surf(rad2deg(phi),amp,val_mod) title('Andamento del modulo
dello spettro','Interpreter','Latex' ) ylabel('$A_{cont}$
$[Pa]$', 'Interpreter','Latex') xlabel('$\phi_{cont}$ $[^\circ]$', 'Interpreter','Latex') zlabel('$|Spettro|$
$[Pa]$', 'Interpreter','Latex') axis([0 360 -1.5 1.5 0 1.5])

figure, [C,h]=contour(rad2deg(phi),amp,val_mod);
set(h,'ShowText','on','LevelList',[0:0.1:5],'TextStep',10,'TextList',[0:0.1:0.5
0.6:0.5:5]) title('Andamento del modulo dello
spettro','Interpreter','Latex' ) ylabel('$A_{cont}$
$[Pa]$', 'Interpreter','Latex') xlabel('$\phi_{cont}$ $[^\circ]$', 'Interpreter','Latex') zlabel('$|Spettro|$
$[Pa]$', 'Interpreter','Latex')

```



## APPENDICE D

# Codice di controllo per il sistema ANC

### D.1 Codice sorgente per l’algoritmo ARMS

Il codice sorgente è stato realizzato con l’ausilio di due differenti editor, un primo editor è quello fornito da National Instruments e noto come *LabWindows CVI* mentre il secondo editor utilizzato è *Microsoft Visual Studio*. Con questi editor è stato realizzato il codice sorgente nel quale sono state implementate le principali funzioni per l’inizializzazione, per il ripristino in caso di terminazione improvvisa del sistema di controllo e per l’esecuzione dell’algoritmo di controllo. Il codice è stato strutturato su tre file: un file principale che contiene le tre funzioni, un header file che consente l’esportazione delle tre funzioni e un secondo header file che contiene le funzioni secondarie necessarie per la realizzazione delle funzioni svolte dall’algoritmo di controllo.

#### D.1.1 Struttura del codice

Vengono di seguito riportati i file dei codici sorgenti per il controllo del rumore mediante minimizzazione del valore quadratico medio del segnale acquisito dal microfono di errore:

1. *cvidll.c*
2. *cvidll.h*
3. *func.lib.h*

## File "cvidll.c"

```

/*-----*/
/* Include files */
/*-----*/
#include <windows.h>
#include <analysis.h>
#include <cvirte.h>
#include "cvidll.h"
#include "func_lib.h"
#include <NIDAQmx.h>
#include <utility.h>
#include <ansi_c.h>
#include <math.h>

/*-----*/
/* Definizione e inizializzazione variabili globali */
/*-----*/
TaskHandle hMyTask=-1; //Task di uscite analogiche
double Fc_out=100000; //frequenza di campionamento delle uscite
double *A=NULL; //array ampiezze rumore e controllo
double *f=NULL; //array frequenze rumore e controllo
double *phi=NULL; //array fasi rumore e controllo
double *segnale=NULL; //array segnale da scrivere in uscita
double L_buffer=500; //numero di punti del buffer di scrittura
double memoria_alloc=100000; //numero di punti iniziale buffer di scrittura

TaskHandle kMyTask=-1; //Task di ingressi analogici
double Fc_in =50000; //frequenza di campionamento lettura ingressi
double scala_mic=3.16456; //Scala conversione Volt -> Pascal
int NumChannels_in=2; //numero di canali in ingresso
double L_buffer_mic=4900; //numero di punti del buffer di lettura
double *array_mic1=NULL; //puntatore array lettura microfono 1
double *array_mic2=NULL; //puntatore array lettura microfono 2
double *array_read=NULL; //puntatore array segnali microfoni

int32 samples_read; //campioni acquisiti

int stato_lettura=0; //flag dello stato della lettura
int passo; //contatore dei passi dell'algoritmo
int passo_old=100; //contatore dei passi dell'algoritmo

double A_cont[10]; //array ampiezze controllo
double f_cont[10]; //array frequenze controllo
double phi_cont[10]; //array fasi controllo

double A_cont_temp[10]; //array ampiezze del controllo
double phi_cont_temp[10]; //array fasi del controllo
double A_rum_temp[10];

```



```

int n_freq_cont=1;           //numero di frequenze del controllo
int n_freq_old=2;           //numero di frequenze di rumore + controllo

uInt32 campioni_acq;        //campioni acquisiti finora dalla scheda
double RMS_0=0;
double RMS_1=0;
double RMS_2=0;
double RMS_3=0;

int condizione_RMS=0;       //flag di verifica della condizione sull'RMS
int step=0;                 //flag operazioni sul calcolo dei gradienti
double grad_phi=0;         //gradiente della fase
double grad_A=0;           //gradiente dell'ampiezza
double k1=0;               //costante di moltiplicazione gradienti
double k1_init=0;          //valore iniziale della cost. di molt. gradienti
double k1_fin=0;           //valore finale della cost. di molt. gradienti
double k2=0;               //costante di moltiplicazione gradienti effettiva

double tc=10;              //tempo ciclo "Timed Loop" espresso in [ms]
double aspetta;           //contatore del tempo di attesa
int itero=0;               //contatore delle iterazioni dell'algoritmo

double grad_phi_old=0;     //valore gradiente fase iterazione precedente
double grad_A_old=0;      //valore gradiente ampiezza ciclo precedente
int salto_phi=0;           //contatore variazioni segno gradiente fase
int salto_A=0;            //contatore variazioni segno gradiente ampiezza
double A_cont_old=0.0;
double phi_cont_old=0.0;

double A_cont_init=0;     //valore iniziale ampiezza controllo
double delta_A=0;         //valore iniziale variazione ampiezza controllo
double max_Vout=3.0;      //massima tensione positiva ammissibile in uscita
double min_Vout=-3.0;     //massima tensione negativa ammissibile in uscita
double Pa2V=0;           //fattore di conversione Pa2Volt altoparlante

double sat_phi;           //valore di controllo sulla variazione di fase
double sat_A;             //valore di controllo sulla variazione di ampiezza
double tempo_attesa=0.0; //tempo espresso in [ms] di attesa dell'algoritmo

double soglia_RMS=0.15;   //soglia [Pa] raggiungimento punto di minimo

int flag_controllo_old=1; //flag accensione e spegnimento segnale controllo
int flag_rumore_old=1;    //flag accensione e spegnimento segnale rumore

int iterazione=0;

```

```

/* ----- Passi dell'algoritmo di controllo ----- */
enum TypePasso
{
    p_scrivoRum ,           //0
    p_attesaRum ,          //1
    p_leggoRum ,           //2
    p_RMSrum ,             //3
    p_scrivoCont ,        //4
    p_attesaCont ,        //5
    p_leggoCont ,         //6
    p_RMScont ,           //7
    p_scrivoContDelta ,   //8
    p_attesaContDelta ,   //9
    p_leggoContDelta ,    //10
    p_RMScontDelta ,     //11
    p_gradiente ,         //12
    p_verificaRMS ,       //13
    p_STOP                //14
};
passo=p_scrivoRum;

/*-----*/
/* DLL Main */
/*-----*/
int __stdcallDllMain (HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    switch (fdwReason)
    {
        case DLLPROCESS_ATTACH:

            // Respond to DLL loading by initializing the RTE
            if (InitCVIRTE (hinstDLL, 0, 0) == 0)
                return 0;

            break;
        case DLLPROCESS_DETACH:

            // Respond to DLL unloading by closing the RTE for its use
            if (!CVIRTEHasBeenDetached ())
                CloseCVIRTE ();

            break;
    }

    //Return 1 to indicate successful initialization
    return 1;
}

```

```

/*-----*/
/* Funzione 1 - Inizializzazione */
/*-----*/
int Inizializzazione (void)
{
    int res =0;

    /* ----- Creazione task per le uscite analogiche -----*/
    if( hMyTask == -1 )
        res = DAQmxCreateTask(TASK_A01_NAME,&hMyTask);
    else
    {
        res=1;
        return res;
    }

    if( res==0 )
        res = DAQmxCreateAOVoltageChan(hMyTask,"PXI1Slot4/ao0:1","",
            min_Vout , max_Vout , DAQmx_Val_Volts , "");
    else
    {
        res=2;
        return res;
    }

    if( res==0 )
        res = DAQmxSetWriteAttribute(hMyTask, DAQmx_Write_RegenMode,
            DAQmx_Val_AllowRegen);
    else
    {
        res=3;
        return res;
    }

    if( res==0 )
        res = DAQmxCfgSampClkTiming(hMyTask,"",Fc_out , DAQmx_Val_Rising ,
            DAQmx_Val_ContSamps,1);
        // Il buffer vale 1 perché dopo viene sovrascritto
    else
    {
        res=4;
        return res;
    }

    /* ----- Creazione task per gli ingressi analogici -----*/
    if( kMyTask == -1 )
        res = DAQmxCreateTask(TASK_A1_NAME,&kMyTask);
    else
    {

```

```

        res=5;
        return res;
    }

    if( res==0 )
        res = DAQmxCreateAIVoltageChan (kMyTask, "PXI1Slot2/ai0:1", "",
            DAQmx_Val_Cfg_Default, -10.0, 10.0, DAQmx_Val_Volts, NULL);
    else
    {
        res=6;
        return res;
    }

    if( res==0 )
        res = DAQmxCfgSampClkTiming(kMyTask,"",Fc_in,DAQmx_Val_Rising,
            DAQmx_Val_FiniteSamps, (uInt32)L_buffer_mic);
    else
    {
        res=7;
        return res;
    }

    if( array_read == NULL )
        array_read=(double*)malloc(L_buffer_mic*NumChannels_in*sizeof(double));
    if( array_mic1 == NULL )
        array_mic1=(double*)malloc(L_buffer_mic*sizeof(double));
    if( array_mic2 == NULL )
        array_mic2=(double*)malloc(L_buffer_mic*sizeof(double));

    //Allocazione temporanea della memoria per segnale
    if( segnale == NULL )
        segnale =(double *)malloc(memoria_alloc*sizeof(double));
    if( A == NULL )
        A =(double *)malloc(n_freq_old*sizeof(double));
    if( f == NULL )
        f =(double *)malloc(n_freq_old*sizeof(double));
    if( phi == NULL )
        phi =(double *)malloc(n_freq_old*sizeof(double));

    // Inizializzazione variabili
    A_cont[0]=0;
    phi_cont[0]=0;
    aspetta=tc;
    k1_init=k1;
    k2=k1;

    return res;
}

```

```

/*-----*/
/* Funzione 2 - Ciclo_lento */
/*-----*/
double Ciclo_lento (int n_freq_rum, double *A_rum, double *f_rum,
double *phi_rum, double *A_cont, double *phi_cont, double *out1,
double*out2, int controllo_ON, int rumore_ON)
{
    int i,j;
    int res=0;
    double delta_phi=(5.0/180.0)*PI;

    int cond_A=0;
    int cond_phi=0;
    double k2_phi=0.0;
    double k2_A=0.0;
    int dimezza_k1=0;
    double phi_cont_volt;

    // Frequenza controllo = Frequenza rumore
    f_cont[0]=f_rum[0];

    iterazione++;

    // Conversione gradi-radiani
    for (j=0; j<n_freq_rum; j++)
        phi_rum[j]=(phi_rum[j]/180.0)*PI;
    //nota: da ora le fasi si intendono tutte in radianti

    passo_old=passo;
    switch (passo)
    {
    case p_scrivoRum:

        if (n_freq_rum==0)
            passo=p_scrivoRum;
        else
        {
            Eseguo_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                n_freq_cont, A_cont, f_cont, phi_cont);

            Setting_parametri(f_rum[0], &sat_A, &sat_phi, &k1,
                &tempo_attesa);

            passo=p_attesaRum;
        }
        break;

    case p_attesaRum:

```

```

    fz_attendi(tempo_attesa);
    break;

case p_leggoRum:

    res = Leggi(array_read, &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoRum;
    if (stato_lettura == 2)
        passo=p_leggoRum;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_RMSrum;
    }
    break;

case p_RMSrum:

    Calcolo_RMS(array_mic2, &RMS_0, L_buffer_mic);

    // Calcolo dell'incremento delta_A
    FRF_cassa_controllo(f_cont[0], 1.0, phi_cont[0], &Pa2V, &phi_cont_volt);
    delta_A=(max_Vout/100.0)/Pa2V;

    A_cont_init=1/Pa2V;
    A_cont[0]=A_cont_init;

    step=0;
    passo=p_scrivoCont;

    out2[0]=k1;
    out2[1]=A_rum[0];
    out2[2]=phi_rum[0];
    out2[3]=f_rum[0];
    out2[4]=RMS_0;
    out2[5]=soglia_RMS;
    out2[6]=tempo_attesa;

    break;

case p_scrivoCont:

    Esegui_scrittura(n_freq_rum, A_rum, f_rum, phi_rum,
                    n_freq_cont, A_cont, f_cont, phi_cont);

    passo=p_attesaCont;

```

```
        break;

    case p_attesaCont:

        fz_attendi(tempo_attesa);
        break;

    case p_leggoCont:

        Leggi(array_read, &stato_lettura);

        if (stato_lettura == 1)
            passo=p_leggoCont;
        if (stato_lettura == 2)
            passo=p_leggoCont;
        if (stato_lettura == 0)
        {
            for (i=0; i<L_buffer_mic; i++)
                array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
            passo=p_RMScont;
        }
        break;

    case p_RMScont:

        Calcolo_RMS(array_mic2, &RMS_1, L_buffer_mic);

        if (RMS_1<(soglia_RMS))
        {
            condizione_RMS=1;
            passo=p_verificaRMS;
        }
        else
        {
            if (itero >15 & RMS_1<0.2*RMS_0)
            {
                condizione_RMS=1;
                passo=p_verificaRMS;
            }
            else
                passo=p_scrivoContDelta;
        }
        break;

    case p_scrivoContDelta:

        if (step==0)
        {
            A_cont_temp[0]=A_cont[0];
```

```

        phi_cont_temp[0]=phi_cont[0]+delta_phi;
    }
    else
    {
        A_cont_temp[0]=A_cont[0]+delta_A;
        phi_cont_temp[0]=phi_cont[0];
    }

    Esegui_scrivtura (n_freq_rum, A_rum, f_rum, phi_rum,
                    n_freq_cont, A_cont_temp, f_cont, phi_cont_temp);

    passo=p_attesaContDelta;

    break;

case p_attesaContDelta:

    fz_attendi(tempo_attesa);
    break;

case p_leggoContDelta:

    Leggi(array_read, &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoContDelta;
    if (stato_lettura == 2)
        passo=p_leggoContDelta;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_RMScontDelta;
    }
    break;

case p_RMScontDelta:

    if (step==0)
        Calcolo_RMS(array_mic2, &RMS.2, L_buffer_mic);

    else
        Calcolo_RMS(array_mic2, &RMS.3, L_buffer_mic);

    passo=p_gradiente;

    break;

case p_gradiente:

```



```
if (step==0)
{
    //Calcolo del gradiente
    grad_phi=(RMS_2-RMS_1)/delta_phi;
    step=1;
    passo=p_scrivoContDelta;
}
else
{
    //Calcolo del gradiente
    grad_A=(RMS_3-RMS_1)/delta_A;

    if ( (fabs(k1*grad_phi)) > sat_phi )
        cond_phi=1;
    else
        cond_phi=0;

    if ( (fabs(k1*grad_A)) > sat_A/Pa2V )
        cond_A=1;
    else
        cond_A=0;

    k2_phi=sat_phi/fabs(grad_phi);
    k2_A=(sat_A/Pa2V)/fabs(grad_A);

    if ( (!cond_A) & cond_phi)
        k2=k2_phi;

    if ( cond_A & (!cond_phi) )
        k2=k2_A;

    if ( cond_A & cond_phi)
    {
        if ( k2_phi >= k2_A )
            k2=k2_A;
        else
            k2=k2_phi;
    }

    if ( (!cond_A) & (!cond_phi) )
        k2=k1;

    A_cont[0]=A_cont[0]-k2*grad_A;
    phi_cont[0]=phi_cont[0]-k2*grad_phi;
}
```

```

////////////////////////////////////// METODO 2
if ( (grad_phi*grad_phi_old)<0.0 )
{
    salto_phi++;
    if ( salto_phi >2)
    {
        salto_phi=0;
        dimezza_k1=1;
    }
}

if ( (grad_A*grad_A_old)<0.0 )
{
    salto_A++;
    if ( salto_A >2)
    {
        salto_A=0;
        dimezza_k1=1;
    }
}

if ( dimezza_k1==1 )
    k1=k1/2;

if ( f_rum[0]>700)
{
    if ( k1<0.05)
        k1=0.05;
}
else
{
    if ( k1<0.005)
        k1=0.005;
}

grad_A_old=grad_A;
grad_phi_old=grad_phi;
////////////////////////////////////// FINE METODO 2

passo=p.verificaRMS;
step=0;

if ( fabs(A_cont[0])>(3.0/Pa2V))
    A_cont[0]=3.0*Pa2V*( fabs(A_cont[0])/ A_cont[0] );

/* ----- Pubblicazione delle variabili -----*/
itero++;
out1[0]=itero;
out1[1]=A_cont[0];

```

```

        out1[2]=phi_cont[0];
        out1[3]=RMS_1;
        out1[4]=A_cont[0]+delta_A;
        out1[5]=RMS_2;
        out1[6]=phi_cont[0]+delta_phi;
        out1[7]=RMS_3;
        out1[8]=grad_A;
        out1[9]=grad_phi;
        out1[10]=k2;
    }
    A_cont_old=A_cont[0];
    phi_cont_old=phi_cont[0];

    break;

case p_verificaRMS:

    if (itero>150 || condizione_RMS==1)
    {
        //prova=(double) iterazione*tc/1000;
        passo=p_STOP;

        /* ----- Pubblicazione delle variabili ----- */
        itero++;
        out1[0]=itero;
        out1[1]=A_cont[0];
        out1[2]=phi_cont[0];
        out1[3]=RMS_1;
        out1[4]=A_cont[0]+delta_A;
        out1[5]=RMS_2;
        out1[6]=phi_cont[0]+delta_phi;
        out1[7]=RMS_3;
        out1[8]=grad_A;
        out1[9]=grad_phi;
        out1[10]=k2;
        itero--;

        break;
    }
    else
        passo=p_scrivoCont;

    break;
}

if ( passo==p_STOP )
{
    passo_old=1000;

```

```

if ( flag_controllo_old != controllo_ON || flag_rumore_old != rumore_ON)
{
    if ( !controllo_ON & rumore_ON)
    {
        resetta(A_cont_temp, n_freq_cont);
        Esegui-scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                        n_freq_cont, A_cont_temp, f_cont, phi_cont);
    }
    if ( controllo_ON & rumore_ON)
    {
        Esegui-scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                        n_freq_cont, A_cont, f_cont, phi_cont);
    }
    if ( !rumore_ON & controllo_ON)
    {
        resetta(A_rum_temp, n_freq_rum);
        Esegui-scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                        n_freq_cont, A_cont, f_cont, phi_cont);
    }
    if ( !rumore_ON & !controllo_ON)
    {
        resetta(A_rum_temp, n_freq_rum);
        resetta(A_cont_temp, n_freq_cont);
        Esegui-scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                        n_freq_cont, A_cont_temp, f_cont, phi_cont);
    }
}
flag_controllo_old=controllo_ON;
flag_rumore_old=rumore_ON;
}
return passo_old;
}

/*-----*/
/* Funzione 3 - Pulizia */
/*-----*/
int Pulizia (void)
{
    int res=0;
    int i=0;

    double n_freq_rum;
    double f_rum[10];
    double phi_rum[10];

    n_freq_rum=n_freq_cont;

    for (i=0; i<n_freq_rum; i++)
    {

```

```

        f_rum [ i ] = 1000.0;
        phi_rum [ i ] = 0.0;
    }

    resetta ( A_rum_temp , n_freq_rum );
    resetta ( A_cont_temp , n_freq_cont );

    Eseguo_scrittura ( n_freq_rum , A_rum_temp , f_rum , phi_rum , n_freq_cont ,
                      A_cont_temp , f_cont , phi_cont );

    if ( hMyTask != -1 )
    {
        res = DAQmxClearTask ( hMyTask );
        hMyTask = -1;
    }
    else
        res = 1;

    if ( res == 0 )
    {
        if ( kMyTask != -1 )
        {
            res = DAQmxClearTask ( kMyTask );
            kMyTask = -1;
        }
    }
    else
        res = 2;

    passo = 0;
    return res;
}

```

### File “cvidll.h”

```

#define PI 3.141593
#define TASK_A01_NAME "TASK_OUT"
#define TASK_A1_NAME "TASK_IN"

/*-----*/
/* Prototipi di funzione */
/*-----*/
int Inizializzazione ( void );

double Ciclo_lento ( int n_freq_rum , double *A_rum , double *f_rum ,
                    double *phi_rum , double *A_cont , double *phi_cont ,
                    double *out1 , double *out2 , int stop_ciclo , int controllo_ON );

int Pulizia ( void );

```

## File "func\_lib.h"

```

/*-----*/
/* INCLUDE */
/*-----*/
#include <NIDAQmx.h>
#include <analysis.h>
#include <ansi_c.h>

/*-----*/
/* Prototipi di funzione */
/*-----*/
int Leggi(double *array_mic, int *stato_lettura);

double Esegui_scrittura (int n_freq_rum, double *A_rum, double *f_rum,
                        double *phi_rum, int n_freq_cont, double *A_cont, double *f_cont,
                        double *phi_cont);

int Scrivi (double L_buffer, double *segnale);

void calcola_periodo (double *frequenze, int n_freq, double *L_buffer);

void rat (double numero, double *N, double *D);

void MCD(double *mcd, double x, double y);

void resetta (double *var, int n);

void fz_attendi(double tempo_attesa);

void Setting_parametri(double f_rum, double *sat_A, double *sat_phi,
                      double *k1, double *k1_fin, double *tempo_attesa);

void Calcolo_costanti_algoritmo(double *k1, double *grad_phi, double *sat_phi,
                                double *grad_A, double *sat_A, double *Pa2V, double *k2, int *salto_phi,
                                int *salto_A, double *f_rum, int j);

void FRF_cassa_controllo(double freq, double A_Pa, double phi_Pa, double *A_volt,
                        double *phi_volt);

void Calcolo_RMS(double *array, double *valore_RMS, double N);

void Calcolo_Ampiezza (double *array, double *valore_Ampiezza, double N,
                      double frequenza);

```

```
/*-----*/
/* Definizione variabili */
/*-----*/
extern TaskHandle hMyTask;
extern TaskHandle kMyTask;

extern double Fc_in;
extern double Fc_out;

extern double *A;
extern double *f;
extern double *phi;

extern double L_buffer;
extern double memoria_alloc;
extern double *segnale;

extern int NumChannels_in;
extern double L_buffer_mic;
extern double *spettro_real;
extern double *spettro_imag;
extern double *modulo;
extern int32 samples_read;

extern int n_freq_old;
extern uInt32 campioni_acq;

extern double tc;
extern double aspetta;

extern double grad_phi_old[10];
extern double grad_A_old[10];

extern double max_Vout;
extern double min_Vout;

extern double sat_phi[10];
extern double sat_A[10];

/*-----*/
/* Funzione - Leggi */
/*-----*/
int Leggi(double *array_mic, int *stato_lettura)
{
    int res=0;

    if (*stato_lettura==0)
    {
        //inizio lettura
    }
}
```

```

    DAQmxStartTask(kMyTask);
    *stato_lettura=1;
}
else
{
    DAQmxGetReadAvailSampPerChan(kMyTask, &campioni_acq);
    if (campioni_acq==L_buffer_mic) //acquisisco per x secondi
    {
        //carico i campioni in memoria
        DAQmxReadAnalogF64 (kMyTask, L_buffer_mic, 10.0, DAQmx_Val_GroupByChannel,
            array_mic, (uInt32)(NumChannels_in*L_buffer_mic), &samples_read, NULL);
        DAQmxStopTask(kMyTask);
        *stato_lettura=0;
    }
    else
    {
        //sto leggendo
        *stato_lettura=2;
    }
}
return res;
}

/*-----*/
/* Funzione - Eseguo_scrittura */
/*-----*/
double Eseguo_scrittura (int n_freq_rum, double *A_rum, double *f_rum,
    double *phi_rum, int n_freq_cont, double *A_cont, double *f_cont,
    double *phi_cont)
{
    int res=0;
    int i=0;
    int j=0;
    double temp_rum=0;
    double temp_cont=0;
    double temp=0;

    double A_temp;
    double phi_temp;

    double n_freq;

    n_freq=n_freq_cont+n_freq_rum;

    if (n_freq>n_freq_old)
    {
        A=(double *) realloc (A, n_freq*sizeof(double));
        f=(double *) realloc (f, n_freq*sizeof(double));
        phi=(double *) realloc (phi, n_freq*sizeof(double));
    }

```



```

    n_freq_old=n_freq;
}

for (j=0; j<n_freq_rum; j++)
{
    A[j]=A_rum[j];
    f[j]=f_rum[j];
    phi[j]=phi_rum[j];
}

for (j=0; j<n_freq_cont; j++)
{
    //funzione di trasferimento della cassa di controllo (Pa 2 V)
    FRF_cassa_controllo(f_cont[j], A_cont[j], phi_cont[j], &A_temp, &phi_temp);

    A[j+n_freq_rum]=A_temp;
    f[j+n_freq_rum]=f_cont[j];
    //phi[j+n_freq_rum]=phi_temp;

    //A[j+n_freq_rum]=A_cont[j];
    phi[j+n_freq_rum]=phi_cont[j];
}

if (n_freq==1)
    L_buffer=floor(1/f[0]*Fc_out);
else
    calcola_periodo(f, n_freq, &L_buffer);

if (2*(L_buffer)>memoria_alloc)
{
    memoria_alloc=2*L_buffer;
    segnale=(double *)realloc(segnale, memoria_alloc*sizeof(double));
}

for (i=0; i<L_buffer; i++)
{
    for (j=0; j<n_freq; j++)
    {
        if (j<n_freq_rum)
            temp_rum=temp_rum+A[j]*sin(2*PI*f[j]*i/Fc_out+phi[j]);
        else
            temp_cont=temp_cont+A[j]*sin(2*PI*f[j]*i/Fc_out+phi[j]);
    }
    segnale[i]=temp_rum;
    segnale[i+(int)L_buffer]=temp_cont;
    temp_rum=0;
    temp_cont=0;
}

```

```

    res = Scrivi(L_buffer , segnale);
    return L_buffer;
}

/*-----*/
/* Funzione - Ciclo per le uscite */
/*-----*/
int Scrivi (double L_buffer , double *segnale)
{
    int32 samples_write=0;
    int res=0;

    if (res==0)
        res = DAQmxStopTask(hMyTask);
    else
        res=1;

    if (res==0)
        res = DAQmxWriteAnalogF64(hMyTask,(int32)L_buffer,0,10.0,
            DAQmx_Val_GroupByChannel,segnale,&samples_write,NULL);
    else
        res=2;

    DAQmxStartTask(hMyTask);

    return res;
}

/*-----*/
/* Funzione - calcola_periodo */
/*-----*/
void calcola_periodo (double *frequenze , int n_freq , double *L_buffer)
{
    int i;
    int j;
    double *K;
    double *rapporti;
    double *n_temp;
    double *n1_temp;
    double *n_temp2;
    double n1_temp2=1;
    double mcd;

    K=(double *) malloc(n_freq*sizeof(double));
    n_temp=(double *) malloc((n_freq-1)*sizeof(double));
    n1_temp=(double *) malloc((n_freq-1)*sizeof(double));
    n_temp2=(double *) malloc((n_freq-1)*sizeof(double));
    rapporti=(double *) malloc((n_freq-1)*sizeof(double));

```

```

for (i=0; i<n_freq; i++)
    *(K+i)=(Fc_out/(*(frequenze+i)));

for (i=0; i<(n_freq-1); i++)
{
    *(rapporti+i)=*K/(*(K+i+1));
    rat(*(rapporti+i), (n_temp+i), (n1_temp+i));
}

for (i=0; i<(n_freq-1); i++)
{
    *(n_temp2+i)=1;
    n1_temp2=*(n1_temp+i)*n1_temp2;
}

for (i=0; i<(n_freq-1); i++)
{
    for (j=0; j<(n_freq-1); j++)
    {
        if (j==i)
            *(n_temp2+i)=*(n_temp2+i)*(*(n_temp+j));
        else
            *(n_temp2+i)=*(n_temp2+i)*(*(n1_temp+j));
    }
}

for (i=0; i<(n_freq-1); i++)
{
    if (i==0)
        MCD(&mcd, n1_temp2, *(n_temp2+i));
    else
        MCD(&mcd, mcd, *(n_temp2+i));
}
n1_temp2=n1_temp2/mcd;

*L_buffer=floor((1/(*(frequenze)))*n1_temp2*Fc_out);

free(K);
free(rapporti);
free(n_temp);
free(n1_temp);
free(n_temp2);
}

```

```

/*-----*/
/* Funzione - rat */
/*-----*/
void rat (double numero, double *N, double *D)
{
    double C[2][2]={{1,0},{0,1}};
    double a;
    double b;
    double d;
    double e;
    double segno;
    double toll=0.0000001*fabs(numero);
    double resto=numero;
    int k=1;

    while (fabs(resto)>toll)
    {
        if (k==1)
        {
            a=resto;
            k=0;
        }
        else
        {
            a=1/resto;
        }
        b=floor(a);
        resto=a-b;
        d=C[0][0]*b+C[0][1];
        e=C[1][0]*b+C[1][1];
        C[0][1]=C[0][0];
        C[1][1]=C[1][0];
        C[0][0]=d;
        C[1][0]=e;
    }
    segno=fabs(C[0][0])/C[0][0];
    *N=C[0][0]/segno;
    *D=fabs(C[1][0]);
}

/*-----*/
/* Funzione - MCD */
/*-----*/
void MCD(double *mcd, double x, double y)
{
    while(x!=y)
    {
        if (x<y)
            y=y-x;
    }
}

```

```

        else
            x=x-y;
    }
    *mcd=x;
}

/*-----*/
/* Funzione - resetta */
/*-----*/
void resetta (double *var, int n)
{
    int j;
    for (j=0; j<n; j++)
        var[j]=0;
}

/*-----*/
/* Funzione - fz_attendi */
/*-----*/
void fz_attendi(double tempo_attesa)
{
    if (aspetta<tempo_attesa)
        aspetta=aspetta+tc;
    else
    {
        aspetta=tc;
        passo++;
    }
}

/*-----*/
/* Funzione - Setting_parametri */
/*-----*/
void Setting_parametri(double f_rum, double *sat_A, double *sat_phi,
    double *k1, double *k1_fin, double *tempo_attesa)
{
    if (f_rum<800)
    {
        *sat_A=(max_Vout/20.0);
        *sat_phi=((10/180.0)*PI);
        *k1=0.5;
    }
    if (f_rum>=800 & f_rum<1350)
    {
        *sat_A=(max_Vout/10.0);
        *sat_phi=((30/180.0)*PI);
        *k1=3;
    }
    if (f_rum>=1350)

```

```

{
    *sat_A=(max_Vout/10.0);
    *sat_phi=((30/180.0)*PI);
    *k1=3;
}

if (f_rum<500)
    *tempo_attesa=500;
else
{
    if (f_rum>=500 & f_rum<1000)
        *tempo_attesa=400;
    else
        *tempo_attesa=200;
}

if (f_rum<700)
    *k1_fin=0.005;
if (f_rum>=700 & f_rum<=1300)
    *k1_fin=0.05;
if (f_rum>1300)
    *k1_fin=0.1;
}

/*-----*/
/* Funzione - Calcolo_costanti_algoritmo */
/*-----*/
void Calcolo_costanti_algoritmo(double *k1, double *grad_phi, double *sat_phi,
    double *grad_A, double *sat_A, double *Pa2V, double *k2,
    int *salto_phi, int *salto_A, double *f_rum, int j)
{
    int cond_phi;
    int cond_A;
    double k2_phi;
    double k2_A;
    int dimezza_k1[10]={0,0,0,0,0,0,0,0,0,0};

    /* ----- Verifica dei passi effettuati dall'algoritmo ----- */
    if ( (fabs(k1[j]*grad_phi[j])) > sat_phi[j] )
        cond_phi=1;
    else
        cond_phi=0;
    if ( (fabs(k1[j]*grad_A[j])) > sat_A[j]/Pa2V[j] )
        cond_A=1;
    else
        cond_A=0;

    k2_phi=sat_phi[j]/fabs(grad_phi[j]);
    k2_A=(sat_A[j]/Pa2V[j])/fabs(grad_A[j]);

```

```
if ( (!cond_A) & cond_phi)
    k2[j]=k2_phi;
if ( cond_A & (!cond_phi) )
    k2[j]=k2_A;
if ( cond_A & cond_phi)
{
    if ( k2_phi >= k2_A )
        k2[j]=k2_A;
    else
        k2[j]=k2_phi;
}
if ( (!cond_A) & (!cond_phi) )
    k2[j]=k1[j];

/* ----- Calcolo della costante k1 -----*/
if ( (grad_phi[j]*grad_phi_old[j])<0.0 )
{
    salto_phi[j]++;
    if (salto_phi[j]>2)
    {
        salto_phi[j]=0;
        dimezza_k1[j]=1;
    }
}

if ( (grad_A[j]*grad_A_old[j])<0.0 )
{
    salto_A[j]++;
    if (salto_A[j]>2)
    {
        salto_A[j]=0;
        dimezza_k1[j]=1;
    }
}

if ( dimezza_k1[j]==1 )
    k1[j]=k1[j]/2;

if (k1[j]<k1_fin[j])
    k1[j]=k1_fin[j];
}
```

```
/*-----*/
/* Funzione - FRF cassa controllo */
/*-----*/
void FRF_cassa_controllo(double freq, double A_Pa, double phi_Pa,
                        double *A_volt, double *phi_volt)
{
    double intervalli_freq[37]={
        200.000000,
        250.000000,
        300.000000,
        ...
        1900.000000,
        1950.000000,
        2000.000000};

    double coef_mod_1[37]={
        -0.009215,
        -0.000332,
        -0.006218,
        ...
        -0.002722,
        0.005436,
        -0.002760};

    double coef_mod_2[37]={
        1.839110,
        1.378340,
        1.361730,
        ...
        1.945850,
        1.809770,
        2.081570};

    double coef_fase_1[37]={
        -0.017682,
        -0.005388,
        -0.004846,
        ...
        0.001533,
        -0.005531,
        -0.001249};

    double coef_fase_2[37]={
        1.577743,
        0.693637,
        0.424239,
        ...
        -3.848259,
        -3.771587,
```



```

        -4.048134};

    int indice=0;
    double FRFcassa_mod;
    double FRFcassa_fase;

    // non consideriamo l'effetto della distanza sulla FdT
    double dist_FRF=1;
    double dist_mic=1;

    while (intervalli_freq[indice]<freq)
    {
        indice++;
    }
    indice--;

    FRFcassa_mod=coef_mod_1[indice]*(freq-intervalli_freq[indice])
                +coef_mod_2[indice];
    FRFcassa_fase=coef_fase_1[indice]*(freq-intervalli_freq[indice])
                +coef_fase_2[indice];

    *A_volt=A_Pa/FRFcassa_mod*dist_mic/dist_FRF;
    *phi_volt=phi_Pa-FRFcassa_fase;
}

/*-----*/
/* Funzione - Calcolo_RMS */
/*-----*/
void Calcolo_RMS (double *array, double *valore_RMS, double N)
{
    int i;
    double valore_RMS_temp=0;

    for (i=0; i<N; i++)
        valore_RMS_temp=valore_RMS_temp+(array[i]*array[i])/N;

    *valore_RMS=sqrt(valore_RMS_temp);
}

/*-----*/
/* Funzione - Calcolo_Ampiezza */
/*-----*/
void Calcolo_Ampiezza (double *array, double *valore_Ampiezza, double N,
                      double frequenza)
{
    int i;
    int indice;
    double df;
    double mod_temp;

```

```

double mod_MAX=0;

//double valore_RMS_temp=0;

// calcolo lo spettro
for (i=0; i<N; i++)
{
    spettro_real[i]=array[i];
    spettro_imag[i]=0;
}
FFT(spettro_real, spettro_imag, N);

//calcolo dell'indice corrispondente alla frequenza di interesse
df=Fc_in/N;
indice=floor(frequenza/df);

//calcolo del modulo dei punti vicini e ricerca del massimo;
for (i=indice; i<(indice+3); i++)
{
    mod_temp=2/N*sqrt((spettro_real[i])*(spettro_real[i])+
        (spettro_imag[i])*(spettro_imag[i]));
    if (mod_temp>mod_MAX)
        mod_MAX=mod_temp;
}
*valore_Ampiezza=mod_MAX;
}

```

## D.1.2 Analisi dettagliata delle funzioni del codice

Vengono analizzate nel dettaglio le funzioni contenute all'interno del codice per l'esecuzione del sistema di controllo ARMS.

### Funzione “Inizializzazione()”

Con questa funzione viene inizializzato il sistema di controllo, in particolare vengono creati due *task*<sup>1</sup>, uno per le uscite analogiche per il comando della cassa di rumore e di controllo e un *task* per la misurazione della pressione sonora mediante i microfoni. La creazione dei *task* e l'utilizzo di tutte le funzioni strettamente legate all'hardware

---

<sup>1</sup>In LabView un *task* è una raccolta di uno o più canali virtuali, con la possibilità di definire le task temporizzazioni, trigger e altre proprietà. Concettualmente rappresenta una misurazione o una generazione che si desidera eseguire. Tutti i canali in un *task* devono essere dello stessa tipologia, quindi devono essere o canali di ingresso o canali di uscita. Tuttavia un singolo *task* può includere canali di differenti tipi di misura, come un canale di ingresso analogico di temperatura e di un ingresso analogico di tensione.

a disposizione, è possibile grazie ai driver di strumento forniti dalla casa produttrice e contenuti all'interno della libreria "DAQmx.h". Questa libreria contiene moltissime funzioni che permettono innanzitutto di creare, caricare, pulire, iniziare o fermare un task, funzioni per la creazione di differenti tipologie di canali, ad esempio canali per gli ingressi e per le uscite analogiche, ingressi e uscite digitali, oppure canali di ingresso e uscita per i contatori, funzioni per la lettura e scrittura canali, per la temporizzazione e il triggering, per la gestione dei buffer sulle schede, funzioni per il logging e il reporting di errori nel sistema e altre funzioni avanzate.

Per la creazione dei due task è stata utilizzata la funzione *DAQmxCreateTask()*, che permette di assegnare un nome al task a cui far riferimento nella successiva creazione dei canali infatti, una volta definiti i task, vengono configurati i canali per eseguire le operazioni di input e output analogico. La creazione dei canali è effettuata con le funzioni mostrate nel seguito.

#### *Creazione dei canali di uscita*

La funzione *DAQmxCreateAOVoltageChan()* permette di creare i canali di uscita analogica per comandare gli altoparlanti, in particolare è possibile definire il numero e la posizione sulla scheda dei canali di output, definire il tipo di misura (tensione o corrente) ed il fondoscala dei canali per evitare di fornire tensioni superiori a quelle supportate dagli attuatori. Avendo a disposizione delle casse acustiche si è scelto in funzione del datasheet fornito dal costruttore un fondoscala di  $\pm 3 V$  per evitarne il danneggiamento.

Dovendo generare segnali sinusoidali, quindi segnali che si ripetono ad intervalli costanti di tempo, si sfrutta la possibilità, offerta dalle schede di output analogico a disposizione, di rigenerare i segnali per mezzo di un buffer collocato direttamente sulla scheda. La rigenerazione hardware del segnale alleggerisce il carico della CPU per la generazione dei segnali infatti, una volta caricato il buffer si demanda alla scheda il compito di generare in modo continuativo il segnale. Il tempo di aggiornamento del buffer della scheda a disposizione è pari a circa 200 ns. Per impostare l'attributo di scrittura con rigenerazione viene utilizzata la funzione *DAQmxSetWriteAttribute()*.

Oltre a queste due funzioni viene utilizzata la funzione *DAQmxCfgSampClkTiming()* che consente di impostare le caratteristiche di temporizzazione, quindi la frequenza e il tipo di campionamento e la posizione di triggering per l'aggiornamento del segnale. In particolare è stata impostata una frequenza di campionamento pari a 100 kHz,

in modo da ottenere una risoluzione del segnale di  $10 \mu s$  e conseguentemente poter generare i segnali sinusoidali alle frequenze di interesse senza incorrere nel problema di aliasing, secondo il teorema del campionamento di Shannon mostrato nel paragrafo 2.2.3. Il range delle frequenze di interesse per questa tipologia di controllo è pari a  $[200, 2000] Hz$ . È stato scelto inoltre un campionamento continuo in modo che la generazione del segnale continui indefinitamente finché non viene rilevato un segnale esterno, ad esempio viene premuto il tasto “Stop” nell’interfaccia utente mostrata nel paragrafo 3.2.2. La lunghezza del buffer che contiene il segnale da generare viene fissata solo al momento della scrittura sulla scheda quando risultano noti tutti i parametri del segnale.

#### *Creazione dei canali di ingresso*

La funzione `DAQmxCreateAIVoltageChan()` è stata utilizzata per creare i canali di ingresso analogico per la misurazione del segnale in tensione proveniente dalla lettura dei microfoni in particolare, con questa funzione, è stato possibile definire il fondoscala degli ingressi, ossia il range in tensione che ci si aspetta dalla misurazione del sensore, e l’unità di misura sulla lettura dei canali.

La funzione `DAQmxCfgSampClkTiming()` ha consentito invece di configurare le caratteristiche sulla temporizzazione della scheda. Per l’acquisizione è stata scelta una frequenza di campionamento pari a  $50 kHz$ , la risoluzione del segnale letto risulta quindi essere di  $20 \mu s$ . Per il teorema di Shannon è possibile quindi andare ad analizzare senza commettere aliasing tutti i segnali nel range di frequenze di interesse. Diversamente da prima il campionamento viene effettuato su un numero finito di punti: il numero di punti, quindi la lunghezza del buffer di acquisizione, è stata assunta pari a 5000, costante e quindi indipendente dai segnali che vengono generati. Avendo una risoluzione temporale sull’acquisizione del segnale pari a  $20 \mu s$  il tempo di acquisizione è di  $100 ms$  e, considerando l’armonica con periodo maggiore (di frequenza pari a  $300 Hz$ ) si acquisiscono non meno di 30 periodi. Per il calcolo dell’RMS risulterebbe sufficiente un solo periodo del segnale, tuttavia la presenza di rumore e di errori nella misurazione non consentono di effettuare una buona stima di questa grandezza utilizzando un solo periodo. Per ovviare a questo problema si considera pertanto un numero di periodi maggiore.

### Funzione “Ciclo\_lento()”

Questa funzione contiene il nocciolo dell’algoritmo di controllo secondo lo schema mostrato nel diagramma a blocchi in figura 3.49. Si analizzano di seguito le singole fasi del diagramma:

- Si considerino le fasi 2), 5), 9) e 9’) del diagramma a blocchi. La scrittura del segnale da generare o più in generale dei segnali avviene per mezzo della funzione secondaria *Eseguo\_scrittura()* contenuta all’interno del file “func\_lib.h”. Questa riceve in ingresso le caratteristiche del segnale di rumore e del segnale di controllo e provvede alla generazione secondo i seguenti passi:
  - Passando attraverso la funzione di trasferimento della cassa di controllo, determinata mediante l’hardware B&K PULSE e importata successivamente all’interno del codice sorgente, è possibile effettuare la conversione del segnale di controllo dalla pressione sonora richiesta alla tensione da fornire all’altoparlante di controllo. L’utilizzo della funzione di trasferimento permette di svincolare l’algoritmo di controllo, basato sulla ricerca del minimo di una funzione, dal sistema fisico all’interno del quale l’algoritmo deve operare. In questo modo si possono garantire le proprietà di convergenza del metodo attraverso un’opportuna taratura dei guadagni del sistema.
  - Attraverso la funzione *calcola\_periodo()*, che riceve in ingresso il vettore delle frequenze e la sua dimensione, viene calcolata la dimensione del buffer necessario alla generazione dei segnali: si ricorda che il segnale di controllo è un segnale sinusoidale con frequenza pari a quella del segnale di rumore, ossia, la frequenza del segnale di controllo non fa parte dei parametri che vengono fatti variare per la minimizzazione del rumore in ingresso. Per quanto riguarda la tecnica di minimizzazione e la tipologia del segnale di controllo adottato si faccia riferimento al paragrafo 2.3.1. È importante determinare la dimensione corretta del buffer in cui scrivere il segnale poiché una dimensione errata porterebbe alla generazione di un segnale non desiderato infatti, se la dimensione del buffer non fosse esatta, questo non potrebbe contenere un numero intero di periodi del segnale. L’errore si rifletterebbe o in una mancanza di valori da assegnare al segnale di uscita qualora la dimensione del buffer fosse superiore a quella del segnale generato o ad un taglio del segnale prima del tempo richiesto per l’esecuzione di un periodo intero se la

dimensione del buffer è inferiore a quella del segnale; questi fenomeni risultano udibili e sgraditi all'orecchio umano e causano delle sovra-sollecitazioni che vanno evitate per non danneggiare le casse acustiche.

Il calcolo della dimensione del buffer è stato effettuato in base alla soluzione del seguente problema generico: avendo più segnali armonici di periodo  $T_1, T_2, T_3, \dots$ , calcolare il periodo  $T$  tale che i periodi  $T_1, T_2, T_3, \dots$  siano contenuti un numero intero di volte all'interno di  $T$ . Tale problema può essere scritto con il seguente sistema:

$$\begin{cases} T = n_1 T_1 \\ T = n_2 T_2 \\ T = n_3 T_3 \\ \dots \end{cases} \quad (\text{D.1})$$

Le variabili intere incognite  $n_1, n_2, n_3, \dots$  rappresentano il numero di periodi di ciascun segnale riferiti al periodo totale  $T$ . Per ridurre l'ordine del sistema è possibile uguagliare tra loro le equazioni. Una volta trovate le incognite  $n_1, n_2, n_3, \dots$ , esse saranno direttamente riferite alla dimensione del buffer necessario per la scrittura del segnale, pertanto occorre dividere le equazioni per la discretizzazione temporale con cui vengono scritti i dati in uscita.

$$\begin{cases} n_1 K_1 = n_2 K_2 \\ n_1 K_1 = n_3 K_3 \\ \dots \end{cases} \quad (\text{D.2})$$

dove  $K_i = \left\lfloor \frac{T_i}{dt} \right\rfloor = \left\lfloor \frac{f_c}{f_i} \right\rfloor$  è il rapporto tra la frequenza  $f_c$  di aggiornamento del segnale delle uscite e l' $i$ -esima frequenza  $f_i$  delle sinusoidi che compongono il problema. A questo punto le equazioni del sistema possono essere riscritte evidenziando il rapporto tra le variabili intere  $n_i$ :

$$\begin{cases} R1 = \frac{K_1}{K_2} = \frac{n_2}{n_1} \\ R2 = \frac{K_1}{K_3} = \frac{n_3}{n_1} \\ \dots \end{cases} \quad (\text{D.3})$$

Dati i rapporti  $R_i$  occorre trovare i relativi numeratori e denominatori interi. Per trovarli si utilizza l'algoritmo dato dalla approssimazione iterativa seguente:

$$\frac{n}{d} = d_1 + \frac{1}{d_2 + \frac{1}{\left(d_3 + \dots + \frac{1}{d_k}\right)}}. \quad (\text{D.4})$$

Si calcolano iterativamente i coefficienti  $d_k$  fino a che risulta verificata la condizione  $|\frac{n_i}{d_i} - R_i| < \text{toll}$ . Il metodo prende il nome di "RAT" (Rational Fraction Approximation), e applicandolo al problema in esame si ottengono i seguenti risultati:

$$\left\{ \begin{array}{l} \text{RAT}(R1) = \frac{n_{2Temp}}{n_{1Temp1}} \\ \text{RAT}(R2) = \frac{n_{3Temp}}{n_{1Temp2}} \\ \dots \end{array} \right. \quad (\text{D.5})$$

Ovviamente i valori delle incognite non sono ancora quelli definitivi, poiché il denominatore di tutte le frazioni è  $n_1$  ma dalla soluzione dell'algoritmo RAT si ottengono diversi valori, qui definiti come  $n_{1Temp}$ . Occorre dunque eseguire una semplificazione delle frazioni appena trovate, come:

$$\left\{ \begin{array}{l} \frac{n_{2Temp}n_{1Temp2} \dots}{n_{1Temp1}n_{1Temp2} \dots} = \frac{n_{2Temp}n_{1Temp2} \dots}{n_{1Tot}} \\ \frac{n_{3Temp}n_{1Temp1} \dots}{n_{1Temp1}n_{1Temp2} \dots} = \frac{n_{3Temp}n_{1Temp1} \dots}{n_{1Tot}} \\ \dots \end{array} \right. \quad (\text{D.6})$$

Calcolando il massimo comune divisore:

$$z = \text{MCD} \left( \frac{n_{2Temp}n_{1Temp2} \dots}{n_{1Tot}}, \frac{n_{3Temp}n_{1Temp1} \dots}{n_{1Tot}}, \dots \right), \quad (\text{D.7})$$

è possibile ricavare il valore corretto di  $n_1$  come:

$$n_{1def} = \frac{n_{1Tot}}{z}. \quad (\text{D.8})$$

Dal valore di  $n_{1def}$  è possibile calcolare la lunghezza del buffer attraverso la seguente relazione:

$$L_{buffer} = \left\lfloor \frac{n_{1def} f_c}{f_1} \right\rfloor. \quad (\text{D.9})$$

- Nota la dimensione del buffer, tale da contenere un numero esatto di periodi del segnale, e noti i segnali da generare, grazie alla funzione *Scrivi()* vengono inviate le informazioni al controllore RT, quindi alla scheda delle uscite che provvede alla generazione del segnale. La funzione *Scrivi()* è composta di tre parti:
  1. La prima è realizzata dalla funzione *DAQmxStopTask()* che ferma il task delle uscite avviato in precedenza, questo step è obbligatorio poiché, a causa dell'attributo di rigenerazione impostato nella funzione *Inizializzazione()* e più precisamente nel task delle uscite, la scheda diventa un'entità completamente svincolata dal controllore e risulta quindi necessario fermare la generazione precedente prima di procedere con l'aggiornamento del segnale presente nel buffer. Solo con una generazione punto-punto non è necessario fermare il task prima di procedere all'aggiornamento delle uscite, ma come già esposto in precedenza, con i driver a disposizione, l'unico modo possibile per generare i segnali con il range di frequenza desiderato è quello di sfruttare la rigenerazione del segnale all'interno della scheda.
  2. La seconda parte è realizzata grazie alla funzione *DAQmxWriteAnalogF64()* che riceve in ingresso il segnale da generare e la dimensione del buffer, calcolata come esposto in precedenza. Il segnale verrà allocato dalla funzione all'interno del buffer della scheda e quest'ultima provvederà ad aggiornare con rigenerazione le uscite della stessa. La funzione riceve in ingresso ulteriori parametri come il layout dei dati del segnale: parametro necessario per l'interpretazione corretta da parte della scheda del segnale stesso.
  3. Terminata l'impostazione della scheda secondo i due passi precedenti è possibile avviare nuovamente il task delle uscite. L'avvio è effettuato per mezzo della funzione *DAQmxStartTask()*. Appena viene chiamata la funzione, la scheda inizia ad effettuare la generazione del segnale.
- Nelle fasi 3), 6), 10) e 10') dello schema a blocchi in figura 3.49, viene chiamata la funzione *Leggi()* che provvede ad effettuare l'acquisizione del segnale in pressione, grazie alla lettura dei microfoni. Al termine della fase di scrittura, ossia dopo la chiamata della funzione *Esegui-scrittura()*, è necessario garantire prima di iniziare l'acquisizione il termine dei fenomeni transitori (par. 3.4.1) che potrebbero



causare delle misurazioni errate. La funzione che realizza l'attesa necessaria al termine di questi fenomeni è *fz\_attendi()* che riceve in ingresso il tempo di attesa. Il tempo di attesa viene definito in funzione della frequenza del segnale di rumore  $f_{rum}$ . Dopo aver effettuato una campagna di prove sperimentali, si è fissato il valore del tempo di attesa secondo il seguente schema:

$$\begin{aligned} T_{ATTESA} &= 500 \text{ ms}, & f_{rum} < 500 \text{ Hz}; \\ T_{ATTESA} &= 400 \text{ ms}, & 500 \leq f_{rum} < 1000 \text{ Hz}; \\ T_{ATTESA} &= 200 \text{ ms}, & f_{rum} \geq 1000 \text{ Hz}. \end{aligned}$$

L'impostazione del tempo di attesa è effettuata dalla funzione *Setting\_parametri()*, che riceve in ingresso il valore della frequenza del segnale di rumore.

Si analizza ora in dettaglio la funzione per la lettura degli ingressi analogici, essa è composta da tre parti fondamentali:

- Nella prima parte viene avviato, grazie alla funzione *DAQmxStartTask()*, il task degli ingressi analogici creato precedentemente nella funzione *Inizializzazione()*. L'avvio di questo task fa iniziare il processo di misurazione.
  - La seconda parte prevede un ciclo che permette di acquisire il segnale misurato. La prima funzione che viene utilizzata in questo step è *DAQmxGetReadAvailSampPerChan()* che permette di conoscere il numero di campioni disponibili all'interno del buffer di lettura; può capitare infatti che la scheda non contenga un numero di campioni pari alla dimensione desiderata. Per ovviare a questo problema, si rileva il valore di campioni nel buffer e, quando questo risulta pari alla dimensione desiderata, si procede all'acquisizione per mezzo della funzione *DAQmxReadAnalogF64()*. Quest'ultima funzione riceve in ingresso il numero di canali di input, che nel nostro caso è pari a 1, e la dimensione del buffer che si vuole acquisire, mentre in uscita restituisce un vettore contenente il segnale letto dal microfono, questo viene moltiplicato per un fattore che permette la conversione da Volt a Pascal.
  - Una volta effettuata la misurazione viene fermato il task di lettura mediante la funzione *DAQmxStopTask()*.
- Nelle fasi 4), 7), 11) e 11') viene effettuato il calcolo del valore quadratico medio (RMS) sul segnale acquisito nelle fasi precedenti. Il calcolo dell'RMS viene effet-

tuato per mezzo della funzione *Calcolo\_RMS()*: questa funzione riceve in ingresso l'array  $a_{mic}$ , in cui è memorizzato il segnale appena acquisito, e la sua dimensione  $N$  e in uscita fornisce il valore dell'RMS. Il calcolo è molto semplice e viene effettuato con la relazione 3.37.

- Nella fase 8) viene effettuato il confronto tra il valore attuale di RMS raggiunto e il valore della soglia di rumore limite ammissibile (fig. 3.50), che è stata fissata precedentemente. Se il valore di RMS attuale è superiore alla soglia, l'algoritmo prosegue alla fase 9) con i nuovi valori dei parametri di controllo, mentre se l'RMS raggiunto è inferiore o uguale al valore di soglia l'algoritmo termina, giungendo alla fase 16). Questi valori sono quelli che consentono di abbattere il rumore generato all'ingresso del condotto fino al valore della soglia di rumore desiderata. Oltre alla soglia è stato fissato inoltre un numero di iterazioni massimo che consente di terminare l'algoritmo qualora questo non abbia ancora raggiunto il valore di soglia desiderato.
- Nelle fasi 9) e 9') dello schema a blocchi viene fornito un incremento alle variabili  $A_{cont}^k$  e  $\phi_{cont}^k$  necessario al calcolo del gradiente mediante la formulazione alle differenze finite mostrata nelle relazioni 2.47 e 2.48. Supposto di aver raggiunto la configurazione in cui i valori delle variabili di controllo sono  $\bar{A}_{cont}^k$  e  $\bar{\phi}_{cont}^k$ , si applica l'incremento a queste variabili in modo differito, ossia, inizialmente viene applicato l'incremento alla variabile  $\phi_{cont}^k$  ed effettuate le fasi da 9) a 13) e successivamente si applica l'incremento alla variabile  $A_{cont}^k$  rispetto al medesimo punto iniziale  $\bar{A}_{cont}^k$  e  $\bar{\phi}_{cont}^k$  ed effettuate le fasi da 9') a 13'), secondo il seguente schema:

- Incremento della variabile  $\phi_{cont}^k$ . I parametri del segnale di controllo con i quali verrà effettuato il calcolo del gradiente rispetto alla variazione del parametro  $\phi_{cont}^k$  sono:

$$A_{cont}^{temp} = \bar{A}_{cont}^k; \quad (D.10)$$

$$\phi_{cont}^{temp} = \bar{\phi}_{cont}^k + \Delta\phi_{cont}. \quad (D.11)$$

- Valutazione del gradiente sulla variabile  $\phi_{cont}^k$ , secondo la relazione 2.47.

- Incremento della variabile  $A_{cont}^k$ . I parametri del segnale di controllo con i quali verrà effettuato il calcolo del gradiente rispetto alla variazione del parametro  $A_{cont}^k$  sono:

$$A_{cont}^{temp} = \bar{A}_{cont}^k + \Delta A_{cont}; \quad (D.12)$$

$$\phi_{cont}^{temp} = \bar{\phi}_{cont}^k. \quad (D.13)$$

- Valutazione del gradiente rispetto alla variazione della sola variabile  $A_{cont}^k$ , secondo la relazione 2.48.
- Nelle fasi 13) e 13') viene implementato il calcolo del gradiente. A seconda della variabile di controllo che ha subito la variazione viene effettuato il calcolo del gradiente utilizzando le relazioni alle differenze finite esposte precedentemente.
- Nella fase 14) dello schema a blocchi viene rivalutato il passo  $G_1^k$  da fornire per la valutazione dei nuovi parametri di controllo. Le relazioni matematiche per il calcolo del nuovo passo sono quelle riportate al paragrafo 3.4.1 al punto 14). Dalle osservazioni sperimentali, per garantire la convergenza del metodo di ricerca del minimo, si sono definite le seguenti variazioni massime dei parametri e il valore del passo iniziale  $G_1^{k=0}$ :

$$A_{SAT} = \frac{V_{max}}{20}, \quad \phi_{SAT} = 10^\circ, \quad G_1^{k=0} = 0.5, \quad f_{rum} < 800 \text{ Hz};$$

$$A_{SAT} = \frac{V_{max}}{10}, \quad \phi_{SAT} = 30^\circ, \quad G_1^{k=0} = 3, \quad f_{rum} \geq 800 \text{ Hz}.$$

dove  $V_{max} = 3 \text{ V}$  è il fondoscala massimo delle uscite della scheda di output.

Il valore di soglia inferiore per il passo  $G_2^k$  è stato assunto in funzione della frequenza pari a:

$$G_{1,MIN}^k = 0.005, \quad f_{rum} \leq 700 \text{ Hz};$$

$$G_{1,MIN}^k = 0.05, \quad f_{rum} \geq 700 \text{ Hz}.$$

L'impostazione di questi parametri è effettuata dalla funzione *Setting\_parametri()*, che riceve in ingresso il valore della frequenza del segnale di rumore.

- Nella fase 15) vengono calcolati i valori dei nuovi parametri di controllo secondo le relazioni 3.45 e 3.46.

### Funzione “Pulizia()”

*Pulizia()*: questa funzione permette di terminare il sistema di controllo con le dovute precauzioni. A seguito dello spegnimento del sistema di controllo è necessario garantire la chiusura e l'eliminazione dei task sul controllore del sistema real-time infatti, qualora questi non vengano eliminati, è possibile interferiscono con un nuovo avvio del sistema di controllo e questo potrebbe causare un arresto improvviso del sistema stesso. La funzione *Pulizia()* non fa altro che richiamare, per ogni task generato nell'inizializzazione del sistema di controllo, la funzione *DAQmxClearTask()*, il cui compito è quello di fermare il task e liberare tutte le risorse occupate dallo stesso.

## D.2 Codice sorgente per l'algoritmo AS

Il codice sorgente è stato creato con l'ausilio dei medesimi software utilizzati per l'implementazione dell'algoritmo ARMS. A partire dal codice creato in precedenza e di cui si è mantenuta la struttura si è andati a modificare le sole funzioni per il calcolo della grandezza da minimizzare.

### D.2.1 Struttura del codice

Vengono di seguito riportati i file dei codici sorgenti per il controllo del rumore mediante minimizzazione del picco spettrale di un rumore mono-armonico rilevato dal microfono di errore:

1. *cvidll.c*
2. *cvidll.h*
3. *func\_lib.h*

#### File “cvidll.c”

```

/*-----*/
/* Include files */
/*-----*/
#include <windows.h>
#include <analysis.h>
#include <cvirte.h>
#include "cvidll.h"
#include <NDAQmx.h>
#include <utility.h>

```

```

#include <ansi.c.h>
#include <math.h>

/*-----*/
/* Definizione e inizializzazione variabili globali */
/*-----*/
TaskHandle hMyTask=-1;           //Task di uscite analogiche
double Fc_out=100000;           //frequenza di campionamento delle uscite
double *A=NULL;                 //array ampiezze rumore e controllo
double *f=NULL;                 //array frequenze rumore e controllo
double *phi=NULL;               //array fasi rumore e controllo
double *segnale=NULL;           //array segnale da scrivere in uscita
double L_buffer=500;            //numero punti del buffer di scrittura
double memoria_alloc=100000;    //numero punti del buffer di scrittura inizialmente assegnati

TaskHandle kMyTask=-1;          //Task di ingressi analogici
double Fc_in =50000;            //frequenza di campionamento della lettura degli ingressi
double scala_mic=3.16456;       //Scala conversione Volt -> Pascal
int NumChannels_in=2;           //numero di canali in ingresso
double L_buffer_mic=4096;       //numero di punti del buffer di lettura
double *array_mic1=NULL;        //puntatore all'array di lettura del microfono 1
double *array_mic2=NULL;        //puntatore all'array di lettura del microfono 2
double *array_read=NULL;        //puntatore all'array contenente i segnali dei microfoni

int32 samples_read;             //campioni letti , caricati dalla memoria locale in ram

int stato_lettura=0;            //flag dello stato della lettura
int passo;                       //contatore dei passi dell'algoritmo
int passo_old=100;              //contatore dei passi dell'algoritmo (passo precedente)

double A_cont[10];              //array ampiezze controllo
double f_cont[10];              //array frequenze controllo
double phi_cont[10];            //array fasi controllo
double A_cont_old=0.0;
double phi_cont_old=0.0;

double A_cont_temp[10];         //array ampiezze controllo
double phi_cont_temp[10];       //array fasi controllo
double A_rum_temp[10];

int n_freq_cont=1;              //numero di frequenze del controllo
int n_freq_old=2;               //numero di frequenze di rumore + controllo

double *spettro_real=NULL;      //puntatore array parte reale dello spettro segnale
double *spettro_imag=NULL;      //puntatore array parte immaginaria dello spettro segnale
double *modulo=NULL;            //puntatore array modulo dello spettro segnale
int32 samples_read;             //campioni letti , caricati dalla memoria locale in ram

uint32 campioni_acq;            //campioni acquisiti finora dalla scheda

```

```

double AMP_0;           //modulo dell'ampiezza della frequenza del rumore
double AMP_1;           //ampiezza rumore + controllo
double AMP_2;           //ampiezza rumore + controllo con variazione di fase
double AMP_3;           //ampiezza rumore + controllo con variazione di ampiezza

int condizione_AMP=0;   //flag di verifica della condizione dell'ampiezza
int step=0;             //flag operazioni relative al calcolo dei gradienti
double grad_phi=0;     //gradiente della fase
double grad_A=0;       //gradiente dell'ampiezza
double k1;              //costante di moltiplicazione dei gradienti iniziale
double k1_init;         //valore iniziale della cost. di molt. dei gradienti
double k1_fin;          //valore finale della cost. di molt. dei gradienti
double k2=0;           //costante di moltiplicazione dei gradienti effettiva

double tc=10;           //Tempo ciclo "Timed Loop" espresso in [ms]
double aspetta;        //contatore del tempo di attesa

int itero=0;           //contatore delle iterazioni dell'algoritmo

double grad_phi_old=0; //valore gradiente fase iterazione precedente
double grad_A_old=0;  //valore gradiente ampiezza ciclo precedente
int salto_phi=0;      //contatore variazioni segno gradiente fase
int salto_A=0;        //contatore variazioni segno gradiente ampiezza

double A_cont_init;   //valore iniziale ampiezza controllo
double delta_A=0;     //valore iniziale variazione ampiezza controllo
double max_Vout=3.0;  //massima tensione positiva ammissibile in uscita
double min_Vout=-3.0; //massima tensione negativa ammissibile in uscita
double Pa2V=0;        //fattore di conversione da Pa a Volt per l'altoparlante

double sat_phi;       //valore di controllo sulla variazione di fase
double sat_A;         //valore di controllo sulla variazione di ampiezza
double tempo_attesa;  //tempo espresso in [ms] di attesa dell'algoritmo

double soglia_AMP=0.15; //soglia in [Pa] raggiungimento punto di minimo algoritmo

int flag_controllo_old=1; //flag per accensione e spegnimento segnale di controllo
int flag_rumore_old=1;   //flag per accensione e spegnimento segnale di rumore

int iterazione=0;

/* —— Passi dell'algoritmo di controllo —— */
enum TypePasso
{
    p_scrivoRum,           //0
    p_attesaRum,           //1
    p_leggoRum,            //2
    p_AMPrum,              //3
    p_scrivoCont,         //4

```

```

    p_attesaCont ,           //5
    p_leggoCont ,           //6
    p_AMPcont ,             //7
    p_scrivoContDelta ,     //8
    p_attesaContDelta ,     //9
    p_leggoContDelta ,     //10
    p_AMPcontDelta ,        //11
    p_gradiente ,           //12
    p_verificaAMP ,         //13
    p_STOP                  //14
};
passo=p_scrivoRum;

/*-----*/
/* DLL Main */
/*-----*/
int __stdcall DllMain (HINSTANCE hinstDLL , DWORD fdwReason , LPVOID lpvReserved)
{
    switch (fdwReason)
    {
        case DLL_PROCESS_ATTACH:

            // Respond to DLL loading by initializing the RTE
            if (InitCVIRTE (hinstDLL , 0 , 0) == 0)
                return 0;
            break;

        case DLL_PROCESS_DETACH:

            // Respond to DLL unloading by closing the RTE for its use
            if (!CVIRTEHasBeenDetached ())
                CloseCVIRTE ();
            break;
    }

    //Return 1 to indicate successful initialization
    return 1;
}

/*-----*/
/* Funzione 1 - Inizializzazione */
/*-----*/
int Inizializzazione (void)
{
    int res =0;

    /* ----- Creazione task per le uscite analogiche -----*/
    if( hMyTask == -1 )
        res = DAQmxCreateTask(TASK_A01.NAME,&hMyTask);
}

```

```

else
{
    res=1;
    return res;
}

if( res==0 )
    res = DAQmxCreateAOVoltageChan(hMyTask,"PXI1Slot4/ao0:1","",min_Vout,
        max_Vout,DAQmx_Val_Volts,"");
else
{
    res=2;
    return res;
}

if( res==0 )
    res = DAQmxSetWriteAttribute(hMyTask,DAQmx_Write_RegenMode,
        DAQmx_Val_AllowRegen);
else
{
    res=3;
    return res;
}

if( res==0 )
    res = DAQmxCfgSampClkTiming(hMyTask,"",Fc_out,DAQmx_Val_Rising,
        DAQmx_Val_ContSamps,1);
    // Il buffer vale 1 perché dopo viene sovrascritto
else
{
    res=4;
    return res;
}

/* ----- Creazione task per ingressi analogici -----*/
if( kMyTask == -1 )
    res = DAQmxCreateTask(TASK_A1_NAME,&kMyTask);
else
{
    res=5;
    return res;
}

if( res==0 )
    res = DAQmxCreateAIVoltageChan(kMyTask,"PXI1Slot2/ai0:1","",
        DAQmx_Val_Cfg_Default,-10.0,10.0,DAQmx_Val_Volts,NULL);
else
{
    res=6;

```



```

        return res;
    }

    if( res==0 )
        res = DAQmxCfgSampClkTiming(kMyTask, "", Fc_in , DAQmx_Val_Rising , DAQmx_Val_FiniteSamps ,
            (uInt32)L_buffer_mic);
    else
    {
        res=7;
        return res;
    }

    if( array_read == NULL )
        array_read=(double*) malloc ( L_buffer_mic*NumChannels_in*sizeof(double));
    if( array_mic1 == NULL )
        array_mic1=(double*) malloc ( L_buffer_mic*sizeof(double));
    if( array_mic2 == NULL )
        array_mic2=(double*) malloc ( L_buffer_mic*sizeof(double));
    if( spettro_real == NULL )
        spettro_real=(double*) malloc ( L_buffer_mic*sizeof(double));
    if( spettro_imag == NULL )
        spettro_imag=(double*) malloc ( L_buffer_mic*sizeof(double));
    if( modulo == NULL )
        modulo=(double*) malloc (( L_buffer_mic/2+1)*sizeof(double));

    //Allocazione temporanea della memoria per segnale (verrà riallocata)
    if( segnale == NULL )
        segnale =(double *) malloc(memoria_alloc*sizeof(double));
    if( A == NULL )
        A =(double *) malloc(n_freq_old*sizeof(double));
    if( f == NULL )
        f =(double *) malloc(n_freq_old*sizeof(double));
    if( phi == NULL )
        phi =(double *) malloc(n_freq_old*sizeof(double));

    // Inizializzazione variabili
    A_cont[0]=0;
    phi_cont[0]=0;
    aspetta=tc;
    k1_init=k1;
    k2=k1;

    return res;
}

```

```

/*-----*/
/* Funzione 2 - Ciclo_lento */
/*-----*/
double Ciclo_lento (int n_freq_rum, double *A_rum, double *f_rum,
                    double *phi_rum, double *A_cont, double *phi_cont, double *out1,
                    double*out2, int controllo_ON, int rumore_ON)
{
    int i, j;
    int res=0;
    double delta_phi=(5.0/180.0)*PI;

    int cond_A=0;
    int cond_phi=0;
    double k2_phi=0.0;
    double k2_A=0.0;
    int dimezza_k1=0;
    double phi_cont_volt;

    // Frequenza controllo = Frequenza rumore
    f_cont[0]=f_rum[0];

    iterazione++;

    // Conversione gradi-radiani
    for (j=0; j<n_freq_rum; j++)
        phi_rum[j]=(phi_rum[j]/180.0)*PI;
    //nota: da ora le fasi si intendono tutte in radianti

    passo_old=passo;
    switch (passo)
    {
    case p_scrivoRum:

        if (n_freq_rum==0)
            passo=p_scrivoRum;
        else
        {
            Eseguo_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                            n_freq_cont, A_cont, f_cont, phi_cont);

            Setting_parametri(f_rum[0], &sat_A, &sat_phi, &k1, &k1_fin,
                            &tempo_attesa);
            passo=p_attesaRum;
        }
        break;

    case p_attesaRum:

        fz_attendi(tempo_attesa);

```

```

    break;

case p_leggoRum:

    res = Leggi(array_read , &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoRum;
    if (stato_lettura == 2)
        passo=p_leggoRum;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPrum;
    }
    break;

case p_AMPrum:

    Calcolo_Ampiezza(array_mic2 , &AMP_0, L_buffer_mic , f_rum[0]);

    // Calcolo dell'incremento delta_A
    FRF_cassa_controllo(f_cont[0] , 1.0 , phi_cont[0] , &Pa2V , &phi_cont_volt);
    delta_A=(max_Vout/100.0)/Pa2V;

    //A_cont[0]=sqrt(RMS_0)*sqrt(2);
    //A_cont_init=1.5/Pa2V;
    A_cont_init=1/Pa2V;
    A_cont[0]=A_cont_init;

    step=0;
    passo=p_scrivoCont;

    out2[0]=k1;
    out2[1]=A_rum[0];
    out2[2]=phi_rum[0];
    out2[3]=f_rum[0];
    out2[4]=AMP_0;
    out2[5]=soglia_AMP;
    out2[6]=tempo_attesa;

    break;

case p_scrivoCont:

    Esegui_scrittura(n_freq_rum , A_rum , f_rum , phi_rum , n_freq_cont , A_cont ,
        f_cont , phi_cont);

```

```
    passo=p_attesaCont;
    break;

case p_attesaCont:

    fz_attendi(tempo_attesa);
    break;

case p_leggoCont:

    Leggi(array_read , &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoCont;
    if (stato_lettura == 2)
        passo=p_leggoCont;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPcont;
    }
    break;

case p_AMPcont:

    Calcolo_Ampiezza(array_mic2 , &AMP_1, L_buffer_mic , f_rum[0]);

    if (AMP_1<(soglia_AMP))
    {
        condizione_AMP=1;
        passo=p_verificaAMP;
    }
    else
    {
        if (itero>15 & (AMP_1<0.2*AMP_0 | AMP_1<soglia_AMP))
        {
            condizione_AMP=1;
            passo=p_verificaAMP;
        }
        else
            passo=p_scrivoContDelta;
    }
    break;

case p_scrivoContDelta:

    if (step==0)
    {
```

```

        A_cont_temp[0]=A_cont[0];
        phi_cont_temp[0]=phi_cont[0]+delta_phi;
    }
    else
    {
        A_cont_temp[0]=A_cont[0]+delta_A;
        phi_cont_temp[0]=phi_cont[0];
    }
    Esegui_scrivtura (n_freq_rum , A_rum , f_rum , phi_rum , n_freq_cont ,
        A_cont_temp , f_cont , phi_cont_temp);

    passo=p_attesaContDelta;
    break;

case p_attesaContDelta:

    fz_attendi(tempo_attesa);
    break;

case p_leggoContDelta:

    Leggi(array_read , &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoContDelta;
    if (stato_lettura == 2)
        passo=p_leggoContDelta;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPcontDelta;
    }
    break;

case p_AMPcontDelta:

    if (step==0)
        Calcolo_Ampiezza(array_mic2 , &AMP_2, L_buffer_mic , f_rum[0]);

    else
        Calcolo_Ampiezza(array_mic2 , &AMP_3, L_buffer_mic , f_rum[0]);

    passo=p_gradiente;
    break;

case p_gradiente:

    if (step==0)

```

```

{
    //Calcolo del gradiente
    grad_phi=(AMP_2-AMP_1)/delta_phi;
    step=1;
    passo=p_scrivoContDelta;
}
else
{
    //Calcolo del gradiente
    grad_A=(AMP_3-AMP_1)/delta_A;

    if ( ( fabs(k1*grad_phi) ) > sat_phi )
        cond_phi=1;
    else
        cond_phi=0;

    if ( ( fabs(k1*grad_A) ) > sat_A/Pa2V )
        cond_A=1;
    else
        cond_A=0;

    k2_phi=sat_phi/fabs(grad_phi);
    k2_A=(sat_A/Pa2V)/fabs(grad_A);

    if ( (!cond_A) & cond_phi)
        k2=k2_phi;

    if ( cond_A & (!cond_phi) )
        k2=k2_A;

    if ( cond_A & cond_phi)
    {
        if ( k2_phi >= k2_A )
            k2=k2_A;
        else
            k2=k2_phi;
    }

    if ( (!cond_A) & (!cond_phi) )
        k2=k1;

    A_cont[0]=A_cont[0]-k2*grad_A;
    phi_cont[0]=phi_cont[0]-k2*grad_phi;

    //////////////////////////////////////// METODO 2
    if ( ( grad_phi*grad_phi_old)<0.0 )
    {
        salto_phi++;
        if ( salto_phi>2)

```

```

        {
            salto_phi=0;
            dimezza_k1=1;
        }
    }

    if ( (grad_A*grad_A_old)<0.0 )
    {
        salto_A++;
        if (salto_A >2)
        {
            salto_A=0;
            dimezza_k1=1;
        }
    }

    if ( dimezza_k1==1 )
        k1=k1/2;

    if (k1<k1_fin)
        k1=k1_fin;

    grad_A_old=grad_A;
    grad_phi_old=grad_phi;
    ////////////////////////////////// FINE METODO 2

    passo=p_verificaAMP;
    step=0;

    if ( fabs(A_cont[0]) > (3.0/Pa2V) )
        A_cont[0]=3.0*Pa2V*( fabs(A_cont[0]) / A_cont[0] );

    /* ----- Pubblicazione delle variabili ----- */
    itero++;
    out1[0]=itero;
    out1[1]=A_cont[0];
    out1[2]=phi_cont[0];
    out1[3]=AMP_1;
    out1[4]=A_cont[0]+delta_A;
    out1[5]=AMP_2;
    out1[6]=phi_cont[0]+delta_phi;
    out1[7]=AMP_3;
    out1[8]=grad_A;
    out1[9]=grad_phi;
    out1[10]=k2;
}
A_cont_old=A_cont[0];
phi_cont_old=phi_cont[0];
break;

```

```

case p_verificaAMP :

    if ( itero >50 || condizione_AMP==1)
    {
        //prova=(double)iterazione*tc/1000;
        passo=p_STOP;

        /* ——— Pubblicazione delle variabili ——— */
        itero++;
        out1[0]=itero ;
        out1[1]=A_cont [0];
        out1[2]=phi_cont [0];
        out1[3]=AMP_1;
        out1[4]=A_cont[0]+delta_A ;
        out1[5]=AMP_2;
        out1[6]=phi_cont [0]+delta_phi ;
        out1[7]=AMP_3;
        out1[8]=grad_A ;
        out1[9]=grad_phi ;
        out1[10]=k2;
        itero --;

        break;
    }
    else
        passo=p_scrivoCont;

    break;
}

if ( passo==p_STOP )
{
    passo_old=1000;

    if ( flag_controllo_old != controllo_ON || flag_rumore_old != rumore_ON)
    {
        if ( !controllo_ON & rumore_ON)
        {
            resetta(A_cont_temp, n_freq_cont);
            Eseguo_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                n_freq_cont, A_cont_temp, f_cont, phi_cont);
        }
        if ( controllo_ON & rumore_ON)
        {
            Eseguo_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                n_freq_cont, A_cont, f_cont, phi_cont);
        }
        if ( !rumore_ON & controllo_ON)

```



```

        {
            resetta(A_rum_temp, n_freq_rum);
            Eseguo_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                            n_freq_cont, A_cont, f_cont, phi_cont);
        }
        if ( !rumore_ON & !controllo_ON)
        {
            resetta(A_rum_temp, n_freq_rum);
            resetta(A_cont_temp, n_freq_cont);
            Eseguo_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                            n_freq_cont, A_cont_temp, f_cont, phi_cont);
        }
    }
    flag_controllo_old=controllo_ON;
    flag_rumore_old=rumore_ON;
}
return passo_old;
}

/*-----*/
/* Funzione 3 - Pulizia */
/*-----*/
int Pulizia (void)
{
    int res=0;
    int i=0;

    double n_freq_rum;
    double f_rum[10];
    double phi_rum[10];

    n_freq_rum=n_freq_cont;

    for (i=0; i<n_freq_rum; i++)
    {
        f_rum[i]=1000.0;
        phi_rum[i]=0.0;
    }

    resetta(A_rum_temp, n_freq_rum);
    resetta(A_cont_temp, n_freq_cont);

    Eseguo_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum, n_freq_cont,
                    A_cont_temp, f_cont, phi_cont);

    if (hMyTask!=-1)
    {
        res = DAQmxClearTask(hMyTask);
        hMyTask=-1;
    }
}

```

```
    }
    else
        res=1;

    if (res==0)
    {
        if (kMyTask!=-1)
        {
            res = DAQmxClearTask(kMyTask);
            kMyTask=-1;
        }
    }
    else
        res=2;

    passo=0;
    return res;
}
```

### File “cvidll.h”

Si veda par. D.1.1.

### File “func\_lib.h”

Si veda par. D.1.1.

## D.2.2 Analisi dettagliata delle funzioni del codice

Vengono analizzate le funzioni contenute all'interno del codice per l'esecuzione del sistema di controllo AS.

### Funzione “Inizializzazione()”

Questa funzione è identica a quella del metodo ARMS, si veda pertanto il paragrafo D.1.2.

### Funzione “Ciclo\_Lento()”

Questa funzione contiene il nocciolo dell'algoritmo di controllo secondo lo schema mostrato nel diagramma a blocchi in figura 3.59. Si analizzano di seguito le sole parti del codice differenti rispetto al metodo ARMS.

- Nelle fasi 4), 7), 11) e 11') viene valutato lo spettro del segnale letto dal microfono e acquisito successivamente dalle schede di input. Il segnale acquisito è stato memorizzato nelle fasi precedenti all'interno di un vettore. Lo spettro del segnale e l'ampiezza dell'armonica viene valutata per mezzo della funzione *Calcolo\_ampiezza()*, questa riceve in ingresso l'array  $a_{mic}$ , in cui è memorizzato il segnale appena acquisito e la sua dimensione  $N$ , mentre in uscita fornisce il valore dell'ampiezza dell'armonica analizzata. La funzione *Calcolo\_ampiezza()* è strutturata nei seguenti passi:
  - Calcolo dello spettro del segnale mediante l'utilizzo della funzione *FFT()*; questa funzione è disponibile all'interno della libreria di funzioni *Analysis.h*.
  - Calcolo del modulo dello spettro determinato al punto precedente.
  - Determinazione dell'indice  $i$  dell'array contenente il modulo dello spettro, calcolato in base alla risoluzione spettrale ottenuta e corrispondente alla frequenza alla quale si vuole minimizzare il rumore.
  - Ricerca del massimo valore dell'ampiezza del picco nell'intorno della frequenza di interesse. Questo è dovuto al fatto che l'asse delle frequenze è discreto e l'energia spettrale è suddivisa in più righe spettrali a causa del fenomeno di leakage.
- Nella fase 14) dello schema a blocchi viene rivalutato il passo  $G_2^k$  da fornire per la valutazione dei nuovi parametri di controllo. Le relazioni matematiche per il calcolo del nuovo passo sono quelle riportate nel paragrafo 3.4.2 al punto 14). Analizzando il comportamento sperimentale dell'algorithm, al fine di garantire la convergenza del metodo di ricerca del minimo, possono essere utilizzati i medesimi valori del metodo basato sull'RMS, in particolare si ha:

$$\begin{aligned}
 A_{SAT} &= \frac{V_{max}}{20}, & \phi_{SAT} &= 10^\circ, & G_1^{k=0} &= 0.5, & f_{rum} &< 800 \text{ Hz}, \\
 A_{SAT} &= \frac{V_{max}}{10}, & \phi_{SAT} &= 30^\circ, & G_1^{k=0} &= 3, & 800 \leq f_{rum} &< 1350 \text{ Hz}, \\
 A_{SAT} &= \frac{V_{max}}{10}, & \phi_{SAT} &= 30^\circ, & G_1^{k=0} &= 3, & f_{rum} &\geq 1350 \text{ Hz},
 \end{aligned}$$

dove  $V_{max} = 3 \text{ V}$  è il fondoscala massimo delle uscite della scheda di output. Il valore di soglia inferiore per il passo  $G_1^k$  è stato assunto in funzione della frequenza pari a:

$$\begin{aligned}
 G_{1,MIN}^k &= 0.005, & f_{rum} &< 700 \text{ Hz}, \\
 G_{1,MIN}^k &= 0.05, & 700 &\leq f_{rum} \leq 1350 \text{ Hz}, \\
 G_{1,MIN}^k &= 0.1, & f_{rum} &\geq 1350 \text{ Hz}.
 \end{aligned}$$

L'impostazione di questi parametri è effettuata dalla funzione *Setting\_parametri()* che riceve in ingresso il valore della frequenza del segnale di rumore.

- Nella fase 15) vengono calcolati i valori dei nuovi parametri di controllo secondo le relazioni 3.45 e 3.46.
- Nella fase 8) viene effettuato il confronto tra il valore attuale di ampiezza e il valore dell'ampiezza di soglia (fig. 3.50). Se il valore dell'ampiezza attuale è superiore alla soglia, l'algoritmo prosegue verso la fase 9) con i nuovi valori dei parametri di controllo mentre, se l'ampiezza raggiunta è inferiore o uguale al valore di soglia, l'algoritmo procede al termine con la fase 16), mantenendo i valori attuali dei parametri di controllo. Questi valori sono quelli che consentono di abbattere il rumore generato all'ingresso del tubo in corrispondenza di una determinata frequenza fino al valore della soglia di rumore desiderata. Inoltre è stato fissato un numero massimo di iterazioni in modo da consentire all'algoritmo di terminare qualora questo non abbia ancora raggiunto il valore di soglia desiderato dopo 15 iterazioni.

### Funzione "Pulizia()"

Questa funzione è identica a quella del metodo ARMS, si veda pertanto il paragrafo D.1.2.

## D.3 Codice sorgente per l'algoritmo MFS

Il codice di controllo è esattamente identico a quello visto per il metodo AS, a patto di richiamare il loop di abbattimento per ciascuna frequenza in cui vi vuole abbattere il rumore, a tal fine si faccia pienamente riferimento a quanto riportato nell'appendice D.2.

### D.3.1 Struttura del codice

Vengono riportati i file dei codici sorgenti per il controllo del rumore mediante minimizzazione seriale dei picchi spettrali contenuti nello spettro del segnale di rumore multi-armonico rilevato dal microfono di errore:

1. *cvidll.c*
2. *cvidll.h*
3. *func\_lib.h*

#### File “cvidll.c”

```

/*-----*/
/* Include files */
/*-----*/
#include <windows.h>
#include <analysis.h>
#include <cvirte.h>
#include "cvidll.h"
#include "func_lib.h"
#include <NIDAQmx.h>
#include <utility.h>
#include <ansi_c.h>
#include <math.h>

/*-----*/
/* Definizione e inizializzazione variabili globali */
/*-----*/
TaskHandle hMyTask=-1; //Task di uscite analogiche
double Fc_out=100000; //frequenza di campionamento delle uscite
double *A=NULL; //array ampiezze rumore e controllo
double *f=NULL; //array frequenze rumore e controllo
double *phi=NULL; //array fasi rumore e controllo
double *segnale=NULL; //array segnale da scrivere in uscita
double L_buffer=500; //numero di punti del buffer di scrittura
double memoria_alloc=100000; //numero di punti iniziale del buffer di scrittura

TaskHandle kMyTask=-1; //Task di ingressi analogici
double Fc_in =50000; //frequenza campionamento lettura ingressi
double scala_mic=3.16456; //Scala conversione Volt -> Pascal
int NumChannels_in=2; //numero di canali in ingresso
double L_buffer_mic=4096; //numero di punti del buffer di lettura
double *array_mic1=NULL; //puntatore array lettura microfono 1
double *array_mic2=NULL; //puntatore array lettura microfono 2
double *array_read=NULL; //puntatore array segnali microfoni

```

```

double *spettro_real=NULL; //puntatore array parte reale spettro segnale
double *spettro_imag=NULL; //puntatore array parte immaginaria spettro segnale
double *modulo=NULL; //puntatore array modulo dello spettro del segnale
int32 samples_read; //campioni letti , caricati in ram

int stato_lettura=0; //flag dello stato della lettura
int passo; //contatore dei passi dell'algoritmo
int passo_old=100; //contatore dei passi dell'algoritmo

double A_cont[10]; //array ampiezze controllo
double f_cont[10]; //array frequenze controllo
double phi_cont[10]; //array fasi controllo

double A_cont_temp[10]; //array ampiezze controllo
double phi_cont_temp[10]; //array fasi controllo
double A_rum_temp[10];

int n_freq_cont=1; //numero di frequenze del controllo
int n_freq_old=2; //numero di frequenze di rumore + controllo

uint32 campioni_acq; //campioni acquisiti finora dalla scheda
double AMP_0[10]; //array moduli ampiezza delle frequenze del rumore
double AMP_1; //ampiezza rumore + controllo
double AMP_2; //ampiezza rumore + controllo con var. fase
double AMP_3; //ampiezza rumore + controllo con var. ampiezza

int condizione_AMP=0; //flag di verifica della condizione dell'ampiezza
int step=0; //flag operazioni calcolo dei gradienti

double grad_phi=0.0; //gradiente della fase
double grad_A=0.0; //gradiente dell'ampiezza
double k1; //costante moltiplicazione gradienti
double k1_init; //valore iniziale della cost. di molt. gradienti
double k1_fin; //valore finale della cost. di molt. gradienti
double k2=0.0; //costante moltiplicazione gradienti effettiva

double tc=10; //Tempo ciclo "Timed Loop" espresso in [ms]
double aspetta; //contatore del tempo di attesa

int itero=0; //contatore delle iterazioni dell'algoritmo

double grad_phi_old=0.0; //valore gradiente fase iterazione precedente
double grad_A_old=0.0; //valore gradiente ampiezza ciclo precedente
int salto_phi=0; //contatore variazioni segno gradiente fase
int salto_A=0; //contatore variazioni segno gradiente ampiezza

double A_cont_init=0; //valore iniziale ampiezza controllo
double delta_A=0.0; //valore iniziale variazione ampiezza controllo

```

```

double max_Vout=3.0;           //massima tensione positiva ammissibile in uscita
double min_Vout=-3.0;         //massima tensione negativa ammissibile in uscita
double Pa2V=0.0;             //fattore conversione da Pa2Volt per l'altoparlante

double sat_phi;               //flag di controllo sulla variazione di fase
double sat_A;                 //flag di controllo sulla variazione di ampiezza
double tempo_attesa;         //tempo espresso in [ms] di attesa dell'algoritmo

double soglia_AMP=0.15;      //soglia in [Pa] di raggiungimento del minimo
int conta_freq=0;           //indice relativo alla frequenza analizzata

int flag_controllo_old=1;     //flag accensione e spegnimento segnale controllo
int flag_rumore_old=1;       //flag accensione e spegnimento segnale rumore

/* ----- Passi dell'algoritmo di controllo ----- */
enum TypePasso
{
    p_scrivoRum ,               //0
    p_attesaRum ,               //1
    p_leggoRum ,                //2
    p_AMPrum ,                  //3
    p_scrivoCont ,              //4
    p_attesaCont ,              //5
    p_leggoCont ,               //6
    p_AMPcont ,                 //7
    p_scrivoContDelta ,         //8
    p_attesaContDelta ,         //9
    p_leggoContDelta ,          //10
    p_AMPcontDelta ,            //11
    p_gradiente ,               //12
    p_verificaAMP ,             //13
    p_STOP                       //14
};
passo=p_scrivoRum;

/*-----*/
/* DLL Main */
/*-----*/
int __stdcall DllMain (HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    switch (fdwReason)
    {
        case DLL_PROCESS_ATTACH:

            // Respond to DLL loading by initializing the RTE
            if (InitCVIRTE (hinstDLL, 0, 0) == 0)
                return 0;
            break;
    }
}

```

```

case DLLPROCESS_DETACH:

    // Respond to DLL unloading by closing the RTE for its use
    if (!CVIRTEHasBeenDetached ())
        CloseCVIRTE ();
    break;
}

//Return 1 to indicate successful initialization
return 1;
}

/*-----*/
/* Funzione 1 - Inizializzazione */
/*-----*/
int Inizializzazione (void)
{
    int res =0;

    /* ----- Creazione task per le uscite analogiche -----*/
    if( hMyTask == -1 )
        res = DAQmxCreateTask(TASK_A01_NAME,&hMyTask);
    else
    {
        res=1;
        return res;
    }

    if( res==0 )
        res = DAQmxCreateAOVoltageChan(hMyTask,"PXI1Slot4/ao0:1", "",
            min_Vout , max_Vout , DAQmx_Val_Volts , "" );
    else
    {
        res=2;
        return res;
    }

    if( res==0 )
        res = DAQmxSetWriteAttribute(hMyTask, DAQmx_Write_RegenMode ,
            DAQmx_Val_AllowRegen );
    else
    {
        res=3;
        return res;
    }

    if( res==0 )
        res = DAQmxCfgSampClkTiming(hMyTask, "" , Fc_out , DAQmx_Val_Rising ,
            DAQmx_Val_ContSamps , 1 );

```



```

        // Il buffer vale 1 perché dopo viene sovrascritto
    else
    {
        res=4;
        return res;
    }

    /* ----- Creazione task per gli ingressi analogici -----*/
    if( kMyTask == -1 )
        res = DAQmxCreateTask(TASK_A1_NAME,&kMyTask);
    else
    {
        res=5;
        return res;
    }

    if( res==0 )
        res = DAQmxCreateAIVoltageChan (kMyTask, "PXI1Slot2/ai0:1", "",
            DAQmx_Val_Cfg_Default, -10.0, 10.0, DAQmx_Val_Volts, NULL);
    else
    {
        res=6;
        return res;
    }

    if( res==0 )
        res = DAQmxCfgSampClkTiming(kMyTask, "", Fc_in, DAQmx_Val_Rising, DAQmx_Val_FiniteSamps,
            (uInt32)L_buffer_mic);
    else
    {
        res=7;
        return res;
    }

    /* ----- Inizializzazione array -----*/
    if( array_read == NULL )
        array_read=(double*) malloc (L_buffer_mic*NumChannels_in*sizeof(double));
    if( array_mic1 == NULL )
        array_mic1=(double*) malloc (L_buffer_mic*sizeof(double));
    if( array_mic2 == NULL )
        array_mic2=(double*) malloc (L_buffer_mic*sizeof(double));
    if( spettro_real == NULL )
        spettro_real=(double*) malloc (L_buffer_mic*sizeof(double));
    if( spettro_imag == NULL )
        spettro_imag=(double*) malloc (L_buffer_mic*sizeof(double));
    if( modulo == NULL )
        modulo=(double*) malloc ((L_buffer_mic/2+1)*sizeof(double));

```

```

//Allocazione temporanea della memoria per il segnale (verrà riallocata)
if( segnale == NULL )
    segnale =(double *) malloc(memoria_alloc*sizeof(double));
if( A == NULL )
    A =(double *) malloc(n_freq_old*sizeof(double));
if( f == NULL )
    f =(double *) malloc(n_freq_old*sizeof(double));
if( phi == NULL )
    phi =(double *) malloc(n_freq_old*sizeof(double));

// Inizializzazione variabili
resetta(A_cont, 10);
resetta(phi_cont, 10);
resetta(A_rum_temp, 10);
resetta(A_cont_temp,10);
aspetta=tc;
k1_init=k1;
k2=k1;

return res;
}

/*-----*/
/* Funzione 2 - Ciclo_lento */
/*-----*/
double Ciclo_lento (int n_freq_rum, double *A_rum, double *f_rum,
double *phi_rum, double *A_cont, double *phi_cont, double *out1,
double*out2, int controllo_ON, int rumore_ON)
{
    int i,j;
    int res=0;
    double delta_phi=(5.0/180.0)*PI;

    int cond_A=0;
    int cond_phi=0;
    double k2_phi=0.0;
    double k2_A=0.0;
    int dimezza_k1=0;
    double phi_cont_volt;

    double AMP_temp;

    n_freq_cont=n_freq_rum;

    for (j=0; j<n_freq_rum; j++)
    {
        f_cont[j]=f_rum[j];
        phi_rum[j]=(phi_rum[j]/180.0)*PI;
    }
}

```

```
passo_old=passo;
switch (passo)
{
case p_scrivoRum:

    if (n_freq_rum==0)
        passo=p_scrivoRum;
    else
    {
        Esegui_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
            n_freq_cont, A_cont, f_cont, phi_cont);

        tempo_attesa=400;
        passo=p_attesaRum;
    }
    break;

case p_attesaRum:

    fz_attendi(tempo_attesa);
    break;

case p_leggoRum:

    res = Leggi(array_read, &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoRum;
    if (stato_lettura == 2)
        passo=p_leggoRum;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPrum;
    }
    break;

case p_AMPrum:

    for (j=0; j<n_freq_rum; j++)
    {
        Calcolo_Ampiezza(array_mic2, &AMP_temp, L_buffer_mic, f_rum[j]);
        AMP_0[j]=AMP_temp;
    }
}
```

```

/* ----- Calcolo dell'incremento delta_A ----- */
FRF_cassa_controllo(f_cont[conta_freq], 1.0, phi_cont[conta_freq], &Pa2V,
    &phi_cont_volt);

delta_A=(max_Vout/100.0)/Pa2V;
A_cont_init=1/Pa2V;
A_cont[conta_freq]=A_cont_init;

Setting_parametri(f_rum[conta_freq], &sat_A, &sat_phi, &k1, &k1_fin,
    &tempo_attesa);

step=0;
passo=p_scrivoCont;

break;

case p_scrivoCont:

    Esegui_scrittura(n_freq_rum, A_rum, f_rum, phi_rum, n_freq_cont, A_cont,
        f_cont, phi_cont);

    passo=p_attesaCont;

    out2[0]=k1;
    out2[1]=A_rum[conta_freq];
    out2[2]=phi_rum[conta_freq];
    out2[3]=f_rum[conta_freq];
    out2[4]=AMP_0[conta_freq];
    out2[5]=soglia_AMP;
    out2[6]=tempo_attesa;

    break;

case p_attesaCont:

    fz_attendi(tempo_attesa);
    break;

case p_leggoCont:

    Leggi(array_read, &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoCont;
    if (stato_lettura == 2)
        passo=p_leggoCont;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)

```

```

        array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPcont;
    }
    break;

case p_AMPcont:

    Calcolo_Ampiezza(array_mic2 , &AMP_1, L_buffer_mic , f_rum[conta_freq]);

    if (AMP_1<(soglia_AMP))
    {
        condizione_AMP=1;
        passo=p_verificaAMP;
    }
    else
    {
        if (itero >15 & (AMP_1<0.2*AMP_0[conta_freq] | AMP_1<soglia_AMP))
        {
            condizione_AMP=1;
            passo=p_verificaAMP;
        }
        else
            passo=p_scrivoContDelta;
    }
    break;

case p_scrivoContDelta:

    if (step==0)
    {
        A_cont_temp[conta_freq]=A_cont[conta_freq];
        phi_cont_temp[conta_freq]=phi_cont[conta_freq]+delta_phi;
    }
    else
    {
        A_cont_temp[conta_freq]=A_cont[conta_freq]+delta_A;
        phi_cont_temp[conta_freq]=phi_cont[conta_freq];
    }

    Esegui_scrittura (n_freq_rum , A_rum , f_rum , phi_rum ,
                    n_freq_cont , A_cont_temp , f_cont , phi_cont_temp);

    passo=p_attesaContDelta;
    break;

case p_attesaContDelta:

    fz_attendi(tempo_attesa);

```

```

    break;

case p_leggoContDelta:

    Leggi(array_read, &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoContDelta;
    if (stato_lettura == 2)
        passo=p_leggoContDelta;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPcontDelta;
    }
    break;

case p_AMPcontDelta:

    if (step==0)
        Calcolo_Ampiezza(array_mic2, &AMP_2, L_buffer_mic, f_rum[conta_freq]);
    else
        Calcolo_Ampiezza(array_mic2, &AMP_3, L_buffer_mic, f_rum[conta_freq]);

    passo=p_gradiente;
    break;

case p_gradiente:

    if (step==0)
    {
        /* ----- Calcolo del gradiente della fase -----*/
        grad_phi=(AMP_2-AMP_1)/delta_phi;
        step=1;
        passo=p_scrivoContDelta;
    }
    else
    {
        /* ----- Calcolo del gradiente dell'ampiezza -----*/
        grad_A=(AMP_3-AMP_1)/delta_A;

        /* ----- Verifica dei passi effettuati dall'algoritmo -----*/
        if ( ( fabs(k1*grad_phi) ) > sat_phi )
            cond_phi=1;
        else
            cond_phi=0;

        if ( ( fabs(k1*grad_A) ) > sat_A/Pa2V )

```

```

        cond_A=1;
    else
        cond_A=0;

    k2_phi=sat_phi/fabs(grad_phi);
    k2_A=(sat_A/Pa2V)/fabs(grad_A);

    if ( (!cond_A) & cond_phi)
        k2=k2_phi;

    if ( cond_A & (!cond_phi) )
        k2=k2_A;

    if ( cond_A & cond_phi)
    {
        if ( k2_phi >= k2_A )
            k2=k2_A;
        else
            k2=k2_phi;
    }

    if ( (!cond_A) & (!cond_phi) )
        k2=k1;

    /* ----- Generazione dei nuovi valori del controllo -----*/
    A_cont[conta_freq]=A_cont[conta_freq]-k2*grad_A;
    phi_cont[conta_freq]=phi_cont[conta_freq]-k2*grad_phi;

    /* ----- Calcolo della costante k1 -----*/
    if ( (grad_phi*grad_phi_old)<0.0 )
    {
        salto_phi++;
        if (salto_phi>2)
        {
            salto_phi=0;
            dimezza_k1=1;
        }
    }

    if ( (grad_A*grad_A_old)<0.0 )
    {
        salto_A++;
        if (salto_A>2)
        {
            salto_A=0;
            dimezza_k1=1;
        }
    }

```

```

if ( dimezza_k1==1 )
    k1=k1/2;

if ( k1<k1_fin )
    k1=k1_fin;

grad_A_old=grad_A;
grad_phi_old=grad_phi;

passo=p_verificaAMP;
step=0;

/* ——— Pubblicazione delle variabili ——— */
itero++;
out1[0]=itero;
out1[1]=A_cont[conta_freq];
out1[2]=phi_cont[conta_freq];
out1[3]=AMP_1;
out1[4]=A_cont[conta_freq]+delta_A;
out1[5]=AMP_2;
out1[6]=phi_cont[conta_freq]+delta_phi;
out1[7]=AMP_3;
out1[8]=grad_A;
out1[9]=grad_phi;
out1[10]=k2;
out1[11]=conta_freq;
}
break;

case p_verificaAMP:

if ( itero > 50 || condizione_AMP==1 )
{
    if (( conta_freq+1)==n_freq_rum )
    {
        passo=p_STOP;

        /* ——— Pubblicazione delle variabili ——— */
        itero++;
        out1[0]=itero;
        out1[1]=A_cont[conta_freq];
        out1[2]=phi_cont[conta_freq];
        out1[3]=AMP_1;
        out1[4]=A_cont[conta_freq]+delta_A;
        out1[5]=AMP_2;
        out1[6]=phi_cont[conta_freq]+delta_phi;
        out1[7]=AMP_3;
        out1[8]=grad_A;
        out1[9]=grad_phi;
    }
}

```



```

        out1[10]=k2;
        out1[11]= conta_freq;
        itero --;
    }
    else
    {
        passo=p_scrivoCont;
        conta_freq++;
        Setting_parametri(f_rum[conta_freq], &sat_A, &sat_phi, &k1, &k1_fin,
            &tempo_attesa);
        FRF_cassa_controllo(f_cont[conta_freq], 1.0, phi_cont[conta_freq],
            &Pa2V, &phi_cont_volt);
        delta_A=(max_Vout/100.0)/Pa2V;

        A_cont_init=1/Pa2V;
        A_cont[conta_freq]=A_cont_init;
        condizione_AMP=0;
        itero=0;
    }
}
else
    passo=p_scrivoCont;

break;
}

if ( passo==p_STOP )
{
    passo_old=1000;

    if ( flag_controllo_old != controllo_ON || flag_rumore_old != rumore_ON)
    {
        if ( !controllo_ON & rumore_ON)
        {
            resetta(A_cont_temp, n_freq_cont);
            Esegui_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                n_freq_cont, A_cont_temp, f_cont, phi_cont);
        }
        if ( controllo_ON & rumore_ON)
        {
            Esegui_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                n_freq_cont, A_cont, f_cont, phi_cont);
        }
        if ( !rumore_ON & controllo_ON)
        {
            resetta(A_rum_temp, n_freq_rum);
            Esegui_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                n_freq_cont, A_cont, f_cont, phi_cont);
        }
    }
}

```

```

        if ( !rumore_ON & !controllo_ON)
        {
            resetta (A_rum_temp, n_freq_rum);
            resetta (A_cont_temp, n_freq_cont);
            Eseguo_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                n_freq_cont, A_cont_temp, f_cont, phi_cont);
        }
    }
    flag_controllo_old=controllo_ON;
    flag_rumore_old=rumore_ON;
}
return passo;
}

/*-----*/
/* Funzione 3 - Pulizia */
/*-----*/
int Pulizia (void)
{
    int res =0;
    int i=0;

    double n_freq_rum;
    double f_rum [10];
    double phi_rum [10];

    n_freq_rum=n_freq_cont;

    for (i=0; i<n_freq_rum; i++)
    {
        f_rum [i]=1000.0;
        phi_rum [i]=0.0;
    }
    resetta (A_rum_temp, n_freq_rum);
    resetta (A_cont_temp, n_freq_cont);

    Eseguo_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum, n_freq_cont,
        A_cont_temp, f_cont, phi_cont);

    if (hMyTask!=-1)
    {
        res = DAQmxClearTask(hMyTask);
        hMyTask=-1;
    }
    else
        res=1;

    if (res==0)
    {

```

```
    if (kMyTask!=-1)
    {
        res = DAQmxClearTask(kMyTask);
        kMyTask=-1;
    }
}
else
    res=2;

passo=0;
return res;
}
```

### File “cvidll.h”

Si veda par. D.1.1.

### File “func\_lib.h”

Si veda par. D.1.1.

## D.3.2 Analisi dettagliata delle funzioni del codice

Per il dettaglio delle funzioni contenute all’interno del codice per l’esecuzione del sistema di controllo MFS si rimanda al paragrafo D.4.2.

## D.4 Codice sorgente per l’algoritmo MFP

Il codice sorgente è stato creato con l’ausilio dei medesimi software utilizzati per l’implementazione dei codici di minimizzazione realizzati in precedenza. A partire da questi, di cui si è mantenuta la struttura, si è modificato il codice secondo il processo mostrato nello schema a blocchi in figura 3.76.

### D.4.1 Struttura del codice

Vengono riportati i file dei codici sorgenti per il controllo del rumore mediante minimizzazione in parallelo dei picchi spettrali contenuti nello spettro del segnale di rumore multi-armonico rilevato dal microfono di errore:

1. *cvidll.c*
2. *cvidll.h*

3. *func\_lib.h*

## File "cvidll.c"

```

/*-----*/
/* Include files */
/*-----*/
#include <windows.h>
#include <analysis.h>
#include <cvirte.h>
#include "cvidll.h"
#include "func_lib.h"
#include <NIDAQmx.h>
#include <utility.h>
#include <ansi_c.h>
#include <math.h>

/*-----*/
/* Definizione e inizializzazione variabili globali */
/*-----*/
TaskHandle hMyTask=-1;           //Task di uscite analogiche
double Fc_out=100000;           //frequenza di campionamento delle uscite
double *A=NULL;                 //array ampiezze rumore e controllo
double *f=NULL;                 //array frequenze rumore e controllo
double *phi=NULL;              //array fasi rumore e controllo
double *segnale=NULL;          //array segnale da scrivere in uscita
double L_buffer=500;           //numero punti buffer di scrittura
double memoria_alloc=100000;   //numero punti iniziale buffer di scrittura

TaskHandle kMyTask=-1;         //Task di ingressi analogici
double Fc_in =50000;           //frequenza campionamento ingressi
double scala_mic=3.16456;      //Scala conversione Volt -> Pascal
int NumChannels_in=2;          //numero di canali in ingresso
double L_buffer_mic=4096;      //numero di punti del buffer di lettura
double *array_mic1=NULL;       //puntatore array lettura microfono 1
double *array_mic2=NULL;       //puntatore array lettura microfono 2
double *array_read=NULL;       //puntatore array segnali microfoni

double *spettro_real=NULL;      //puntatore parte reale spettro segnale
double *spettro_imag=NULL;     //puntatore parte immaginaria spettro segnale
double *modulo=NULL;           //puntatore modulo spettro segnale
int32 samples_read;           //campioni letti caricati in ram

int stato_lettura=0;           //flag dello stato della lettura
int passo;                     //contatore dei passi dell'algoritmo
int passo_old=100;             //contatore dei passi dell'algoritmo

double A_cont[10];             //array ampiezze controllo
double f_cont[10];             //array frequenze controllo

```

```

double phi_cont[10];           //array fasi controllo

double A_cont_temp[10];       //array ampiezze controllo
double phi_cont_temp[10];     //array fasi controllo

int n_freq_cont=1;           //numero di frequenze del controllo
int n_freq_old=2;            //numero di frequenze di rumore + controllo

uint32 campioni_acq;         //campioni acquisiti finora dalla scheda
double AMP_0[10];            //array moduli ampiezza frequenze rumore
double AMP_1[10];            //ampiezze rumore + controllo
double AMP_2[10];            //ampiezze rumore + controllo con var. fase
double AMP_3[10];            //ampiezze rumore + controllo con var. ampiezza

int condizione_AMP_singola[10]; //flag verifica condizione ampiezza
int condizione_AMP_tot=1;     //flag verifica condizione su tutte le ampiezze
int step=0;                   //flag operazioni calcolo dei gradienti
double grad_phi[10];          //gradiente della fase
double grad_A[10];            //gradiente dell'ampiezza
double k1[10];                //costante molt. gradienti iniziale
double k1_init[10];           //valore iniziale cost. di molt. dei gradienti
double k1_fin[10];            //valore finale cost. di molt. dei gradienti
double k2[10];                //costante di molt. dei gradienti effettiva

double tc=10;                 //Tempo ciclo "Timed Loop" espresso in [ms]
double aspetta;               //contatore del tempo di attesa
int itero=0;                  //contatore delle iterazioni dell'algoritmo

double grad_phi_old[10];      //gradiente fase iterazione precedente
double grad_A_old[10];        //gradiente ampiezza ciclo precedente
int salto_phi[10];            //contatore variazioni segno gradiente fase
int salto_A[10];              //contatore variazioni segno gradiente ampiezza

double A_cont_init;           //valore iniziale ampiezza controllo
double delta_A[10];           //valore iniziale variazione ampiezza controllo
double max_Vout=3.0;          //massima tensione positiva ammissibile in uscita
double min_Vout=-3.0;         //massima tensione negativa ammissibile in uscita
double Pa2V[10];              //fattore conversione Pa2Volt per l'altoparlante

double sat_phi[10];           //valore di controllo sulla variazione di fase
double sat_A[10];             //valore di controllo sulla variazione di ampiezza
double tempo_attesa=0.0;      //tempo espresso in [ms] di attesa dell'algoritmo

double soglia_AMP=0.15;       //soglia in [Pa] raggiungimento punto di minimo

int flag_controllo_old=1;     //flag accensione e spegnimento segnale controllo
int flag_rumore_old=1;        //flag accensione e spegnimento segnale rumore

/* ——— Passi dell'algoritmo di controllo ——— */

```

```

enum TypePasso
{
    p_scrivoRum ,           //0
    p_attesaRum ,          //1
    p_leggoRum ,           //2
    p_AMPrum ,             //3
    p_scrivoCont ,         //4
    p_attesaCont ,         //5
    p_leggoCont ,          //6
    p_AMPcont ,            //7
    p_scrivoContDelta ,    //8
    p_attesaContDelta ,    //9
    p_leggoContDelta ,     //10
    p_AMPcontDelta ,       //11
    p_gradiente ,          //12
    p_verificaAMP ,        //13
    p_STOP                  //14
};
passo=p_scrivoRum;

/*-----*/
/* DLL Main */
/*-----*/
int __stdcall DllMain (HINSTANCE hinstDLL , DWORD fdwReason , LPVOID lpvReserved)
{
    switch (fdwReason)
    {
        case DLLPROCESS_ATTACH:

            // Respond to DLL loading by initializing the RTE
            if (InitCVIRTE (hinstDLL , 0 , 0) == 0)
                return 0;
            break;

        case DLLPROCESS_DETACH:

            // Respond to DLL unloading by closing the RTE for its use
            if (!CVIRTEHasBeenDetached ())
                CloseCVIRTE ();
            break;
    }
    //Return 1 to indicate successful initialization
    return 1;
}

```

```
/*-----*/
/* Funzione 1 - Inizializzazione */
/*-----*/
int Inizializzazione (void)
{
    int res =0;
    int j=0;

    /* ----- Creo il task per le uscite analogiche -----*/
    if( hMyTask == -1 )
        res = DAQmxCreateTask(TASK_A01.NAME,&hMyTask);
    else
    {
        res=1;
        return res;
    }

    if( res==0 )
        res = DAQmxCreateAOVoltageChan(hMyTask,"PXI1Slot4/ao0:1","",min_Vout ,
            max_Vout ,DAQmx_Val_Volts,"");
    else
    {
        res=2;
        return res;
    }

    if( res==0 )
        res = DAQmxSetWriteAttribute(hMyTask,DAQmx_Write_RegenMode,
            DAQmx_Val_AllowRegen);
    else
    {
        res=3;
        return res;
    }

    if( res==0 )
        res = DAQmxCfgSampClkTiming(hMyTask,"",Fc_out ,DAQmx_Val_Rising ,
            DAQmx_Val_ContSamps,1);
        // Il buffer vale 1 perché dopo viene sovrascritto
    else
    {
        res=4;
        return res;
    }
}
```

```

/* ----- Creo il task per gli ingressi analogici -----*/
if( kMyTask == -1 )
    res = DAQmxCreateTask(TASK_A1_NAME,&kMyTask);
else
{
    res=5;
    return res;
}

if( res==0 )
    res = DAQmxCreateAIVoltageChan (kMyTask, "PXI1Slot2/ai0:1", "",
        DAQmx_Val_Cfg_Default, -10.0, 10.0, DAQmx_Val_Volts, NULL);
else
{
    res=6;
    return res;
}

if( res==0 )
    res = DAQmxCfgSampClkTiming(kMyTask,"",Fc_in,DAQmx_Val_Rising,DAQmx_Val_FiniteSamps,
        (uInt32)L_buffer_mic);
else
{
    res=7;
    return res;
}

/* ----- Allocazione della memoria -----*/
if( array_read == NULL )
    array_read=(double*)malloc(L_buffer_mic*NumChannels_in*sizeof(double));
if( array_mic1 == NULL )
    array_mic1=(double*)malloc(L_buffer_mic*sizeof(double));
if( array_mic2 == NULL )
    array_mic2=(double*)malloc(L_buffer_mic*sizeof(double));
if( spettro_real == NULL )
    spettro_real=(double*)malloc(L_buffer_mic*sizeof(double));
if( spettro_imag == NULL )
    spettro_imag=(double*)malloc(L_buffer_mic*sizeof(double));
if( modulo == NULL )
    modulo=(double*)malloc((L_buffer_mic/2+1)*sizeof(double));

//Allocazione temporanea della memoria per segnale (verrà riallocata)
if( segnale == NULL )
    segnale =(double *)malloc(memoria_alloc*sizeof(double));
if( A == NULL )
    A =(double *)malloc(n_freq_old*sizeof(double));
if( f == NULL )
    f =(double *)malloc(n_freq_old*sizeof(double));
if( phi == NULL )

```



```

    phi =(double *) malloc(n_freq_old*sizeof(double));

    /* ----- Inizializzazione degli array ----- */
    resetta(A_cont, 10);
    resetta(phi_cont, 10);
    resetta(A_cont_temp, 10);
    resetta(phi_cont_temp, 10);
    resetta(A_rum_temp, 10);
    resetta(AMP_0, 10);
    resetta(AMP_1, 10);
    resetta(AMP_2, 10);
    resetta(AMP_3, 10);
    resetta(grad_A, 10);
    resetta(grad_phi, 10);
    resetta(delta_A, 10);
    resetta(Pa2V, 10);

    aspetta=tc;

    for (j=0; j<10; j++)
    {
        k1_init[j]=k1[j];
        k2[j]=k1[j];

        condizione_AMP_singola[j]=0;
        salto_A[j]=0;
        salto_phi[j]=0;
    }
    return res;
}

/*-----*/
/* Funzione 2 - Ciclo_lento */
/*-----*/
double Ciclo_lento (int n_freq_rum, double *A_rum, double *f_rum,
    double *phi_rum, double *A_cont, double *phi_cont, double *out1,
    double *out2, int controllo_ON, int rumore_ON)
{
    int i,j,k;
    //i = contatore sulla lunghezza del buffer
    //j = contatore sul numero di frequenze
    //k = contatore sull'array delle uscite
    int res=0;

    double delta_phi=(5.0/180.0)*PI;
    double phi_cont_volt;
    double tempo_attesa_temp;

    n_freq_cont=n_freq_rum;

```

```

for (j=0; j<n_freq_rum; j++)
{
    f_cont[j]=f_rum[j];
    phi_rum[j]=(phi_rum[j]/180.0)*PI;
}
//nota: da ora le fasi si intendono tutte in radianti

passo_old=passo;
switch (passo)
{
case p_scrivoRum:

    if (n_freq_rum==0)
        passo=p_scrivoRum;
    else
    {
        Eseguo_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
            n_freq_cont, A_cont, f_cont, phi_cont);

        tempo_attesa=400;
        passo=p_attesaRum;
    }
    break;

case p_attesaRum:

    fz_attendi(tempo_attesa);
    break;

case p_leggoRum:

    res = Leggi(array_read, &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoRum;
    if (stato_lettura == 2)
        passo=p_leggoRum;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPrum;
    }
    break;

case p_AMPrum:

    for (j=0; j<n_freq_rum; j++)

```

```

{
    Calcolo_Ampiezza(array_mic2 , &AMP_0[j] , L_buffer_mic , f_rum[j]);

    /* ————— Calcolo dell'incremento delta_A ————— */
    FRF_cassa_controllo(f_cont[j] , 1.0 , phi_cont[j] , &Pa2V[j] ,
        &phi_cont_volt);

    delta_A[j]=(max_Vout/100.0)/Pa2V[j];
    A_cont_init=0.5;
    A_cont[j]=A_cont_init;

    Setting_parametri(f_rum[j] , &sat_A[j] , &sat_phi[j] , &k1[j] ,
        &k1_fin[j] , &tempo_attesa_temp);

    if (tempo_attesa_temp>tempo_attesa)
        tempo_attesa=tempo_attesa_temp;
}

for (j=0; j<6; j++)
{
    k=j*7;
    if (j<n_freq_rum)
    {
        out2[k]=k1[j];
        out2[k+1]=A_rum[j];
        out2[k+2]=phi_rum[j];
        out2[k+3]=f_rum[j];
        out2[k+4]=AMP_0[j];
        out2[k+5]=soglia_AMP;
        out2[k+6]=tempo_attesa;
    }
    else
    {
        out2[k]=0.0;
        out2[k+1]=0.0;
        out2[k+2]=0.0;
        out2[k+3]=0.0;
        out2[k+4]=0.0;
        out2[k+5]=0.0;
        out2[k+6]=0.0;
    }
}

step=0;
passo=p_scrivoCont;

break;

case p_scrivoCont:

```

```

    res2=Esegui_scrittura (n_freq_rum , A_rum , f_rum , phi_rum , n_freq_cont ,
        A_cont , f_cont , phi_cont);

    passo=p_attesaCont;
    break;

case p_attesaCont:

    fz_attendi(tempo_attesa);
    break;

case p_leggoCont:

    Leggi(array_read , &stato_lettura);

    if (stato_lettura == 1)
        passo=p_leggoCont;
    if (stato_lettura == 2)
        passo=p_leggoCont;
    if (stato_lettura == 0)
    {
        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPcont;
    }
    break;

case p_AMPcont:

    condizione_AMP_tot=1;

    for (j=0; j<n_freq_rum; j++)
    {
        Calcolo_Ampiezza(array_mic2 , &AMP_1[j] , L_buffer_mic , f_rum[j]);

        if (AMP_1[j]<(soglia_AMP))
            condizione_AMP_singola[j]=1;
        else
        {
            if (itero >15 & (AMP_1[j]<0.2*AMP_0[j] | AMP_1[j]<soglia_AMP))
                condizione_AMP_singola[j]=1;
            else
                passo=p_scrivoContDelta;
        }
        condizione_AMP_tot=condizione_AMP_tot*condizione_AMP_singola[j];
    }

    if (condizione_AMP_tot==1)

```

```
        passo=p_verificaAMP;

        break;

    case p_scrivoContDelta:

        for (j=0; j<n_freq_rum; j++)
        {
            if (condizione_AMP_singola[j]==0)
            {
                if (step==0)
                {
                    A_cont_temp[j]=A_cont[j];
                    phi_cont_temp[j]=phi_cont[j]+delta_phi;
                }
                else
                {
                    A_cont_temp[j]=A_cont[j]+delta_A[j];
                    phi_cont_temp[j]=phi_cont[j];
                }
            }
            else
            {
                A_cont_temp[j]=A_cont[j];
                phi_cont_temp[j]=phi_cont[j];
            }
        }

        Esegui_scrittura (n_freq_rum, A_rum, f_rum, phi_rum, n_freq_cont,
            A_cont_temp, f_cont, phi_cont_temp);

        passo=p_attesaContDelta;
        break;

    case p_attesaContDelta:

        fz_attendi(tempo_attesa);
        break;

    case p_leggoContDelta:

        Leggi(array_read, &stato_lettura);

        if (stato_lettura == 1)
            passo=p_leggoContDelta;
        if (stato_lettura == 2)
            passo=p_leggoContDelta;
        if (stato_lettura == 0)
        {
```

```

        for (i=0; i<L_buffer_mic; i++)
            array_mic2[i]=scala_mic*(array_read[i+(int)L_buffer_mic]);
        passo=p_AMPcontDelta;
    }
    break;

case p_AMPcontDelta:
    for (j=0; j<n_freq_rum; j++)
    {
        if (condizione_AMP_singola[j]==0)
        {
            if (step==0)
                Calcolo_Ampiezza(array_mic2, &AMP.2[j],
                                L_buffer_mic, f_rum[j]);
            else
                Calcolo_Ampiezza(array_mic2, &AMP.3[j],
                                L_buffer_mic, f_rum[j]);
        }
        else
        {
            AMP.2[j]=0.0;
            AMP.3[j]=0.0;
        }
    }
    passo=p_gradiente;
    break;

case p_gradiente:

    if (step==0)
    {
        /* ----- Calcolo del gradiente della fase -----*/
        for (j=0; j<n_freq_rum; j++)
        {
            if (condizione_AMP_singola[j]==0)
                grad_phi[j]=(AMP.2[j]-AMP.1[j])/delta_phi;
            else
                grad_phi[j]=0.0;
        }
        step=1;
        passo=p_scrivoContDelta;
    }
    else
    {
        /* ----- Calcolo del gradiente dell'ampiezza -----*/
        for (j=0; j<n_freq_rum; j++)
        {
            if (condizione_AMP_singola[j]==0)
            {

```

```

        grad_A[j]=(AMP_3[j]-AMP_1[j])/delta_A[j];
        Calcolo_costanti_algoritmo(k1, grad_phi, sat_phi,
            grad_A, sat_A, Pa2V, k2, salto_phi, salto_A, f_rum, j);
    }
    else
    {
        grad_A[j]=0.0;
        k1[j]=0.0;
        k2[j]=0.0;
    }

    /* ----- Generazione dei nuovi valori del controllo -----*/
    A_cont[j]=A_cont[j]-k2[j]*grad_A[j];
    phi_cont[j]=phi_cont[j]-k2[j]*grad_phi[j];

    grad_A_old[j]=grad_A[j];
    grad_phi_old[j]=grad_phi[j];
}
passo=p_verificaAMP;
step=0;

/* ----- Pubblicazione delle variabili -----*/
itero++;
for (j=0; j<6; j++)
{
    k=j*12;
    if (j<n_freq_rum)
    {
        out1[k]=itero;
        out1[k+1]=A_cont[j];
        out1[k+2]=phi_cont[j];
        out1[k+3]=AMP_1[j];
        out1[k+4]=A_cont[j]+delta_A[j];
        out1[k+5]=AMP_2[j];
        out1[k+6]=phi_cont[j]+delta_phi;
        out1[k+7]=AMP_3[j];
        out1[k+8]=grad_A[j];
        out1[k+9]=grad_phi[j];
        out1[k+10]=k2[j];
        out1[k+11]=condizione_AMP_singola[j];
    }
    else
    {
        out1[k]=0.0;
        out1[k+1]=0.0;
        out1[k+2]=0.0;
        out1[k+3]=0.0;
        out1[k+4]=0.0;
        out1[k+5]=0.0;
    }
}

```

```

        out1[k+6]=0.0;
        out1[k+7]=0.0;
        out1[k+8]=0.0;
        out1[k+9]=0.0;
        out1[k+10]=0.0;
        out1[k+11]=0.0;
    }
}
}
break;

case p_verificaAMP:

    if (itero > 50 || condizione_AMP_tot == 1)
    {
        passo = p_STOP;

        /* ----- Pubblicazione delle variabili ----- */
        itero++;
        for (j=0; j<6; j++)
        {
            k=j*12;
            if (j<n_freq_rum)
            {
                out1[k]=itero;
                out1[k+1]=A_cont[j];
                out1[k+2]=phi_cont[j];
                out1[k+3]=AMP_1[j];
                out1[k+4]=A_cont[j]+delta_A[j];
                out1[k+5]=AMP_2[j];
                out1[k+6]=phi_cont[j]+delta_phi;
                out1[k+7]=AMP_3[j];
                out1[k+8]=grad_A[j];
                out1[k+9]=grad_phi[j];
                out1[k+10]=k2[j];
                out1[k+11]=condizione_AMP_singola[j];
            }
            else
            {
                out1[k]=0.0;
                out1[k+1]=0.0;
                out1[k+2]=0.0;
                out1[k+3]=0.0;
                out1[k+4]=0.0;
                out1[k+5]=0.0;
                out1[k+6]=0.0;
                out1[k+7]=0.0;
                out1[k+8]=0.0;
                out1[k+9]=0.0;
            }
        }
    }
}

```



```

        out1[k+10]=0.0;
        out1[k+11]=0.0;
    }
}
    itero--;
}
else
    passo=p_scrivoCont;

break;
}

if ( passo==p_STOP )
{
    passo_old=1000;

    if ( flag_controllo_old != controllo_ON || flag_rumore_old != rumore_ON)
    {
        if ( !controllo_ON & rumore_ON)
        {
            resetta(A_cont_temp, n_freq_cont);
            Eseguo_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                n_freq_cont, A_cont_temp, f_cont, phi_cont);
        }
        if ( controllo_ON & rumore_ON)
        {
            Eseguo_scrittura (n_freq_rum, A_rum, f_rum, phi_rum,
                n_freq_cont, A_cont, f_cont, phi_cont);
        }
        if ( !rumore_ON & controllo_ON)
        {
            resetta(A_rum_temp, n_freq_rum);
            Eseguo_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                n_freq_cont, A_cont, f_cont, phi_cont);
        }
        if ( !rumore_ON & !controllo_ON)
        {
            resetta(A_rum_temp, n_freq_rum);
            resetta(A_cont_temp, n_freq_cont);
            Eseguo_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                n_freq_cont, A_cont_temp, f_cont, phi_cont);
        }
    }
    flag_controllo_old=controllo_ON;
    flag_rumore_old=rumore_ON;
}
return res2;
}

```

```
/*-----*/
/* Funzione 3 - Pulizia */
/*-----*/
int Pulizia (void)
{
    int res =0;
    int i=0;

    double n_freq_rum;
    double f_rum[10];
    double phi_rum[10];

    n_freq_rum=n_freq_cont;

    for (i=0; i<n_freq_rum; i++)
    {
        f_rum[i]=1000.0;
        phi_rum[i]=0.0;
    }

    resetta(A_rum_temp, n_freq_rum);
    resetta(A_cont_temp, n_freq_cont);

    Eseguo_scrittura (n_freq_rum, A_rum_temp, f_rum, phi_rum,
                    n_freq_cont, A_cont_temp, f_cont, phi_cont);

    if (hMyTask!=-1)
    {
        res = DAQmxClearTask(hMyTask);
        hMyTask=-1;
    }
    else
        res=1;

    if (res==0)
    {
        if (kMyTask!=-1)
        {
            res = DAQmxClearTask(kMyTask);
            kMyTask=-1;
        }
    }
    else
        res=2;

    passo=0;

    return res;
}
```

**File “cvidll.h”**

Si veda par. D.1.1.

**File “func\_lib.h”**

Si veda par. D.1.1.

**D.4.2 Analisi dettagliata delle funzioni del codice**

Di seguito vengono analizzate nel dettaglio le funzioni contenute all’interno del codice per l’esecuzione del sistema di controllo MFP.

**Funzione “Inizializzazione()”**

Questa funzione è identica a quella dei metodi di minimizzazione precedenti, si veda pertanto il paragrafo D.1.2.

**Funzione “Ciclo\_lento()”**

Questa funzione contiene il nocciolo dell’algoritmo di controllo secondo lo schema mostrato nel diagramma a blocchi in figura 3.76. Si analizzano di seguito le sole parti del codice differenti rispetto al metodo di minimizzazione di un rumore mono-armonico riportato nel paragrafo D.2.

- Nelle fasi 4), 7), 11) e 11’) viene valutato lo spettro del segnale letto dal microfono. Il segnale acquisito è stato memorizzato nelle fasi precedenti all’interno di un vettore, denominato  $a_{mic}$ . Lo spettro viene calcolato grazie alla funzione *Calcolo\_Spettro()* che riceve in ingresso il vettore  $a_{mic}$ . Le ampiezze dei picchi dello spettro appena ricavato vengono calcolate in modo ricorsivo per mezzo di un ciclo *for* che effettua la chiamata alla funzione *Calcolo\_ampiezza\_multi()*; questa riceve in ingresso l’array  $a_{mic}$  in cui è memorizzato il segnale appena acquisito, la sua dimensione  $N$ , l’array contenente i valori delle frequenze alle quali si vuole ridurre il rumore e la relativa dimensione dell’array, mentre in uscita fornisce il valore dell’ampiezza del picco in corrispondenza delle suddette frequenze.
- Nella fase 8) viene effettuato il confronto tra i valori attuali delle ampiezze dei picchi raggiunte alle differenti frequenze che compongono il segnale di rumore con i valori delle ampiezze di soglia (fig. 3.50). È stato fissato inoltre un numero

di iterazioni massimo (pari a 15) che consente di terminare l'algoritmo qualora questo non abbia ancora raggiunto il valore di soglia di rumore desiderato. In quest'ultimo caso l'algoritmo può terminare se si verifica uno dei seguenti casi:

- Il valore delle ampiezze di tutti i picchi raggiungono il 20 % del loro valore iniziale, ossia del valore di ampiezza che avevano quando nel sistema era presente solo il rumore.
- Viene raggiunta la soglia desiderata a tutte le frequenze.

In particolare il controllo viene effettuato per mezzo di un *flag*: vale 1 solamente se è stata raggiunta la condizione di minimizzazione su tutte le frequenze interessate, altrimenti assume valore pari a 0. Quindi se il *flag* vale 0 prosegue verso la fase successiva con i nuovi valori dei parametri di controllo, mentre se questo vale 1 l'algoritmo termina mantenendo i valori attuali dei parametri di controllo. Questi valori sono quelli che consentono di abbattere il rumore generato all'ingresso del condotto in corrispondenza delle diverse armoniche che costituiscono il segnale di rumore.

- Nelle fasi 9) e 9') dello schema a blocchi viene fornito un incremento alle variabili  $A_{cont}^{k,i}$  e  $\phi_{cont}^{k,i}$  necessario al calcolo del gradiente mediante la formulazione alle differenze finite mostrata nelle relazioni 2.62 e 2.63. Vengono incrementate, ricorsivamente, mediante un ciclo *for* tutte le frequenze interessate alla minimizzazione, in particolare tutte le frequenze i quali  $flag = 0$ .
- Nelle fasi 13) e 13') vengono calcolati i gradienti delle grandezze di controllo. A differenza dei metodi precedenti il calcolo viene effettuato in modo iterativo su tutte le frequenze che compongono il segnale di rumore, a meno delle frequenze dove è già stata raggiunta la condizione di minimo. In corrispondenza di queste frequenze viene imposto a zero il valore del gradiente, in modo da non modificare il nuovo valore delle grandezze di controllo. I gradienti vengono calcolati secondo le relazioni 2.62 e 2.63.
- Nella fase 14) dello schema a blocchi viene chiamata, per ciascuna frequenza interessata alla minimizzazione, la funzione *Calcolo\_costanti\_algoritmo()* che provvede a calcolare il nuovo valore del passo  $G_2^{k,i}$  da fornire per la valutazione dei nuovi parametri di controllo. Questa funzione fornisce anche il nuovo valore di  $G_1^{(k+1),i}$  qualora si realizzino le condizioni di saltellamento attorno alla soluzione.

Le relazioni matematiche per il calcolo del nuovo passo sono quelle riportate nel paragrafo D.2.2 al punto 14). Le grandezze necessarie all'implementazione della funzione suddetta vengono determinate con la funzione *Setting\_parametri()*, definita nei metodi di minimizzazione precedenti. In particolare, per quanto riguarda i valori delle soglie  $A_{SAT}$  e  $\phi_{SAT}$ , il valore del guadagno iniziale  $G_1^{k=0}$  e della soglia inferiore  $G_{1,MIN}$ , si sono utilizzati gli stessi valori riportati nel metodo di minimizzazione con l'analisi spettrale sulla singola frequenza.

- Nella fase 15) vengono calcolati i valori dei nuovi parametri di controllo secondo le relazioni 3.55 e 3.56.

### **Funzione “Pulizia()”**

Questa funzione è identica a quella del metodo ARMS, si veda pertanto il paragrafo D.1.2.



## APPENDICE E

# Script di MATLAB utilizzati per i sistemi risonanti

### E.1 Analisi della misura su compressore industriale

Viene di seguito riportato lo script per l'analisi delle misure svolte durante la campagna di misura su compressore industriale, descritta nel paragrafo 5.1.1.

```
clear all, close all, clc

%% ANALISI DATI COMPRESSORE

% dati della macchina
tau=1.2533333333333333;

%% Caricamento dati da analizzare
flag_exit=1; while flag_exit==1
    clc
    close all
    % Menù scelta gruppo
    opt_gruppo=menu('Scegli il gruppo da analizzare:' ,...
        'Gruppo 1','Gruppo 2','Gruppo 3','Gruppo 4','Gruppo 5',...
        'Gruppo 6','Gruppo 7','Gruppo 8','Gruppo 9');
    nome_file1=num2str(opt_gruppo);

    % Menù scelta pressione nominale
    opt_pnom=menu('Scegli la pressione nominale da analizzare:' ,...
        'Parzializzazione 100%','2000 kPa','4000 kPa','4000 kPa mantenuti');
    switch opt_pnom
        case 1
            nome_file2='0_0';
            pnom_text='0';
```

```

        parz_text='SI';
        pmant_text='NO';
    case 2
        nome_file2='20_0';
        pnom_text='2000';
        parz_text='NO';
        pmant_text='NO';
    case 3
        nome_file2='40_0';
        pnom_text='4000';
        parz_text='NO';
        pmant_text='NO';
    case 4
        nome_file2='40_1';
        pnom_text='4000';
        parz_text='NO';
        pmant_text='SI';
end

% Caricamento del file da analizzare
nome_file=sprintf('G%s_%s.mat',nome_file1,nome_file2);
files=dir;
check_file=0;
for ii=1:length(files)
    if (strcmp(files(ii).name,nome_file));
        check_file=1;
    end
end
if check_file==1
    load(nome_file);

    %% Analisi del file
    % (all'interno dell'if)

    % vengono analizzati:
    % - numero giri reali nel ciclo
    % - numero giri medi
    % - temperature, kw, cos(phi)
    % - pressioni medie
    % - spettri segnali non mediati
    % - spettri segnali mediati
    % - pressione nominale, parzializzazione

    opt_nprove=menu('Vuoi visualizzare i risultati per tutti i giri motore?:','SI','NO');
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if opt_nprove==1
        close all
        flag_no_vis=1;
    end
end

```



```
while flag_no_vis==1
    opt_visu_multi_rpm=menu('Cosa vuoi visualizzare?', 'Spettrogrammi', ...
    'RMS', 'Altre grandezze', 'Esci');
    switch opt_visu_multi_rpm
        case 1
            visu_specgram
        case 2
            visu_RMS
        case 3
            visu_grandezze_aggregate
        case 4
            flag_no_vis=0;
    end
end

elseif opt_nprove==2
    % menu di scelta del numero di giri da analizzare
    opt_giri=menu('Scegli il numero di giri nominale da analizzare:', ...
    '546 rpm', '608 rpm', '670 rpm', '732 rpm', '794 rpm', '856 rpm', '918 rpm', ...
    '980 rpm', '1042 rpm', '1104 rpm', '1166 rpm', '1228 rpm', '1290 rpm', ...
    '1352 rpm', '1414 rpm', '1476 rpm');

    switch opt_giri
        case 1
            dati_rpm=dati.rpm0;
            rpm_nom_text='546';
        case 2
            dati_rpm=dati.rpm1;
            rpm_nom_text='608';
        case 3
            dati_rpm=dati.rpm2;
            rpm_nom_text='670';
        case 4
            dati_rpm=dati.rpm3;
            rpm_nom_text='732';
        case 5
            dati_rpm=dati.rpm4;
            rpm_nom_text='794';
        case 6
            dati_rpm=dati.rpm5;
            rpm_nom_text='856';
        case 7
            dati_rpm=dati.rpm6;
            rpm_nom_text='918';
        case 8
            dati_rpm=dati.rpm7;
            rpm_nom_text='980';
        case 9
            dati_rpm=dati.rpm8;
```

```

        rpm_nom_text='1042';
    case 10
        dati_rpm=dati.rpm9;
        rpm_nom_text='1104';
    case 11
        dati_rpm=dati.rpm10;
        rpm_nom_text='1166';
    case 12
        dati_rpm=dati.rpm11;
        rpm_nom_text='1228';
    case 13
        dati_rpm=dati.rpm12;
        rpm_nom_text='1290';
    case 14
        dati_rpm=dati.rpm13;
        rpm_nom_text='1352';
    case 15
        dati_rpm=dati.rpm14;
        rpm_nom_text='1414';
    case 16
        dati_rpm=dati.rpm15;
        rpm_nom_text='1476';
end

rpm_medi

disp_dati

% scalo tutti i dati in pressione in kPa
dati_rpm.pres(:,2:end)=dati_rpm.pres(:,2:end)/1000;

flag_no_vis=1;
while flag_no_vis==1
    opt_visualizza_1rpm=menu('Cosa vuoi visualizzare?',...
        'RPM nel ciclo','Spettri',...
        'Spettri con frequenze normalizzate',...
        'Segnali nel tempo','Fasi tra i segnali',...
        'Spettri raggruppati',...
        'Spettri raggruppati con frequenze normalizzate',...
        'Spettrogramma della singola storia',...
        'Spettrogramma singola storia frequenze normalizzate',...
        'Segnali nel tempo sul giro',...
        'Esci');
    switch opt_visualizza_1rpm
        case 1
            rpm_ciclo
        case 2
            visu_spettri
        case 3

```

```

        visu_spettri_norm
    case 4
        visu_tempo
    case 5
        visu_fasi
    case 6
        visu_spettri_group
    case 7
        visu_spettri_group_norm
    case 8
        visu_specgram_1storia
    case 9
        visu_specgram_1storia_norm
    case 10
        visu_tempo_giro
    case 11
        flag_no_vis=0;
    end
end
end %end if optprove
else
    clc
    disp('La prova non è stata effettuata')
end %end if check_file
flag_exit=menu('Vuoi analizzare un''altra prova?:' ,...
    'SI','NO');

end %end while flag_exit

```

### Script “visu specgram”

Script per la visualizzazione degli spettrogrammi.

```

load('MyColormaps','mycmap')
%analisi spettrale e altre grandezze
for ii=1:8
    clear pres_rpm
    for jj=1:16
        nome_dato=sprintf('dati.rpm%d',jj-1);
        dati_punto=eval(nome_dato);
        dati_punto.pres(:,2:end)=dati_punto.pres(:,2:end)/1000; %scalo da Pa a kPa

        % preparazione dei dati aggregati
        if jj==1
            pres_rpm(:,jj)=dati_punto.pres(:,ii+1);
            N_rpm=length(dati_punto.pres);
        else
            N=length(dati_punto.pres);
        end
    end
end

```

```

    Nmin=min(N,N_rpm);
    pres_rpm=pres_rpm(1:Nmin,:);
    pres_rpm(1:Nmin,jj)=dati_punto.pres(1:Nmin,ii+1);
    N_rpm=Nmin;
end

% analisi altre grandezze
if ii==1
    nome_dato=sprintf('dati.rpm%d',jj-1);
    dati_punto=eval(nome_dato);
    I=find(diff(dati_punto.pres(:,1))<-10);
    rpm_real=1./(diff(I)./dati.fc)*60;
    rpm_mean_albero(jj)=mean(rpm_real);
    rpm_mean_mot(jj)=rpm_mean_albero(jj)*tau;
end
end

% spettrogramma frequenze, rpm, pressioni
N=length(pres_rpm);
if (N/2-floor(N/2))>0
    N=N-1;
end
Tacq=1/dati.fc*N;
df=1/Tacq;
spettro=fft(pres_rpm,[],1);
mod=abs(spettro(1:N/2,:));
mod(1,:)=mod(1,:)/N;
mod(2:N/2,:)=mod(2:N/2,:)/(N/2);
freq=0:df:(N/2-1)*df;
nome_fig=sprintf('Spettrogramma punto %d',dati.pos(ii));
figure('Name',nome_fig,'NumberTitle','off')
fig=gcf;
surf(freq,rpm_mean_albero,(mod')./max(max(abs(mod))), 'EdgeColor','none');
%shading interp
set(gca,'YTick',round(rpm_mean_albero));
colormap(jet);
view(0,90);
grid on
nome_title=sprintf('Spettrogramma punto %d [kPa/kPa]',dati.pos(ii));
title(nome_title,'Interpreter','Latex')
xlabel('Frequenze $[Hz]$', 'Interpreter','Latex')
ylabel('RPM albero compressore', 'Interpreter','Latex')
colorbar
set(fig,'Colormap',mycmap);
axis([0 200, rpm_mean_albero(1) rpm_mean_albero(end)]);
hold on
%disegno le frequenze teoriche
for kk=1:20
    f_teor=rpm_mean_albero/60*kk;
    h=ones(1,length(rpm_mean_albero))*max(max(mod));

```

```

        plot3(f_teor,rpm_mean_albero,h,'r-',f_teor,rpm_mean_albero,h,'or')
        %plot3(f_teor,rpm_mean_albero,h,'r-')
    end

    %% nota, per settare la colorbar
    % colormapeditor % setto i parametri della colorbar
    % h=gcf; % prendo l'handle della digura corrente
    % mycmap = get(h,'Colormap') % prendo la variabile della colormap
    % save('MyColormaps','mycmap') % salvo la colormap
    % load('MyColormaps','mycmap') % leggo la colormap
    % set(fig,'Colormap',mycmap) % applico la colormap alla figura
    % con l'handle fig
    %
end % end for ii analisi spettrale.

```

### Script “visu RMS”

Script per la visualizzazione dell'andamento dell'RMS del segnale di pressione.

```

% analisi RMS
opt_rms_perc=menu('Vuoi visualizzare anche l''RMS scalato alla
pressione media?', 'SI', 'NO');

%clear pres_rpm
for ii=1:8
    clear rms rms_perc pres_rpm
    for jj=1:16
        nome_dato=sprintf('dati.rpm%d',jj-1);
        dati_punto=eval(nome_dato);
        dati_punto.pres(:,2:end)=dati_punto.pres(:,2:end)/1000; %scalo da Pa a kPa

        % preparazione dei dati aggregati
        if jj==1
            pres_rpm(:,jj)=dati_punto.pres(:,ii+1);
            N_rpm=length(dati_punto.pres);
        else
            N=length(dati_punto.pres);
            Nmin=min(N,N_rpm);
            pres_rpm=pres_rpm(1:Nmin,:);
            pres_rpm(1:Nmin,jj)=dati_punto.pres(1:Nmin,ii+1);
            N_rpm=Nmin;
        end

        if ii==1
            pmed_rpm(:,jj)=(dati_punto.pmed')*100; %scalo i dati da bar a kPa
            I=find(diff(dati_punto.pres(:,1))<-10);
            rpm_real=1./(diff(I)./dati.fc)*60;
            rpm_mean_albero(jj)=mean(rpm_real);
        end
    end
end

```

```

end

N=length(pres_rpm);

% rms in funzione degli rpm
rms=sqrt(sum(pres_rpm.^2,1)/N);
nome_fig=sprintf('RMS punto %d',dati.pos(ii));
figure('Name',nome_fig,'NumberTitle','off')
plot(rpm_mean_albero,rms,'o'), grid on
nome_title=sprintf('RMS punto %d [kPa]',dati.pos(ii));
title(nome_title,'Interpreter','Latex')
xlabel('RPM albero compressore','Interpreter','Latex')
ylabel('RMS $[kPa]$', 'Interpreter','Latex')

if opt_rms_perc==1
    % rms in rapporto alla pressione media
    if (dati.pos(ii)>=11 && dati.pos(ii)<=19)
        pmed_rms=pmed_rpm(1,:);
    elseif (dati.pos(ii)>=21 && dati.pos(ii)<=28)
        pmed_rms=pmed_rpm(2,:);
    elseif (dati.pos(ii)>=31 && dati.pos(ii)<=34)
        pmed_rms=pmed_rpm(3,:);
    end
    rms_perc=rms./pmed_rms*100;
    nome_fig=sprintf('RMS%% punto %d',dati.pos(ii));
    figure('Name',nome_fig,'NumberTitle','off')
    plot(rpm_mean_albero,rms_perc,'o'), grid on
    nome_title=sprintf('RMS\\%% punto %d [kPa]',dati.pos(ii));
    title(nome_title,'Interpreter','Latex')
    xlabel('RPM albero compressore','Interpreter','Latex')
    ylabel('RMS $\\%[P_{med}]$', 'Interpreter','Latex')
end
end % end for ii analisi rms

```

## Script “visu grandezze aggregate”

Script per la visualizzazione di grandezze aggregate come pressioni medie, temperature medie, potenza elettrica assorbita e sfasamento tra tensione e corrente.

```

for jj=1:16
    nome_dato=sprintf('dati.rpm%d',jj-1);
    dati_punto=eval(nome_dato);
    pmed_rpm(:,jj)=(dati_punto.pmed')*100; %scalo i dati da bar a kPa
    T_rpm(:,jj)=dati_punto.T';
    kW_rpm(jj)=dati_punto.kW;
    cos_phi_rpm(jj)=dati_punto.cos_phi;

    I=find(diff(dati_punto.pres(:,1))<-10);

```

```

rpm_real=1./(diff(I)./dati.fc)*60;
rpm_mean_albero(jj)=mean(rpm_real);
rpm_mean_mot(jj)=rpm_mean_albero(jj)*tau;
end

%rappresentazione risultati altre grandezze
if (isnan(pmed_rpm(1))==0)
    figure('Name','Pressioni medie','NumberTitle','off')
    plot(rpm_mean_albero,pmed_rpm)
    grid on
    title('Pressioni medie','Interpreter','Latex')
    xlabel('RPM albero compressore','Interpreter','Latex')
    ylabel('$P_{med}$ $[kPa]$', 'Interpreter','Latex')
    leg=legend('$P_1$', '$P_2$', '$P_{3,1}$', '$P_{3,2}$');
    set(leg,'Interpreter','Latex');
    set(leg,'Location','NorthWest');
    xlim([rpm_mean_albero(1) rpm_mean_albero(end)])
end

if (isnan(T_rpm(1))==0)
    figure('Name','Temperature medie','NumberTitle','off')
    plot(rpm_mean_albero,T_rpm)
    grid on
    title('Temperature medie','Interpreter','Latex')
    xlabel('RPM albero compressore','Interpreter','Latex')
    ylabel('$T_{med}$ $[^\circ C]$', 'Interpreter','Latex')
    leg=legend('$T_1$', '$T_2$', '$T_3$');
    set(leg,'Interpreter','Latex');
    set(leg,'Location','NorthEastOutside');
    xlim([rpm_mean_albero(1) rpm_mean_albero(end)])
end

if (isnan(kW_rpm(1))==0)
    figure('Name','Potenza assorbita','NumberTitle','off')
    plot(rpm_mean_albero,kW_rpm)
    grid on
    title('Potenza assorbita','Interpreter','Latex')
    xlabel('RPM albero compressore','Interpreter','Latex')
    ylabel('$Pot_{med}$ $[kW]$', 'Interpreter','Latex')
    xlim([rpm_mean_albero(1) rpm_mean_albero(end)])
end

if (isnan(cos_phi_rpm(1))==0)
    figure('Name','Cos phi','NumberTitle','off')
    plot(rpm_mean_albero,cos_phi_rpm)
    grid on
    title('Sfasamento tra tensione e corrente','Interpreter','Latex')
    xlabel('RPM albero compressore','Interpreter','Latex')
    ylabel('$\cos \phi$', 'Interpreter','Latex')
end

```

```

    xlim([rpm_mean_albero(1) rpm_mean_albero(end)])
end

```

## Script “rpm ciclo”

Script per la visualizzazione dell’andamento degli RPM istanei dell’albero del compressore.

```

% calcolo gli rpm all'interno del ciclo
indici=find(diff(dati_rpm.pres(:,1))<-10);
N=length(dati_rpm.pres(:,1)); complemento_giro=0; for ii=1:N
    giri_cum(ii)=dati_rpm.pres(ii,1)+complemento_giro;
    if (sum(ii==indici))
        complemento_giro=complemento_giro+360;
    end
end

% filtro per posizione encoder
% filtro con f_cut=10*f_rot, la frequenza di rotazione dell'albero
% compressore.

% Tutte le frequenze sono in Hz.
Fs = 4096; % Sampling Frequency

% Fpass = round(9*rpm_mean_albero/60); % Passband Frequency
% Fstop = round(10*rpm_mean_albero/60); % Stopband Frequency
% Apass = 1; % Passband Ripple (dB)
% Astop = 30; % Stopband Attenuation (dB)
% match = 'stopband'; % Band to match exactly
% h = fdesign.lowpass(Fpass, Fstop, Apass, Astop, Fs);
% Hd = design(h, 'butter', 'MatchExactly', match);

% altro filtro
N = 3; % Order
Fcut = round(10*rpm_mean_albero/60); % Cutoff Frequency

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.lowpass('N,Fc', N, Fcut, Fs); Hd = design(h,
'butter');

dati_filt=filter(Hd,giri_cum);

rpm_ist=dati.fc./360*diff(dati_filt)*60;
tempo_rpm_ist=0:1/dati.fc:(length(dati_filt)-2)/dati.fc;
figure('Name','RPM istantanei','NumberTitle','off')
plot(tempo_rpm_ist,rpm_ist) grid on title('RPM istantanei albero
compressore','Interpreter','Latex') xlabel('Tempo
[s]','Interpreter','Latex') ylabel('RPM','Interpreter','Latex')

```



```

axis([0 tempo_rpm_ist(end) rpm_mean_albero*0.9
rpm_mean_albero*1.1])

N=length(rpm_ist);

if (N/2-floor(N/2))>0
    N=N-1;
end
Tacq=1/dati.fc*N;
df=1/Tacq;
spettro=fft(rpm_ist);
mod=abs(spettro(1:N/2));
mod(1)=mod(1)/N;
mod(2:N/2)=mod(2:N/2)/(N/2);
freq=0:df:(N/2-1)*df;
nome_fig=sprintf('Spettro RPM istantanei');
figure('Name',nome_fig,'NumberTitle','off')
plot(freq,mod)
grid on
title_label_freq=sprintf('Spettro velocità di rotazione');
title(title_label_freq,'Interpreter','Latex');
xlabel('Frequenze [Hz]','Interpreter','Latex')
ylabel('Velocità [RPM]','Interpreter','Latex')
%xlim([0 200])

```

### Script “visu spettri”

Script per la visualizzazione degli spettri dei segnali in pressione.

```

% calcolo gli spettri dei segnali in pressione
N=length(dati_rpm.pres); tempo=0:1/dati.fc:(N-1)/dati.fc; for
ii=1:8
    if (N/2-floor(N/2))>0
        N=N-1;
    end
    Tacq=1/dati.fc*N;
    df=1/Tacq;
    spettro=fft(dati_rpm.pres(:,ii+1));
    mod=abs(spettro(1:N/2));
    mod(1)=mod(1)/N;
    mod(2:N/2)=mod(2:N/2)/(N/2);
    freq=0:df:(N/2-1)*df;
    nome_fig=sprintf('Spettro punto %d',dati.pos(ii));
    figure('Name',nome_fig,'NumberTitle','off')
    plot(freq,mod./max(abs(mod)))
    grid on
    title_label_freq=sprintf('Spettro segnale pressione nel punto %d',dati.pos(ii));

```

```

title(title_label_freq,'Interpreter','Latex');
xlabel('Frequenze $[Hz]$', 'Interpreter','Latex')
ylabel('Pressione $[kPa/kPa]$', 'Interpreter','Latex')
xlim([0 200])
ylim([0 1.1])
end

```

### Script “visu spettri norm”

Script per la visualizzazione dell'andamento degli spettri dei segnale in pressione, normalizzati alla componente di rotazione dell'albero del compressore.

```

% calcolo gli spettri dei segnali in pressione scalati alla componente 1x
% giro
N=length(dati_rpm.pres); tempo=0:1/dati.fc:(N-1)/dati.fc; for
ii=1:8
    if (N/2-floor(N/2))>0
        N=N-1;
    end
    Tacq=1/dati.fc*N;
    df=1/Tacq;
    spettro=fft(dati_rpm.pres(:,ii+1));
    mod=abs(spettro(1:N/2));
    mod(1)=mod(1)/N;
    mod(2:N/2)=mod(2:N/2)/(N/2);
    freq=0:df:(N/2-1)*df;
    freq_norm=freq./f_teor(1);
    nome_fig=sprintf('Spettro punto %d',dati.pos(ii));
    figure('Name',nome_fig,'NumberTitle','off')
    plot(freq_norm,mod./max(abs(mod)),50/f_teor(1),1,'ro')
    grid on
    title_label_freq=sprintf('Spettro segnale pressione nel punto %d',dati.pos(ii));
    title(title_label_freq,'Interpreter','Latex');
    xlabel('Frequenze 1 x giro $[Hz/Hz]$', 'Interpreter','Latex')
    ylabel('Pressione $[kPa/kPa]$', 'Interpreter','Latex')
    xlim([0 8])
    ylim([0 1.1])
end

```

### Script “visu tempo”

Script per la visualizzazione dell'andamento dei segnali in pressione nel tempo.

```

for ii=1:8
    tempo=0:1/dati.fc:(length(dati_rpm.pres)-1)/dati.fc;
    nome_fig=sprintf('Storia temporale punto %d',dati.pos(ii));
    figure('Name',nome_fig,'NumberTitle','off')
    plot(tempo,dati_rpm.pres(:,ii+1)./max(abs(dati_rpm.pres(:,ii+1))))
end

```

```

grid on
title_label_freq=sprintf('Storia temporale nel punto %d',dati.pos(ii));
title(title_label_freq,'Interpreter','Latex');
xlabel('Tempo [s]', 'Interpreter','Latex')
ylabel('Pressione [kPa/kPa]', 'Interpreter','Latex')
xlim([0 tempo(end)])
end

```

## Script “visu fasi”

Script per la visualizzazione delle fasi tra le componenti spettrali di segnali in pressione di punti di misura diversi.

```

opt_continua=1; while opt_continua==1
    opt_punto1=menu('Seleziona il primo punto:',num2str(dati.pos(1)),num2str(dati.pos(2)),...
        num2str(dati.pos(3)),num2str(dati.pos(4)),num2str(dati.pos(5)),num2str(dati.pos(6)),...
        num2str(dati.pos(7)),num2str(dati.pos(8)));
    opt_punto2=menu('Seleziona il secondo punto:',num2str(dati.pos(1)),num2str(dati.pos(2)),...
        num2str(dati.pos(3)),num2str(dati.pos(4)),num2str(dati.pos(5)),num2str(dati.pos(6)),...
        num2str(dati.pos(7)),num2str(dati.pos(8)));

    N=length(dati_rpm.pres);
    if (N/2-floor(N/2))>0
        N=N-1;
    end
    Tacq=1/dati.fc*N;
    df=1/Tacq;
    freq=0:df:(N/2-1)*df;
    spettroin=fft(dati_rpm.pres(:,opt_punto1+1));
    spettroout=fft(dati_rpm.pres(:,opt_punto2+1));
    %calcolo fdt
    fdt=spettroout./spettroin;
    modulo0=abs(fdt(1:N/2));
    fasi0=angle(fdt(1:N/2));
    %calcolo crosspettri
    Sxy=conj(spettroin).*spettroout;
    Syx=conj(spettroout).*spettroin;
    Sxx=conj(spettroin).*spettroin;
    Syy=conj(spettroout).*spettroout;
    %calcolo H1
    H1=Sxy./Sxx;
    modulo1=abs(H1(1:N/2));
    fasi1=angle(H1(1:N/2));
    %calcolo H2
    H2=Syy./Syx;
    modulo2=abs(H2(1:N/2));
    fasi2=angle(H2(1:N/2));
end

```

```

% calcolo funzione di coerenza
gammaxy=abs(Sxy).^2./(Sxx.*Syy);

figure('Name','FdT da spettri','NumberTitle','off')
subplot(2,1,1), plot(freq,modulo0), grid on
nome_title=sprintf('FdT tra i punti %d e %d',dati.pos(opt_punto1),dati.pos(opt_punto2));
title(nome_title,'Interpreter','Latex')
xlabel('Frequenza $[Hz]$', 'Interpreter','Latex')
ylabel('Modulo $[kPa/kPa]$', 'Interpreter','Latex')
xlim([0 200])
subplot(2,1,2), plot(freq,fasi0), grid on
xlabel('Frequenza $[Hz]$', 'Interpreter','Latex')
ylabel('Fase $[rad]$', 'Interpreter','Latex')
xlim([0 200])

figure('Name','FdT con H1','NumberTitle','off')
subplot(2,1,1), plot(freq,modulo1), grid on
nome_title=sprintf('FdT tra i punti %d e %d - stimatore H1',dati.pos(opt_punto1),dati.pos(opt_punto2));
title(nome_title,'Interpreter','Latex')
xlabel('Frequenza $[Hz]$', 'Interpreter','Latex')
ylabel('Modulo $[kPa/kPa]$', 'Interpreter','Latex')
xlim([0 200])
subplot(2,1,2), plot(freq,fasi1), grid on
xlabel('Frequenza $[Hz]$', 'Interpreter','Latex')
ylabel('Fase $[rad]$', 'Interpreter','Latex')
xlim([0 200])

figure('Name','FdT con H1','NumberTitle','off'),
subplot(2,1,1), plot(freq,modulo2), grid on
nome_title=sprintf('FdT tra i punti %d e %d - stimatore H2',dati.pos(opt_punto1),dati.pos(opt_punto2));
title(nome_title,'Interpreter','Latex')
xlabel('Frequenza $[Hz]$', 'Interpreter','Latex')
ylabel('Modulo $[kPa/kPa]$', 'Interpreter','Latex')
xlim([0 200])
subplot(2,1,2), plot(freq,fasi2), grid on
xlabel('Frequenza $[Hz]$', 'Interpreter','Latex')
ylabel('Fase $[rad]$', 'Interpreter','Latex')
xlim([0 200])
figure('Name','Funzione di coerenza','NumberTitle','off')
plot(freq,gammaxy(1:N/2)), grid on
title('Funzione di coerenza','Interpreter','Latex')
xlabel('Frequenza $[Hz]$', 'Interpreter','Latex')
ylabel('$\gamma^2_{xy}$', 'Interpreter','Latex')
axis([0 freq(end) 0 1.5])

opt_continua=menu('Vuoi visualizzare un''altra fase?','SI','NO');
end

```

**Script “visu spettri group”**

Script per la visualizzazione dell'andamento degli spettri di pressione raggruppati per i vari regimi di velocità di rotazione dell'albero del compressore.

```
% calcolo gli spettri raggruppati dei segnali in pressione
load('MyColormaps','mycmap') N=length(dati_rpm.pres); if
(N/2-floor(N/2))>0
    N=N-1;
end Tacq=1/dati.fc*N; df=1/Tacq;
spettro=fft(dati_rpm.pres(:,2:end),[],1);
mod=abs(spettro(1:N/2,:)); mod(1,:)=mod(1,:)/N;
mod(2:N/2,:)=mod(2:N/2,:)/(N/2); freq=0:df:(N/2-1)*df;

nome_fig=sprintf('Spettri raggruppati %s rpm',rpm_nom_text);
figure('Name',nome_fig,'NumberTitle','off') fig=gcf;
surf(freq,[1:8],mod,'EdgeColor','none');
%shading interp
set(gca,'YTickLabel',dati.pos);
% colormap(jet);
view(0,90); grid on
nome_title=sprintf('Spettri raggruppati a %s rpm, %[kPa]$',rpm_nom_text);
title(nome_title,'Interpreter','Latex') xlabel('Frequenze
$[Hz]$', 'Interpreter','Latex') ylabel('Punti
rilevati', 'Interpreter','Latex') colorbar
set(fig,'Colormap',mycmap); axis([0 200, 1 8]); hold on

nome_fig=sprintf('Spettri raggruppati %s rpm',rpm_nom_text);
figure('Name',nome_fig,'NumberTitle','off') fig=gcf; hold on for
ii=1:8
    plot3(freq,ones(size(freq))*(ii),mod(:,ii))
end grid on set(gca,'YTick',1:8,'YTickLabel',dati.pos);
% colormap(jet);
view(45,45); grid on
nome_title=sprintf('Spettri raggruppati a %s rpm, %[kPa]$',rpm_nom_text);
title(nome_title,'Interpreter','Latex') xlabel('Frequenze
$[Hz]$', 'Interpreter','Latex') ylabel('Punti
rilevati', 'Interpreter','Latex') colorbar
set(fig,'Colormap',mycmap); axis([0 200, 1 8]); hold on
```

**Script “visu spettri group norm”**

Script per la visualizzazione dell'andamento degli spettri di pressione raggruppati per i vari regimi di velocità di rotazione dell'albero del compressore, normalizzati alla velocità di rotazione.

```

% calcolo gli spettri raggruppati e scalati alla frequenza 1 x giro
load('MyColormaps','mycmap') N=length(dati_rpm.pres); if
(N/2-floor(N/2))>0
    N=N-1;
end Tacq=1/dati.fc*N; df=1/Tacq;
spettro=fft(dati_rpm.pres(:,2:end),[],1);
mod=abs(spettro(1:N/2,:)); mod(1,:)=mod(1,+)/N;
mod(2:N/2,:)=mod(2:N/2,+)/N; freq=0:df:(N/2-1)*df;
freq_norm=freq./f_teor(1);

nome_fig=sprintf('Spettri raggruppati %s rpm',rpm_nom_text);
figure('Name',nome_fig,'NumberTitle','off') fig=gcf;
surf(freq_norm,[1:8],mod','EdgeColor','none');
%shading interp
set(gca,'YTickLabel',dati.pos);
% colormap(jet);
view(0,90); grid on
nome_title=sprintf('Spettri raggruppati a %s rpm',rpm_nom_text);
title(nome_title,'Interpreter','Latex') xlabel('Frequenze 1 x giro
$[Hz/Hz]$', 'Interpreter','Latex') ylabel('Punti
rilevati','Interpreter','Latex') colorbar
set(fig,'Colormap',mycmap); axis([0 8, 1 8]); hold on

nome_fig=sprintf('Spettri raggruppati %s rpm',rpm_nom_text);
figure('Name',nome_fig,'NumberTitle','off') fig=gcf; hold on for
ii=1:8
    plot3(freq_norm,ones(size(freq))*(ii),mod(:,ii))
end grid on set(gca,'YTick',1:8,'YTickLabel',dati.pos);
% colormap(jet);
view(45,45); grid on
nome_title=sprintf('Spettri raggruppati a %s rpm',rpm_nom_text);
title(nome_title,'Interpreter','Latex') xlabel('Frequenze 1 x giro
$[Hz/Hz]$', 'Interpreter','Latex') ylabel('Punti
rilevati','Interpreter','Latex') colorbar
set(fig,'Colormap',mycmap); axis([0 8, 1 8]); hold on

```

### Script “visu specgram 1 storia”

Script per la visualizzazione dello spettrogramma di un record di acquisizione del segnale in pressione.

```

%calcolo lo spettrogramma di una singola storia
load('MyColormaps','mycmap') N=length(dati_rpm.pres);
tempo_window=1;

window=ones(1,floor(tempo_window*dati.fc));
noverlap=floor(0.9*length(window)); F1=0:1:200; for ii=1:8
    if (N/2-floor(N/2))>0

```

```

        N=N-1;
    end
    Tacq=1/dati.fc*N;
    df=1/Tacq;
    [S,F,T] = spectrogram(dati_rpm.pres(:,ii+1),window,noverlap,F1,dati.fc);
    mod=abs(S)./length(window);
    nome_fig=sprintf('Spettrogramma punto %d',dati.pos(ii));
    figure('Name',nome_fig,'NumberTitle','off')
    fig=gcf;
    surf(F,T,(mod')./max(max(abs(mod))), 'EdgeColor','none');
    colormap(jet); view(0,90);
    nome_title=sprintf('Spettrogramma punto %d $[kPa/kPa]$',dati.pos(ii));
    title(nome_title,'Interpreter','Latex')
    ylabel('Tempo $[s]$', 'Interpreter','Latex');
    xlabel('Frequenze $[Hz]$', 'Interpreter','Latex');
    grid on
    colorbar
    xlim([F(1) F(end)])
    ylim([T(1) T(end)])
%     set(fig,'Colormap',mycmap);
end

```

### Script “visu specgram 1 storia norm”

Script per la visualizzazione dello spettrogramma di un record di acquisizione del segnale in pressione, normalizzato alla velocità di rotazione dell’albero del compressore.

```

%calcolo lo spettrogramma di una singola storia
%load('MyColormaps','mycmap')
N=length(dati_rpm.pres); tempo_window=1;

window=ones(1,floor(tempo_window*dati.fc));
noverlap=floor(0.9*length(window)); F1=0:1:200; for ii=1:8
    if (N/2-floor(N/2))>0
        N=N-1;
    end
    Tacq=1/dati.fc*N;
    df=1/Tacq;
    [S,F,T] = spectrogram(dati_rpm.pres(:,ii+1),window,noverlap,F1,dati.fc);
    F_norm=F/f_teor(1);
    mod=abs(S)./length(window);
    nome_fig=sprintf('Spettrogramma punto %d',dati.pos(ii));
    figure('Name',nome_fig,'NumberTitle','off')
    fig=gcf;
    surf(F_norm,T,mod', 'EdgeColor','none');
    colormap(jet);
    view(0,90);
    nome_title=sprintf('Spettrogramma punto %d $[kPa]$',dati.pos(ii));

```

```

title(nome_title,'Interpreter','Latex')
ylabel('Tempo [s]', 'Interpreter','Latex');
xlabel('Frequenze 1 x giro [Hz/Hz]', 'Interpreter','Latex');
grid on
colorbar
xlim([F_norm(1) F_norm(end)])
ylim([T(1) T(end)])
%set(fig,'Colormap',mycmap);
end

```

### Script “visu tempo giro”

Script per l’analisi delle storie temporali mediante media sul giro di rotazione dell’albero del compressore.

```

% analisi delle storie temporali mediate sul giro
% suddivido la storia in sottorecord da 1 giro
I=find((diff(dati_rpm.pres(:,1))<(-300)); count=1;
n_record=length(I)-1; l_min_record=min(diff(I)); for ii=1:8
    record_med=0;
    for jj=1:n_record; %medio il segnale
        record_temp=dati_rpm.pres(I(jj):I(jj+1),ii+1);%I(jj)+1:I(jj+1),ii+1);
        record_temp=record_temp(1:l_min_record); %ritaglio il segnale sul record più piccolo
        record_med=record_med+record_temp;
        deg_temp=dati_rpm.pres(I(jj):I(jj+1),1);
    end
    record_med=record_med./n_record;
    gradi=dati_rpm.pres(I(1)+1:I(2),1);
    gradi=gradi(1:l_min_record);
    nome_fig=sprintf('Storia sul giro punto %d',dati.pos(ii));
    figure('Name',nome_fig,'NumberTitle','off')
    plot(gradi,record_med./max(abs(record_med)))
    grid on
    title_label_freq=sprintf('Pressione nell''angolo giro, nel punto %d',dati.pos(ii));
    title(title_label_freq,'Interpreter','Latex');
    xlabel('Rotazione [deg]', 'Interpreter','Latex')
    ylabel('Pressione [kPa/kPa]', 'Interpreter','Latex')
    xlim([0 gradi(end)])
end

```

## E.2 Script MatLab modello condotto e risonatore di Helmholtz

Script di MatLab per la determinazione dell’andamento delle pressioni e delle portate dinamiche lungo il condotto principale in assenza e in presenza di un risonatore di Helmholtz.



```

%% Simulazione impianto costituito da:
% Condotto principale + risonatore di Helmholtz
% Utilizzo della modellazione riportata nel paragrafo \ref{par:modello_x_helmholtz}

clear all, close all, clc

%% Dati problema
T=273.15+20;           % [K] temperatura aria nel condotto
R=287;                % [J/(kg*K)]
gamma=1.4;            % Cp/Cv
c0=sqrt(gamma*R*T);  % [m/s] velocità del suono
p=10e3;               % [Pa] ampiezza pressione dinamica
p0=p*10;              % [Pa] pressione statica
rho0=p0/(R*T);        % [kg/m^3] densità aria nel condotto
C=2.2089e-004;        % [m^3] cilindrata compressore
nu=200e-7;            % [m^2/s] viscosità cinematica aria a 100°C e 1bar

d_cnd=10/1000;        % [m] diametro condotto
S_cnd=pi*d_cnd^2/4;   % [m^2] area sezione condotto
Ltot=20;              % [m] lunghezza totale condotto
Lr=3                  % [m] posizione inserimento risonatore
f_ris=20;             % [Hz] frequenza di progetto del risonatore (2Xgiro)
w_comp=2*pi*f_ris     % [rad/sec] pulsazione 1X

%% Condizioni al contorno
p_in = p; p_out= 0;

%% Scelta parametri di analisi
xx=[0:0.1:Ltot];      % [m] posizione di analisi
f_an=5:0.1:40;        % [Hz] frequenza di analisi
omega=2*pi*f_an;      % [rad/sec] pulsazione di analisi

r=2*rho0/d_cnd*...
    sqrt(2*nu*w_comp); % fattore di smorzamento nel condotto
g=r/(2*c0*rho0)+...
    j*omega/c0;        % smorzamento gamma

%% MODELLO CONDOTTO SENZA RISONATORE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni in ingresso e uscita
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L1=Ltot;
S1=S_cnd;              % [m^2] area sezione condotto

coeff1 = fcn_coeff_ABCD(rho0,c0,g,omega,L1,S1);

for ww=1:length(omega)
    mat1 = fcn_matrix_ABCD(coeff1, ww);

```

```

A_ = mat1(1,1);
B_ = mat1(1,2);
C_ = mat1(2,1);
D_ = mat1(2,2);

P_in(ww) = p_in;
P_out(ww) = p_out;
Q_in(ww) = A_/C_*P_in(ww)-(A_*D_/C_-B_)*P_out(ww);
Q_out(ww) = 1/C_*P_in(ww)-D_/C_*P_out(ww);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni lungo la coordinata x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:length(xx)
    L2=Ltot-xx(ii);
    S2=S_cnd;

    coeff2_x = fcn_coeff_ABCD(rho0,c0,g,omega,L2,S2);

    for ww=1:length(omega)
        mat2_x = fcn_matrix_ABCD(coeff2_x, ww);

        out_xx = mat2_x*[Q_out(ww);P_out(ww)];
        Qx(ww,ii) = out_xx(1);
        Px(ww,ii) = out_xx(2);
    end
end

nome_fig=sprintf('Pressioni nel condotto senza risonatore');
figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,abs(Px),'EdgeColor','none'), grid on, colorbar
colormap Jet title('Pressioni nel condotto senza risonatore
$[Pa]$', 'Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex') zlabel('$P_{x1}, P_{x2}$
$[Pa]$', 'Interpreter','Latex') caxis([0 5e4])

nome_fig=sprintf('Portate nel condotto senza risonatore');
figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,abs(Qx),'EdgeColor','none'), grid on, colorbar
colormap Jet title('Portate nel condotto senza risonatore
$[m^3/s]$', 'Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex') zlabel('$Q_{x1}, Q_{x2}$
$[m^3/s]$', 'Interpreter','Latex') caxis([0 3e-3])

```

```

%% MODELLO CONDOTTO CON RISONATORE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Grandezze caratteristiche risonatore
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D_col=d_cnd;           % [m] diametro collo risonatore
L_col=0.8;             % [m] lunghezza collo risonatore
D_vol=50/1000;        % [m] diametro volume risonatore

S_col=pi*D_col^2/4;   % [m^2] area sezione collo risonatore
S_vol=pi*D_vol^2/4;   % [m^2] area sezione camera risonatore
V_vol=S_col/L_col*c0^2/...
    (2*pi*f_ris)^2;    % [m^3] volume camera risonatore
L_vol=V_vol/S_vol     % [m] lunghezza camera risonatore

% Fattore smorzamento nel risonatore
r_col=2*rho0/D_col*...
    sqrt(2*nu*w_comp); % fattore di smorzamento nel collo
g_col=r_col/(2*c0*rho0)+...
    j*omega/c0;        % smorzamento gamma nel collo
r_vol=2*rho0/D_vol*...
    sqrt(2*nu*w_comp); % fattore di smorzamento nella camera
g_vol=r_vol/(2*c0*rho0)+...
    j*omega/c0;        % smorzamento gamma nella camera

LR1=Lr; LR3=Ltot-Lr;
LR5=L_col;           % [m] lunghezza collo risonatore
LR6=L_vol;           % [m] lunghezza camera risonatore

S1=S_cnd;           % [m^2] area sezione
S2=S_cnd;           % [m^2] area sezione
S3=S_cnd;           % [m^2] area sezione
S4=S_cnd;           % [m^2] area sezione
S5=S_col;           % [m^2] area sezione collo risonatore
S6=S_vol;           % [m^2] area sezione camera risonatore

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni in ingresso e uscita
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
coeff1_R = fcn_coeff_ABCD(rho0,c0,g,omega,LR1,S1); coeff3_R =
fcn_coeff_ABCD(rho0,c0,g,omega,LR3,S3); coeff5_R =
fcn_coeff_ABCD(rho0,c0,g_col,omega,LR5,S5); coeff6_R =
fcn_coeff_ABCD(rho0,c0,g_vol,omega,LR6,S6);

Z500 = (coeff5_R.C .* coeff6_R.B + coeff5_R.D .* coeff6_R.D)/...
    (coeff5_R.A .* coeff6_R.B + coeff5_R.B .* coeff6_R.D);

for ww=1:length(omega)
    mat1_R = fcn_matrix_ABCD(coeff1_R, ww);
    mat3_R = fcn_matrix_ABCD(coeff3_R, ww);

```

```

matZ500 = [1 1/Z500(ww)
           0 1          ];

totMatrix = mat1_R*matZ500*mat3_R;

A_ = totMatrix(1,1);
B_ = totMatrix(1,2);
C_ = totMatrix(2,1);
D_ = totMatrix(2,2);

P_in_R(ww) = p_in;
P_out_R(ww) = p_out;
Q_in_R(ww) = A_/C_*P_in_R(ww)-(A_*D_/C_-B_)*P_out_R(ww);
Q_out_R(ww) = 1/C_*P_in_R(ww)-D_/C_*P_out_R(ww);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni lungo la coordinata x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:length(xx)
    if xx(ii)<=Lr
        LR2=Lr-xx(ii);
        coeff2_Rx = fcn_coeff_ABCD(rho0,c0,g,omega,LR2,S2);

        for ww=1:length(omega)
            mat2_Rx = fcn_matrix_ABCD(coeff2_Rx, ww);
            mat3_R = fcn_matrix_ABCD(coeff3_R, ww);
            matZ500 = [1 1/Z500(ww)
                       0 1          ];

            out_xx = mat2_Rx*matZ500*mat3_R*[Q_out_R(ww);P_out_R(ww)];
            Qx_R(ww,ii) = out_xx(1);
            Px_R(ww,ii) = out_xx(2);
        end
    else
        LR4=Ltot-xx(ii);
        coeff4_Rx = fcn_coeff_ABCD(rho0,c0,g,omega,LR4,S4);

        for ww=1:length(omega)
            mat4_Rx = fcn_matrix_ABCD(coeff4_Rx,ww);

            out_xx = mat4_Rx*[Q_out_R(ww);P_out_R(ww)];
            Qx_R(ww,ii) = out_xx(1);
            Px_R(ww,ii) = out_xx(2);
        end
    end
end
end
end

```

```

nome_fig=sprintf('Pressioni nel condotto. Risonatore in x=%s m',num2str(Lr));
figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,abs(Px_R),'EdgeColor','none'), grid on, colorbar
colormap Jet title('Pressioni nel condotto con risonatore
$[Pa]$', 'Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex') zlabel('$P_{x1}, P_{x2}$
[Pa]$', 'Interpreter','Latex') caxis([0 5e4])

nome_fig=sprintf('Portate nel condotto. Risonatore in x=%s m',num2str(Lr));
figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,abs(Qx_R),'EdgeColor','none'), grid on, colorbar
colormap Jet title('Portate nel condotto con risonatore
$[m^3/s]$', 'Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex') zlabel('$Q_{x1}, Q_{x2}$
[m^3/s]$', 'Interpreter','Latex') caxis([0 3e-3])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni lungo la coordinata y
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
yy=[0.1:0.01:LR5+LR6-0.1]; % [m] posizione di analisi

for ww=1:length(omega)

    mat1_R = fcn_matrix_ABCD(coeff1_R, ww);
    mat5_R = fcn_matrix_ABCD(coeff5_R, ww);
    mat6_R = fcn_matrix_ABCD(coeff6_R, ww);
    matZ500 = [1 1/Z500(ww)
              0 1          ];

    out_Lr = inv(mat1_R)*[Q_in(ww);P_in(ww)];
    QLr_R(ww) = out_Lr(1);
    PLr_R(ww) = out_Lr(2);

    out_03 = inv(mat1_R*matZ500)*[Q_in(ww);P_in(ww)];
    Q03_R(ww) = out_03(1);
    P03_R(ww) = out_03(2);

    % Condizioni in ingresso al risonatore
    Q05_R(ww) = QLr_R(ww)-Q03_R(ww);
    P05_R(ww) = P03_R(ww);

    % Condizioni alla fine del volume del risonatore
    out_L6 = inv(mat5_R*mat6_R)*[Q05_R(ww);P05_R(ww)];
    QL6_R(ww) = out_L6(1);
    QL6_R(ww) = 0; % 10^-17: forzata a zero
    PL6_R(ww) = out_L6(2);

```

```

% Condizioni in ingresso al camera del risonatore
out_06 = mat6_R*[QL6_R(ww);PL6_R(ww)];
Q06_R(ww) = out_06(1);
P06_R(ww) = out_06(2);
end

for ii=1:length(yy)
    if yy(ii)<=LR5
        LR5y=LR5-yy(ii);
        coeff5_Ry = fcn_coeff_ABCD(rho0,c0,g_col,omega,LR5y,S5);

        for ww=1:length(omega)
            mat5_Ry = fcn_matrix_ABCD(coeff5_Ry, ww);
            mat6_R = fcn_matrix_ABCD(coeff6_R, ww);

            out_yy = mat5_Ry*mat6_R*[QL6_R(ww);PL6_R(ww)];
            Qy_R(ww,ii) = out_yy(1);
            Py_R(ww,ii) = out_yy(2);
        end
    else
        LR6y=LR5+LR6-yy(ii);
        coeff6_Ry = fcn_coeff_ABCD(rho0,c0,g_vol,omega,LR6y,S6);

        for ww=1:length(omega)
            mat6_Ry = fcn_matrix_ABCD(coeff6_Ry, ww);

            out_yy = mat6_Ry*[QL6_R(ww);PL6_R(ww)];
            Qy_R(ww,ii) = out_yy(1);
            Py_R(ww,ii) = out_yy(2);
        end
    end
end

end

%% Plot rapporto pressione con e senza risonatore
pos_analisi = 100; ABB =
abs(Px_R(:,pos_analisi))./abs(Px(:,pos_analisi));
nome_fig=sprintf('Rapporto pressioni con e senza risonatore');
figure('Name',nome_fig,'NumberTitle','off'), plot(f_an, ABB), grid
on title('Rapporto pressioni con e senza
risonatore','Interpreter','Latex') xlabel('$f_R$
$[Hz]$', 'Interpreter','Latex')
ylabel('$\frac{P_{cH}}{P_{sC}}$', 'Interpreter','Latex')

%%
ABB = abs(Px_R)./abs(Px); index = find(ABB > 1); ABB(index) = nan;

nome_fig=sprintf('Rapporto pressioni con e senza risonatore
Px_R/Px'); figure('Name',nome_fig,'NumberTitle','off')

```

```

contourf(xx,f_an,ABB,'EdgeColor','none'), grid on, colorbar
% camlight right
colormap Jet title('Rapporto pressioni con e senza
risonatore','Interpreter','Latex') xlabel('$x_1, x_2$
[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex')
zlabel('$P_{xR}/P_{x}$', 'Interpreter','Latex') caxis([0 1])

```

### Funzione “fcn\_coeff\_ABCD”

Funzione per la determinazione delle costanti  $A$ ,  $B$ ,  $C$  e  $D$  del modello presentato nel capitolo 4.2.

```

function coeff = fcn_coeff_ABCD(rho0,c0,gamma,omega,L,S)

coeff.A = cosh(gamma*L); coeff.B =
(j*omega*S)/(rho0*c0^2*gamma)*(sinh(gamma*L)); coeff.C =
(rho0*c0^2*gamma)/(j*omega*S)*(sinh(gamma*L)); coeff.D =
coeff.A;

```

### Funzione “fcn\_matrix\_ABCD”

Funzione che determina la matrice delle costanti  $A$ ,  $B$ ,  $C$  e  $D$  in funzione della pulsazione di analisi.

```

function matrix = fcn_matrix_ABCD(coeff,index)

matrix = [coeff.A(index) coeff.B(index)
          coeff.C(index) coeff.D(index)];

```

## E.3 Script MatLab modello condotto e risonatore side-branch

Script di MatLab per la determinazione dell'andamento delle pressioni e delle portate dinamiche lungo il condotto principale in assenza e in presenza di un risonatore side-branch.

```

%% Simulazione impianto costituito da:
% Condotto principale + risonatore side-branch
% Utilizzo della modellazione riportata nel paragrafo \ref{par:modello_x_sidebranch}

clear all, close all, clc

%% Dati problema
T=273.15+20;           % [K] temperatura aria nel condotto
R=287;                % [J/(kg*K)]

```

```

gamma=1.4;           % Cp/Cv
c0=sqrt(gamma*R*T); % [m/s] velocità del suono
p=10e3;             % [Pa] ampiezza pressione dinamica
p0=p*10;           % [Pa] pressione statica
rho0=p0/(R*T);     % [kg/m^3] densità aria nel condotto
C=2.2089e-004;     % [m^3] cilindrata compressore
nu=200e-7;         % [m^2/s] viscosità cinematica aria a 100°C e 1bar

d_cnd=10/1000;     % [m] diametro condotto
S_cnd=pi*d_cnd^2/4; % [m^2] area sezione condotto
Ltot=20;           % [m] lunghezza totale condotto
Lr=3;              % [m] posizione inserimento risonatore
f_ris=20;          % [Hz] frequenza di progetto del risonatore (2Xgiro)
w_comp=2*pi*f_ris  % [rad/sec] pulsazione 1X

%% Condizioni al contorno
p_in = p; p_out= 0;

%% Scelta dati di analisi
xx=[0:0.1:Ltot];   % [m] posizione di analisi
f_an=5:0.1:40;     % [Hz] frequenza di analisi
omega=2*pi*f_an;   % [rad/sec] pulsazione di analisi

r=2*rho0/d_cnd*...
    sqrt(2*nu*w_comp); % fattore di smorzamento nel condotto
g=r/(2*c0*rho0)+...
    j*omega/c0;       % smorzamento gamma

%% MODELLO CONDOTTO SENZA RISONATORE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni in ingresso e uscita
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L1=Ltot;
S1=S_cnd;           % [m^2] area sezione condotto

coeff1 = fcn_coeff_ABCD(rho0,c0,g,omega,L1,S1);

for ww=1:length(omega)
    mat1 = fcn_matrix_ABCD(coeff1, ww);

    A_ = mat1(1,1);
    B_ = mat1(1,2);
    C_ = mat1(2,1);
    D_ = mat1(2,2);

    P_in(ww) = p_in;
    P_out(ww) = p_out;
    Q_in(ww) = A_/C_*P_in(ww)-(A_*D_/C_-B_)*P_out(ww);
    Q_out(ww) = 1/C_*P_in(ww)-D_/C_*P_out(ww);

```



```

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni lungo la coordinata x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:length(xx)
    L2=Ltot-xx(ii);
    S2=S_cnd;

    coeff2_x = fcn_coeff_ABCD(rho0,c0,g,omega,L2,S2);

    for ww=1:length(omega)
        mat2_x = fcn_matrix_ABCD(coeff2_x, ww);

        out_xx = mat2_x*[Q_out(ww);P_out(ww)];
        Qx(ww,ii) = out_xx(1);
        Px(ww,ii) = out_xx(2);
    end
end

nome_fig=sprintf('Pressioni nel condotto senza risonatore');
figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,abs(Px),'EdgeColor','none'), grid on, colorbar
colormap Jet title('Pressioni nel condotto senza risonatore
$[Pa]$', 'Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex') zlabel('$P_{x1}, P_{x2}
[Pa]$', 'Interpreter','Latex') caxis([0 5e4])

nome_fig=sprintf('Portate nel condotto senza risonatore');
figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,abs(Qx),'EdgeColor','none'), grid on, colorbar
colormap Jet title('Portate nel condotto senza risonatore
$[m^3/s]$', 'Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex') zlabel('$Q_{x1}, Q_{x2}
[m^3/s]$', 'Interpreter','Latex') caxis([0 3e-3])

%% MODELLO CONDOTTO CON RISONATORE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Grandezze caratteristiche risonatore
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D_RIS = d_cnd;           % [m] diametro risonatore
L_RIS = c0/(4*f_ris);    % [m] lunghezza collo risonatore

S_RIS = pi*D_RIS^2/4; %S_RIS = S_cnd

% Fattore smorzamento nel risonatore
r_side=2*rho0/D_RIS*...

```

```

    sqrt(2*nu*w_comp);      % fattore di smorzamento nel side-branch
g_side=r_side/(2*c0*rho0)+...
    j*omega/c0;           % smorzamento gamma nel side-branch

LR1=Lr; LR3=Ltot-Lr;
LR5=L_RIS;                % [m] lunghezza collo risonatore

S1=S_cnd;                 % [m^2] area sezione
S2=S_cnd;                 % [m^2] area sezione
S3=S_cnd;                 % [m^2] area sezione
S4=S_cnd;                 % [m^2] area sezione
S5=S_RIS;                 % [m^2] area sezione collo risonatore

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni in ingresso e uscita
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
coeff1_R = fcn_coeff_ABCD(rho0,c0,g,omega,LR1,S1); coeff3_R =
fcn_coeff_ABCD(rho0,c0,g,omega,LR3,S3); coeff5_R =
fcn_coeff_ABCD(rho0,c0,g_side,omega,LR5,S5);

Z500 = coeff5_R.D ./ coeff5_R.B;

for ww=1:length(omega)
    mat1_R = fcn_matrix_ABCD(coeff1_R, ww);
    mat3_R = fcn_matrix_ABCD(coeff3_R, ww);

    matZ500 = [1 1/Z500(ww)
               0 1          ];

    totMatrix = mat1_R*matZ500*mat3_R;

    A_ = totMatrix(1,1);
    B_ = totMatrix(1,2);
    C_ = totMatrix(2,1);
    D_ = totMatrix(2,2);

    P_in_R(ww) = p_in;
    P_out_R(ww) = p_out;
    Q_in_R(ww) = A_/C_*P_in_R(ww)-(A_*D_/C_-B_)*P_out_R(ww);
    Q_out_R(ww) = 1/C_*P_in_R(ww)-D_/C_*P_out_R(ww);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinazione portate e pressioni lungo la coordinata x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:length(xx)
    if xx(ii)<=Lr
        LR2=Lr-xx(ii);

```

```

coeff2_Rx = fcn_coeff_ABCD(rho0,c0,g,omega,LR2,S2);

for ww=1:length(omega)
    mat2_Rx = fcn_matrix_ABCD(coeff2_Rx, ww);
    mat3_R = fcn_matrix_ABCD(coeff3_R, ww);
    matZ500 = [1 1/Z500(ww)
               0 1          ];

    out_xx = mat2_Rx*matZ500*mat3_R*[Q_out_R(ww);P_out_R(ww)];
    Qx_R(ww,ii) = out_xx(1);
    Px_R(ww,ii) = out_xx(2);
end
else
LR4=Ltot-xx(ii);
coeff4_Rx = fcn_coeff_ABCD(rho0,c0,g,omega,LR4,S4);

for ww=1:length(omega)
    mat4_Rx = fcn_matrix_ABCD(coeff4_Rx,ww);

    out_xx = mat4_Rx*[Q_out_R(ww);P_out_R(ww)];
    Qx_R(ww,ii) = out_xx(1);
    Px_R(ww,ii) = out_xx(2);
end
end
end

nome_fig=sprintf('Pressioni nel condotto. Risonatore in x=%s m',num2str(Lr));
figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,abs(Px_R),'EdgeColor','none'), grid on, colorbar
colormap Jet title('Pressioni nel condotto con risonatore
$[Pa]$', 'Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex') zlabel('$P_{x1}, P_{x2}$
$[Pa]$', 'Interpreter','Latex') caxis([0 5e4])

nome_fig=sprintf('Portate nel condotto. Risonatore in x=%s m',num2str(Lr));
figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,abs(Qx_R),'EdgeColor','none'), grid on, colorbar
colormap Jet title('Portate nel condotto con risonatore
$[m^3/s]$', 'Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex') zlabel('$Q_{x1}, Q_{x2}$
$[m^3/s]$', 'Interpreter','Latex') caxis([0 3e-3])

%% Plot rapporto pressione con e senza risonatore
pos_analisi = 100; ABB =
abs(Px_R(:,pos_analisi))./abs(Px(:,pos_analisi));
nome_fig=sprintf('Rapporto pressioni con e senza risonatore');
figure('Name',nome_fig,'NumberTitle','off'), plot(f_an, ABB), grid

```

```

on title('Rapporto pressioni con e senza
risonatore','Interpreter','Latex') xlabel('$f_R$
$[Hz]$', 'Interpreter','Latex')
ylabel('$\frac{P_{cH}}{P_{sC}}$', 'Interpreter','Latex')

ABB = abs(Px_R)./abs(Px); index = find(ABB > 1); ABB(index) = nan;

nome_fig=sprintf('Rapporto pressioni con e senza risonatore
Px_R/Px'); figure('Name',nome_fig,'NumberTitle','off')
contourf(xx,f_an,ABB,'EdgeColor','none'), grid on, colorbar
% camlight right
colormap Jet title('Rapporto pressioni con e senza
risonatore','Interpreter','Latex') xlabel('$x_1, x_2$
$[m]$', 'Interpreter','Latex') ylabel('$f_R
[Hz]$', 'Interpreter','Latex')
zlabel('$P_{xR}/P_{x}$', 'Interpreter','Latex') caxis([0 1])

```

### Funzione “fcn\_coeff\_ABCD”

Funzione per la determinazione delle costanti  $A$ ,  $B$ ,  $C$  e  $D$  del modello presentato nel capitolo 4.2. Per il codice MatLab si faccia riferimento all'appendice E.2.

### Funzione “fcn\_matrix\_ABCD”

Funzione che determina la matrice delle costanti  $A$ ,  $B$ ,  $C$  e  $D$  in funzione della pulsazione di analisi. Per il codice MatLab si faccia riferimento all'appendice E.2.

## APPENDICE F

# Progetto del risonatore di Helmholtz

### F.1 Progettazione del risonatore di Helmholtz a geometria variabile

Vengono di seguito riportate alcune messe in tavola relative a parti costituenti il risonatore di Helmholtz a geometria variabile progettato e realizzato per la fase di valutazione delle performance di abbattimento presentate nel paragrafo 6.2. Nella figura F.1 si riporta il disegno di assieme della trasmissione realizzata per la movimentazione dello stantuffo del risonatore a geometria variabile, dove vengono messe in evidenza le dimensioni principali. La figura F.2 mette in evidenza gli elementi principali costituenti la trasmissione. Per la trasformazione del moto da rotatorio a lineare è necessario realizzare nel sistema di trasmissione un opportuno supporto per l'alloggiamento della chiocciola della vite a ricircolo e il collegamento dello stantuffo. Tale supporto, mostrato in figura F.3, è stato progettato al fine di garantire uno spostamento lineare con adeguate tolleranze di parallelismo e ortogonalità, nonché resistere alle forze trasmesse dallo stantuffo (par. 6.2.1).

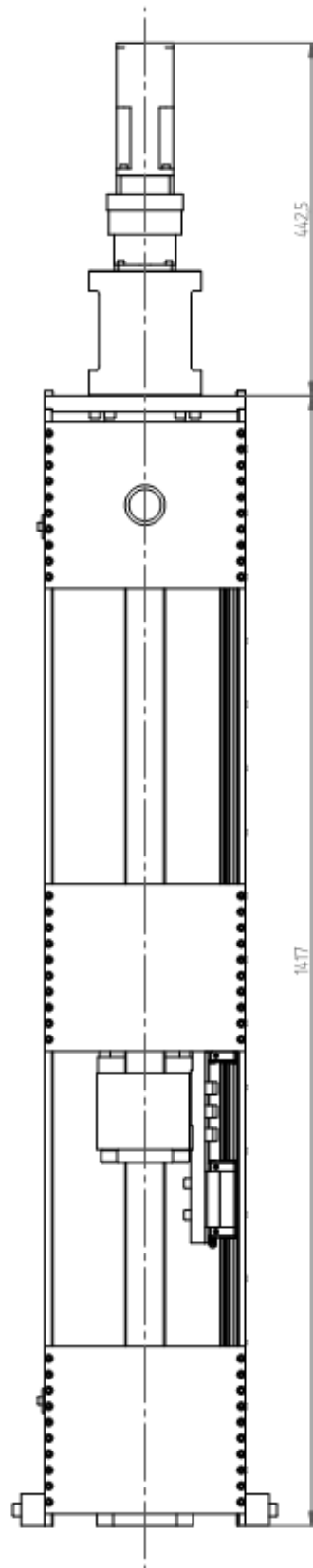


Figura F.1: *Assieme della trasmissione per la movimentazione dello stantuffo del risonatore di Helmholtz a geometria variabile.*

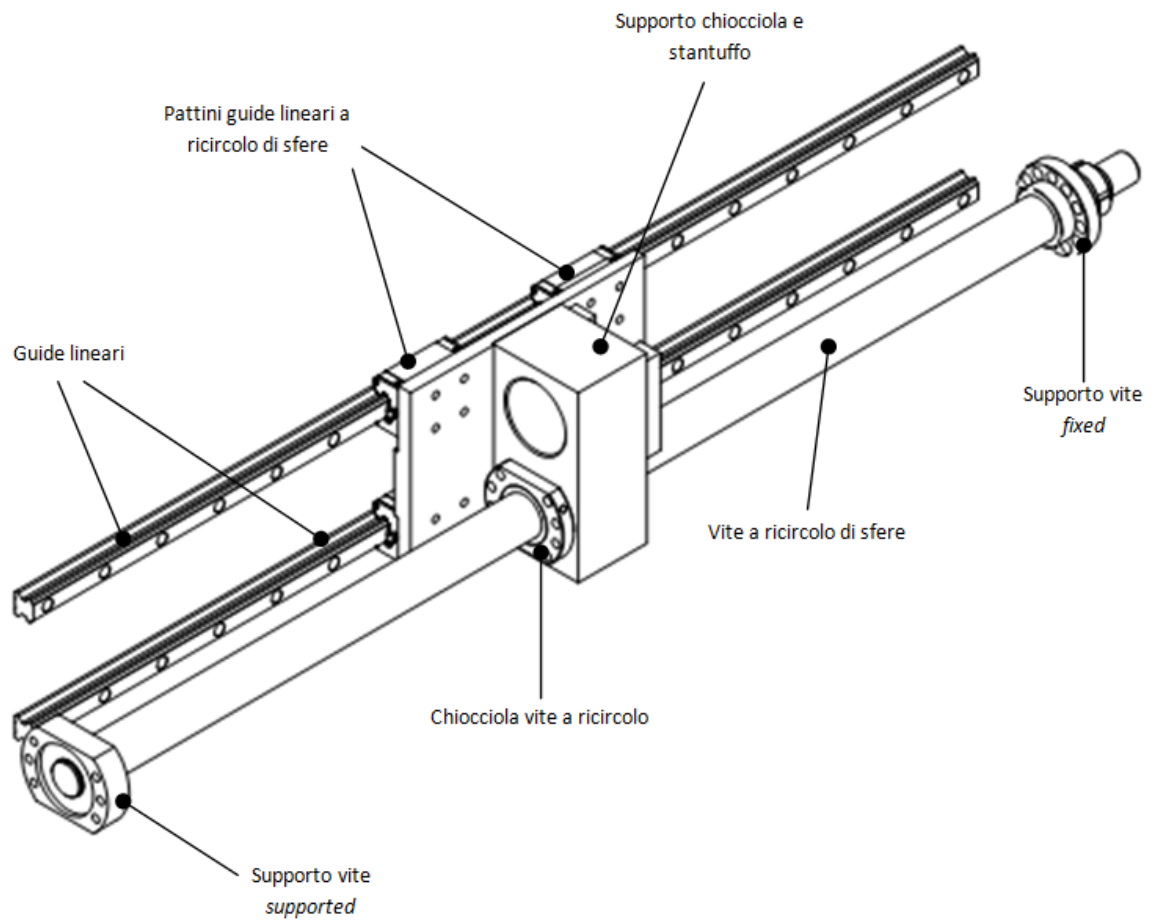


Figura F.2: *Elementi principali della trasmissione per la movimentazione dello stantuffo del risonatore di Helmholtz a geometria variabile.*

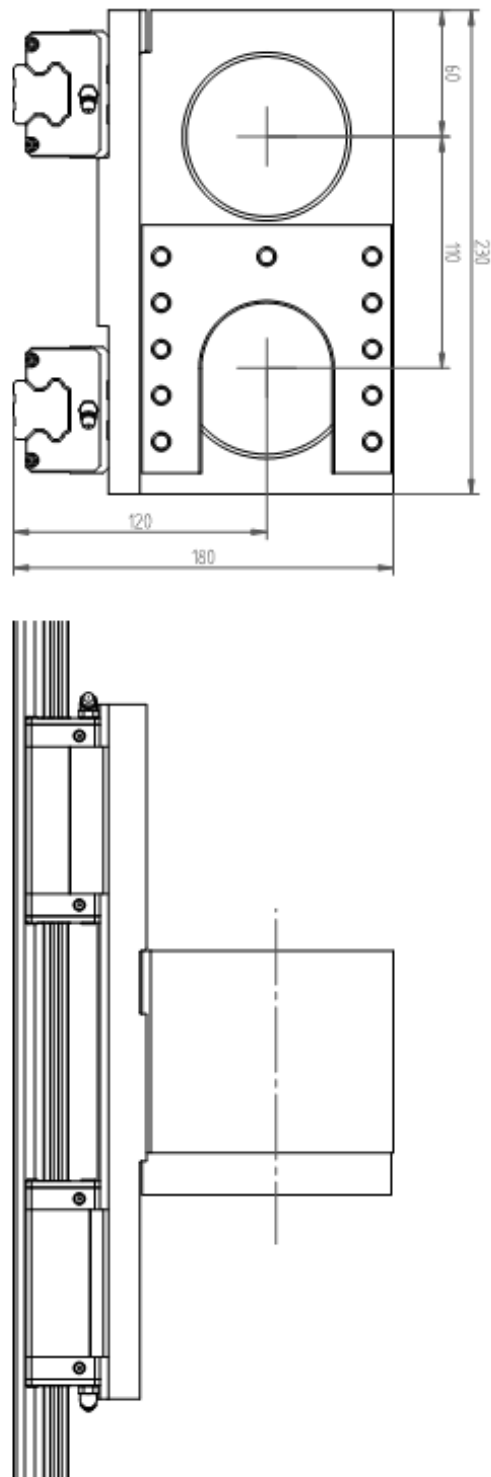


Figura F.3: Particolari della trasmissione per la movimentazione dello stantuffo del risonatore di Helmholtz a geometria variabile.



## APPENDICE G

# Codice di controllo per risonatore a geometria variabile

### G.1 Codice sorgente per il risonatore di Helmholtz a geometria variabile

Analogamente al controllo del rumore, il codice sorgente è stato realizzato con l'ausilio di due differenti editor, un primo editor è quello fornito da National Instruments e noto come *LabWindows CVI* mentre il secondo editor utilizzato è *Microsoft Visual Studio*. Con questi editor è stato realizzato il codice sorgente nel quale sono state implementate le principali funzioni per l'inizializzazione, per il ripristino in caso di terminazione improvvisa del sistema di controllo e per l'esecuzione dell'algoritmo di controllo.

#### G.1.1 Struttura del codice

Il codice è stato strutturato su quattro file con il codice sorgente e cinque header file. Vengono riportati i file dei codici sorgenti per il controllo del moto del risonatore a geometria variabile:

1. *cvidll.c*

File contenente la struttura principale dell'algoritmo di controllo dove vengono gestite le diverse funzioni del sistema di controllo, secondo una funzionalità a stati macchina (funzionalità del sistema di controllo).

2. *cvidll.h*

Header file contenente i prototipi di funzione del file *cvidll.c*.

3. *func\_lib.c*

File contenente le funzioni elementari necessarie all'esecuzione degli stati macchina.

4. *func\_lib.h*

Header file contenente i prototipi di funzione del file *func\_lib.c*.

5. *func\_moto.c*

File contenente le funzioni proprie del controllo del moto, nonché la definizione dei parametri principali del sistema meccanico ed elettrico, la definizione dei guadagni del regolatore PID (par. 6.3.2) e i parametri propri del risonatore di Helmholtz per la determinazione del setpoint di posizione.

6. *func\_moto.h*

Header file contenente i prototipi di funzione del file *func\_moto.c*.

7. *func\_statiMAC.c*

File contenente le funzioni dei singoli stati macchina del sistema di controllo. Queste funzioni vengono gestite dal file *cvidll.c*.

8. *func\_statiMAC.h*

Header file contenente i prototipi di funzione del file *func\_statiMAC.c*.

9. *dati\_header.h*

Header file contenente la definizione delle strutture dati e delle costanti utilizzate nel codice sorgente.

**File “cvidll.c”**

```

/*-----*/
/* Include files */
/*-----*/
#include <windows.h>
#include <analysis.h>
#include <cvirte.h>
#include <utility.h>

```

```

#include <NIDAQmx.h>
#include <ansi_c.h>
#include <math.h>

#include "myfilter.h"
#include "cvidll.h"
#include "dati_header.h"
#include "func_lib.h"
#include "func_moto.h"
#include "func_statiMAC.h"

/*-----*/
/* Definizione e inizializzazione variabili globali */
/*-----*/
TypePasso statoMAC = sM_InizializzaDati;

/*-----*/
/* DLL Main */
/*-----*/
int __stdcall DllMain (HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
switch (fdwReason)
{
case DLL_PROCESS_ATTACH:

// Respond to DLL loading by initializing the RTE
if (InitCVIRTE (hinstDLL, 0, 0) == 0)
return 0;

break;
case DLL_PROCESS_DETACH:

// Respond to DLL unloading by closing the RTE for its use
if (!CVIRTEHasBeenDetached ())
CloseCVIRTE ();

break;
}
return 1;
}

/*-----*/
/* Funzione principale - Controllo macchina */
/*-----*/
int ControlloRisonatore(int *inputINT, double *inputDBL, double *storeDBL,
double *panDBL, double *varPlot)
{
static TipoEsecuzioneFase statoFase;

```

```

// VARIABILI PER ESECUZIONE LOOP
double k_eff;
static int STOP_ceck=OFF;

// VARIABILI UTENTE
int STOP=inputINT[0];
double k=inputDBL[0];

// LETTURA DELLA CONFIGURAZIONE DEL SISTEMA
res += readDigitalChannel();
res += readAnalogChannel();
if (res != 0)
    statoMAC = sM.FaultSW;

// VERIFICA SE L'UTENTE HA PREMUTO STOP
if ((STOP == ON) && (STOP_ceck == OFF) && (statoMAC != sM.TwHalt)
    && (statoMAC != sM.Halt))
{
    statoMAC = sM.TwHalt;
    STOP_ceck = ON;
}

// CALCOLO DEL TEMPO T_TILDE
k_eff = calc_K(k);
t_tilde = calcTime2(k_eff, t_tilde);

// ESECUZIONE DEGLI STATI MACCHINA
switch(statoMAC)
{
    case sM_InizializzaDati:    // Inizializzazione dati
    {
        statoFase = FaseEnd;

        if( statoFase == FaseEnd )
            statoMAC = sM.OkAvvio;

        break;
    }

    case sM.OkAvvio:          // Wait comando utente Ok inizia
    {
        statoFase = func_OkAvvio(inputINT[1]);

        if( statoFase == FaseEnd )
            statoMAC = sM.WaitOkHw;

        break;
    }
}

```

```
case sM_WaitOkHw:           // Wait emergenza disattivata
{
    statoFase = func_WaitOkHw();

    if( statoFase == FaseEnd )
        statoMAC = sM_OkZero;

    break;
}

case sM_OkZero:            // Aspetta comando utente OK eseguo ZERO
{
    statoFase = func_OkZero();

    if( statoFase == FaseEnd )
        statoMAC = sM_initZero;

    break;
}

case sM_initZero:         // Inizializzazione stato di ZERO
{
    statoFase = FaseInit;
    statoMAC = sM_Zero;

    break;
}

case sM_Zero:             // Esecuzione stato di ZERO
{
    statoFase = func_RunZero(statoFase);

    if( statoFase == FaseEnd )
        statoMAC = sM_initPreReady;

    break;
}

case sM_initPreReady:     // Inizializzazione operazioni di pre_Ready
{
    statoFase = FaseInit;
    statoMAC = sM_PreReady;

    break;
}

case sM_PreReady:        // Esecuzione operazioni di pre_Ready
{
    statoFase = func_PreReady(statoFase);
```

```
    if( statoFase == FaseEnd )
        statoMAC = sM_initReady;

    break;
}

case sM_initReady:           // Inizializzazione stato di Ready
{
    statoFase = FaseInit;
    statoMAC = sM_Ready;

    break;
}

case sM_Ready:               // Esecuzione stato di Ready
{
    statoFase = func_Ready(statoFase , inputINT [1]);

    if( statoFase == FaseEnd )
    {
        int sceltaReady = inputINT [2];

        if (sceltaReady == 0)
            statoMAC = sM_initValutaParSyst;
        else if (sceltaReady == 1)
            statoMAC = sM_initStimaPar;
        else if (sceltaReady == 2)
            statoMAC = sM_initContmoto;
        else if (sceltaReady == 3)
            statoMAC = sM_initContVibr;
    }
    break;
}

case sM_initContmoto:       // Inizializzazione stato CONTROLLO DEL MOTO
{
    statoFase = FaseInit;
    statoMAC = sM_ContMoto;

    break;
}

case sM_ContMoto:           // Esecuzione stato CONTROLLO DEL MOTO
{
    statoFase = func_ContMoto(statoFase , inputINT [1] , inputINT [2] ,
        inputINT [3] , inputINT [4] , inputDBL [1]);

    if( statoFase == FaseEnd )
```

```
        statoMAC = sM_initPreReady;

        break;
    }

    case sM_initFaultHW:          // Inizializzazione stato di FAULT hardware
    {
        statoFase = FaseInit;
        statoMAC = sM_FaultHW;

        break;
    }

    case sM_FaultHW:             // Esecuzione stato di FAULT hardware
    {
        statoFase = func_FaultHW(statoFase, inputINT[1]);

        if( statoFase == FaseEnd )
            statoMAC = sM_initPreReady;

        break;
    }

    case sM_initFaultHW_FC:      // Inizializzazione stato di FAULT finecorsa
    {
        statoFase = FaseInit;
        statoMAC = sM_FaultHW_FC;

        break;
    }

    case sM_FaultHW_FC:          // Esecuzione stato di FAULT finecorsa
    {
        statoFase = func_FaultHW_FC(statoFase, inputINT[1]);

        if( statoFase == FaseEnd )
            statoMAC = sM_initZero;

        break;
    }

    case sM_FaultSW:             // Esecuzione stato di FAULT software
    {
        statoFase = func_FaultSW();

        // Necessario riavviare il sistema di controllo PXI.
        errorSW = -1;

        break;
    }
}
```

```

    }

    case sM_TwHalt:           // Condizioni sicure di OFF
    {
        statoFase = func_TwHalt ();

        if( statoFase == FaseEnd )
            statoMAC = sM_Halt;

        break;
    }

    case sM_Halt:           // Arresto sicuro macchina
    {
        statoFase = func_Halt ();

        break;
    }
} // end switch(statoMAC)

// CONTROLLO PRESENZA FAULTS
// FAULT 1: Controllo presenza FAULTS emergenza
if ((errorHW == OFF) // && (errorHW_FC == OFF))
    errorHW = check_error(tipoFault);
if ((errorHW != OFF) && (flag_FAULT == OFF))
{
    statoMAC = sM_initFaultHW;
    flag_FAULT = ON;
}
// FAULT 2: Controllo presenza FAULTS fincorsa elettrici e limitazioni sw:
if (errorHW_FC == OFF)
    errorHW_FC = check_errorFC(tipoFault);
if ((errorHW_FC != OFF) && (flag_FAULT_FC == OFF))
{
    statoMAC = sM_TwHalt;
    flag_FAULT_FC = ON;
}

// REGOLATORE: CALCOLO DELL'USCITA ANALOGICA
RegolaAsse(&R);

// SCRITTURA DELLE USCITE DIGITALI E ANALOGICHE
if (task_puliti == OFF)
{
    output_digital ();
    output_analog ();
}

```



```
// VARIABILI DA VISUALIZZARE SU PANNELLO LABVIEW
panDBL[4] = passoStatoMAC; // Passo interno allo stato macchina
panDBL[5] = errorHW; // Indicatore errore hardware emergenza
panDBL[6] = errorHW_FC; // Indicatore errore hardware finecorsa
panDBL[7] = errorSW; // Indicatore errore software
panDBL[8] = waitUtenteSPctrl; // Flag: attesa input comando utente
panDBL[9] = statoNow.posLin0; // Posizione effettiva risonatore
panDBL[10] = statoFisicoNow.pressP[P01]; // Pressione punto P01 [Pa]
panDBL[11] = statoFisicoNow.pressP[P02]; // Pressione punto P02 [Pa]
panDBL[12] = statoFisicoNow.pressP[P03]; // Pressione punto P03 [Pa]
panDBL[13] = statoFisicoNow.pressP[P04]; // Pressione punto P04 [Pa]
panDBL[14] = statoFisicoNow.tempT[T01];
panDBL[15] = statoFisicoNow.tempT[T02];

// VARIABILI DA SALVARE SU FILE
storeDBL[0] = tickCount;
storeDBL[1] = statoMAC;
storeDBL[2] = passoStatoMAC;
storeDBL[3] = R.SP_Asse.posizione;
storeDBL[4] = R.SP_Asse.velocita;
storeDBL[5] = R.SP_Asse.accelerazione;
storeDBL[6] = R.stato_Asse->posLin0;
storeDBL[7] = R.stato_Asse->velLin;
storeDBL[8] = R.I_comando;
storeDBL[9] = statoFisicoNow.pressP[P01];
storeDBL[10] = statoFisicoNow.pressP[P02];
storeDBL[11] = statoFisicoNow.pressP[P03];
storeDBL[12] = statoFisicoNow.pressP[P04];
storeDBL[13] = statoFisicoNow.tempT[T01];
storeDBL[14] = statoFisicoNow.tempT[T02];
storeDBL[15] = statoNow.omega; // omega motore risonatore
storeDBL[16] = statoCompNow.omega; // omega motore compressore
storeDBL[17] = digStatus.Em_IN;
storeDBL[18] = digStatus.FC_Sx;
storeDBL[19] = digStatus.FC_Dx;

// VARIABILI PER I DIAGRAMMI IN LABVIEW
varPlot[0] = R.REG_pos.comp_FB;
varPlot[1] = R.REG_pos.comp_FF_Attriti;
varPlot[2] = R.REG_pos.comp_FF_Dyn;
varPlot[3] = R.I_comando;
varPlot[4] = statoFisicoNow.pressP[P01];
varPlot[5] = statoFisicoNow.pressP[P02];
varPlot[6] = statoFisicoNow.pressP[P03];
varPlot[7] = statoFisicoNow.pressP[P04];
varPlot[8] = statoFisicoNow.tempT[T01];
varPlot[9] = statoFisicoNow.tempT[T02];
```

```
// DETERMINAZIONE DEL TICK ASSOLUTO DI ESECUZIONE DEL LOOP
tickCount++;

return statoMAC;
}
```

### File “cvidll.h”

```
/*-----*/
/* Prototipi di funzione */
/*-----*/
int Creazione_task (double tempo_ciclo);
int ControlloRisonatore(int *inputINT, double *inputDBL,
                        double *storeDBL, double *panDBL, double *varPlot);
int Pulizia_task (void);
```

### File “func\_lib.c”

```
/*-----*/
/* Include files */
/*-----*/
#include <NIDAQmx.h>
#include <analysis.h>
#include <ansi_c.h>
#include "dati_header.h"
#include "func_statiMAC.h"
#include "myfilter.h"

/*-----*/
/* Definizione variabili e costanti interne */
/*-----*/
TaskHandle analog_input_task_handle;
TaskHandle analog_output_task_handle;
TaskHandle encoder_task_handle[3];
TaskHandle digital_input_task_handle;
TaskHandle digital_output_task_handle;

struct filtroIord filtra_velLin;
struct filtroIord filtra_RPMcomp;
struct filtroContinua calcMedia_sigEncComp;
```

```

/*-----*/
/* FUNZIONE CONTROLLA ERRORI RISONATORE - check_error() */
/* */
/* HELP: */
/* 1. Questa funzione controlla gli errori che non presuppongono */
/* necessariamente un riavvio della macchina: emergenza premuta. */
/*-----*/
int check_error(struct FaultMacchina qualeFault)
{
    int tipo_error=0;

    // CONTROLLO STATO DI EMERGENZA
    if ((qualeFault.checkEMERG_error == ON) && (tipo_error == 0))
    {
        // Condizione di emergenza premuta
        if (digStatus.Em_IN == 0)    tipo_error = (int)EMERG_error;
    }
    return tipo_error;
}

/*-----*/
/* FUNZIONE CONTROLLA ERRORI RISONATORE - check_errorFC() */
/* */
/* HELP: */
/* 1. Questa funzione controlla gli errori che necessitano un riavvio del */
/* sistema: raggiungimento di limiti software o dei fincorsa. */
/*-----*/
int check_errorFC(struct FaultMacchina qualeFault)
{
    int tipo_error=0;

    // CONTROLLO FINECORSO ELETTRICI
    if ((qualeFault.checkFC_error == ON) && (tipo_error == 0))
    {
        if ((digStatus.FC_Sx == ON) || (digStatus.FC_Dx == ON))
            tipo_error = (int)FC_error;
    }

    // CONTROLLO LIMITI SOFTWARE DI VELOCITA' MASSIMA
    if ((qualeFault.checkVELMAX_error == ON) && (tipo_error == 0))
    {
        // Condizione di velocità massima su un asse
        if ((fabs(statoNow.velLin_filt) >= Asse.vel_max) && (tipo_error == 0))
            tipo_error = (int)VELMAX_error;
    }

    // CONTROLLO LIMITI SOFTWARE DI POSIZIONE
    if ((qualeFault.checkPOSMAX_error == ON) && (tipo_error == 0))
    {

```

```

    // Condizione di superamento limiti corsa software
    if ((statoNow.posLin0 <= Asse.limPosMin) ||
        (statoNow.posLin0 >= Asse.limPosMax))
    {
        if (tipo_error == 0)
            tipo_error = (int)POSMAX_error;
    }
}
return tipo_error;
}

/*-----*/
/* Funzione - Funzione filtro continua */
/*-----*/
void iniz_filtroContinua(struct filtroContinua* data)
{
    memset(data,0x0,sizeof(struct filtroContinua));
}

void setup_filtroContinuaPunti(struct filtroContinua* data, int nPunti)
{
    int pt = nPunti;
    if( pt < 0 )
        pt = 0;
    else if(pt > MAX_PT_FILTRO_CONTINUA)
        pt = MAX_PT_FILTRO_CONTINUA;

    data->idx = 0;
    data->npt = pt;
}

double filtraCalcMedia(struct filtroContinua* data, double in)
{
    int k;
    double media = 0.0;

    if( data->idx >= data->npt )
        data->idx = 0;

    data->buffer[data->idx++] = in;
    for(k=0;k<data->npt;k++)
        media += data->buffer[k];

    media /= ((double)data->npt);

    return (media);
}

```

```

/*-----*/
/* Funzione - Funzione filtro passa basso II ordine */
/*-----*/
void iniz_SecondoOrdine(struct filtroIIord *data)
{
    memset(data,0x0,sizeof(struct filtroIIord));
}

void setup_SecondoOrdine(struct filtroIIord *data, double freq, double csi)
{
    data->k1 = 4.0*csi*PI*freq;
    data->k2 = PI*freq/csi;
    data->alpha = 4.0+2.0*Tc*(data->k1)+data->k1*data->k2*pow(Tc,2);
    data->beta = 2.0*data->k1*data->k2*pow(Tc,2) - 8.0;
    data->gamma = 4.0 - 2.0*data->k1*Tc+data->k1*data->k2*pow(Tc,2);
    data->K = (data->k1*data->k2*pow(Tc,2))/data->alpha;
    data->a1 = data->beta/data->alpha;
    data->a0 = data->gamma/data->alpha;
}

double filtraIIOrdine (struct filtroIIord *data, double in)
{
    data->ingresso [2] = data->ingresso [1];
    data->ingresso [1] = data->ingresso [0];
    data->ingresso [0] = in;

    data->uscita [2] = data->uscita [1];
    data->uscita [1] = data->uscita [0];
    data->uscita [0] = -(data->a1)*(data->uscita [1]) - (data->a0)*(data->uscita [2]) +
        (data->K)*(data->ingresso [0]+2.0*(data->ingresso [1])+data->ingresso [2]);

    return data->uscita [0];
}

/*-----*/
/* Funzione - Funzione inizializzazione filtri */
/*-----*/
void initFilters ()
{
    iniz_SecondoOrdine(&filtra_velLin);
    setup_SecondoOrdine(&filtra_velLin, 1000.0, 0.7);

    iniz_filtroContinua(&calcMedia_sigEncComp);
    setup_filtroContinuaPunti(&calcMedia_sigEncComp, 1000);
}

```

```

/*-----*/
/* Funzione - Creazione Task */
/*-----*/
int Creazione_task (double tempo_ciclo)
{
    func_InizializzaDati(tempo_ciclo);

    if (task_puliti == ON)
    {
        // ANALOG INPUTS
        // Acquisizione segnali di pressione e temperatura
        // (4 segnali di pressione, 2 segnali di temperatura).
        if (res == 0)
            res = DAQmxCreateTask (TASK_ANALOG_INPUT, &analog_input_task_handle);
        if (res == 0)
            res = DAQmxCreateAIVoltageChan (analog_input_task_handle,
                "PXI1Slot2/ai0:5", "", DAQmx_Val_Diff, min_Vin, max_Vin,
                DAQmx_Val_Volts, NULL);
        if (res == 0)
            res = DAQmxStartTask (analog_input_task_handle);

        // ANALOG OUTPUTS
        // Generazione del segnale di comando per il motore.
        if (res == 0)
            res = DAQmxCreateTask (TASK_ANALOG_OUTPUT, &analog_output_task_handle);
        if (res == 0)
            res = DAQmxCreateAOVoltageChan (analog_output_task_handle,
                "PXI1Slot4/ao0", "", min_Vout, max_Vout, DAQmx_Val_Volts, NULL);
        if (res == 0)
            res = DAQmxStartTask (analog_output_task_handle);

        // COUNTERS ENCODER
        // Acquisizione segnale encoder risonatore
        if (res == 0)
            res = DAQmxCreateTask (TASK_COUNTER_ANGULAR_ENC_RIS,
                &encoder_task_handle[ENC_RIS]);
        if (res == 0)
            res = DAQmxCreateCIAngEncoderChan (encoder_task_handle[0],
                "PXI_counters/ctr0", "", DAQmx_Val_X4, 0, 0, DAQmx_Val_AHighBHigh,
                DAQmx_Val_Radians, RISOLUZIONE_ENC_RIS, 0, "");
        if (res == 0)
            res = DAQmxStartTask (encoder_task_handle[0]);
        // Acquisizione segnale encoder albero compressore.
        if (res == 0)
            res = DAQmxCreateTask (TASK_COUNTER_ANGULAR_ENC_COMP,
                &encoder_task_handle[ENC_COMP]);
        if (res == 0)
            res = DAQmxCreateCIAngEncoderChan (encoder_task_handle[1],

```

```

        "PXI_counters/ctr1", "", DAQmx_Val_X4, 0, 0, DAQmx_Val_AHighBHigh,
        DAQmx_Val_Radians, RISOLUZIONE_ENC_COMP, 0, "");
    if (res == 0)
        res = DAQmxStartTask (encoder_task_handle [1]);

    // DIGITAL OUTPUTS
    // Generazione segnale per enable azionamento.
    // Generazione segnale digitale TTL sempre alto, per relè emergenza.
    // Generazione segnale alto-basso per verifica stabilità tempo ciclo macchina.
    // Totale: 3 canali di output digitali.
    if (res == 0)
        res = DAQmxCreateTask (TASK_DIGITAL_OUTPUT, &digital_output_task_handle);
    if (res == 0)
        res = DAQmxCreateDOChan (digital_output_task_handle,
            "PXI1Slot4/port0/line0:3", "", DAQmx_Val_ChanForAllLines);
    if (res == 0)
        res = DAQmxStartTask (digital_output_task_handle);

    // DIGITAL INPUTS
    // Acquisizione stato pulsante di emergenza.
    // Acquisizione stato dei due finecorsa elettrici.
    // Totale: 3 canali di input digitali.
    if (res == 0)
        res = DAQmxCreateTask (TASK_DIGITAL_INPUT, &digital_input_task_handle);
    if (res == 0)
        res = DAQmxCreateDIChan (digital_input_task_handle,
            "PXI1Slot4/port0/line4:6", "", DAQmx_Val_ChanForAllLines);
    if (res == 0)
        res = DAQmxStartTask (digital_input_task_handle);

    task_puliti = OFF;
}
return res;
}

/*-----*/
/* Funzione - Pulizia Task */
/*-----*/
int Pulizia_task (void)
{
    int i, res;

    if (task_puliti == OFF)
    {
        // ANALOG INPUTS
        if (analog_input_task_handle != -1)
        {
            DAQmxStopTask (analog_input_task_handle);

```

```
    res = DAQmxClearTask ( analog_input_task_handle );
    analog_input_task_handle = -1;
}
if ( res != 0 )
{
    res = 1; return res;
}

// ANALOG OUTPUTS
if ( analog_output_task_handle != -1)
{
    DAQmxStopTask ( analog_output_task_handle );
    res = DAQmxClearTask ( analog_output_task_handle );
    analog_output_task_handle = -1;
}
if ( res != 0 )
{
    res = 2; return res;
}

// COUNTERS ENCODER
//for ( i=0;i<=2;i++)
for ( i=0;i<2;i++)
{
    if ( encoder_task_handle [ i ] != -1)
    {
        DAQmxStopTask ( encoder_task_handle [ i ] );
        res = DAQmxClearTask ( encoder_task_handle [ i ] );
        encoder_task_handle [ i ] = -1;
    }
}
if ( res != 0 )
{
    res = 3; return res;
}

// DIGITAL INPUTS
if ( digital_input_task_handle != -1)
{
    DAQmxStopTask ( digital_input_task_handle );
    res = DAQmxClearTask ( digital_input_task_handle );
    digital_input_task_handle = -1;
}
if ( res != 0 )
{
    res = 4; return res;
}
```



```

// DIGITAL OUTPUTS
if (digital_output_task_handle != -1)
{
    DAQmxStopTask (digital_output_task_handle);
    res = DAQmxClearTask (digital_output_task_handle);
    digital_output_task_handle = -1;
}
if ( res != 0 )
{
    res = 5; return res;
}
task_puliti = ON;
}
return res;
}

/*-----*/
/* Funzione - write_digital */
/*-----*/
void write_dig (int mask, int stato)
{
    dig_out = (stato == ON) ? (dig_out | mask) : (dig_out & (~mask));
}

/*-----*/
/* Funzione - Lettura dei segnali digitali e degli encoder */
/*-----*/
int readDigitalChannel (void)
{
    int i, res=0;
    uInt32 dig_in=0; // Variabile temporanea ingressi digitali
    uInt8 st;

    if (task_puliti == OFF)
    {
        // LETTURA SEGNALI DIGITALI
        if (res == 0)
            res = DAQmxReadDigitalScalarU32 (digital_input_task_handle , 10.0,
                &dig_in , NULL);

        // LETTURA ENCODER
        if (res == 0)
            res = DAQmxReadCounterScalarF64 (encoder_task_handle [ENC_RIS] , 10.0 ,
                &statoNow.ENC, NULL);
        if (res == 0)
            res = DAQmxReadCounterScalarF64 (encoder_task_handle [ENC_COMP] , 10.0 ,
                &statoCompNow.ENC, NULL);
    }
}

```

```

// Elaborazione segnali digitali
for (i=4;i<7;i++)
{
    st = ((dig_in >> i) & 1) == 1;

    if( st == 0 )    // Stato FC e emergenza ATTIVI
    {
        stAlto[i] = 0;
        if( ++stBasso[i] > 10 )
        {
            statoDig[i] = 0;
            stBasso[i] = stAlto[i] = 0;
        }
    }
    else            // Stato FC e emergenza NON ATTIVI
    {
        stBasso[i] = 0;
        if( ++stAlto[i] > 10 )
        {
            statoDig[i] = 1;
            stBasso[i] = stAlto[i] = 0;
        }
    }
}

// Finecorsa attivo: segnale digitale ON
digStatus.Em.IN = statoDig[EMERG.IN]; // Stato pulsante emergenza
digStatus.FC.Sx = statoDig[FC.SX];    // Stato finecorsa lato motore
digStatus.FC.Dx = statoDig[FC.DX];    // Stato finecorsa lato condotto
}
return res;
}

/*-----*/
/* Funzione per la linearizzazione del segnale delle termocoppie */
/*-----*/
double volt2gradiCelsius(double Vin)
{
    double a= 0.00718;
    double b=-0.19520;
    double c=21.23630;
    double d= 0.02530;

    return a*pow(Vin,3) + b*pow(Vin,2) + c*Vin + d;
}

```

```

/*-----*/
/* Funzione - Lettura degli ingressi analogici del sistema */
/*-----*/
int readAnalogChannel (void)
{
    int i, res=0;
    int32 sampsPerChanRead;
    double AI[6];

    bassobasso = !bassobasso;
    write_dig (StartTrigger, bassobasso);

    if (task_puliti == OFF)
    {
        // LETTURA SEGNALI ANALOGICI
        if (res == 0)
            res = DAQmxReadAnalogF64 (analog_input_task_handle, 1, 10.0,
                DAQmx_Val_GroupByScanNumber, AI, 6, &sampsPerChanRead, NULL);
    }

    // LETTURA PARAMETRI COMPRESSORE
    statoCompOld.ENC = statoCompNow.ENC;
    statoCompOld.sigEnc = statoCompNow.sigEnc;
    statoCompOld.omega = statoCompNow.omega;

    statoCompNow.sigEnc = filtraCalcMedia(&calcMedia.sigEncComp, statoCompNow.ENC);

    // Velocità angolare [rad/sec]
    statoCompNow.omega = (statoCompNow.sigEnc - statoCompOld.sigEnc) / Tc;

    // LETTURA STATO ASSE
    statoOld.ENC = statoNow.ENC;
    statoOld.sigEncZERO = statoNow.sigEncZERO;
    statoOld.sigEnc = statoNow.sigEnc;
    statoOld.sigEnc0 = statoNow.sigEnc0;
    statoOld.posLin = statoNow.posLin;
    statoOld.posLin0 = statoNow.posLin0;
    statoOld.velLin = statoNow.velLin;
    statoOld.velLin_filt = statoNow.velLin_filt;
    statoOld.omega = statoNow.omega;

    // Segnale encoder senza azzeramento [rad]
    statoNow.sigEnc = statoNow.ENC;

    // Posizione lineare senza azzeramento [m]
    statoNow.posLin = statoNow.sigEnc * Asse.tau;

    // Velocità lineare [m/s]
    statoNow.velLin = (statoNow.posLin - statoOld.posLin) / Tc;

```

```

// Velocità lineare filtrata [m/s], utilizzata dalla funzione di verifica faults
statoNow.velLin_filt = filtraIIOrdine(&filtra_velLin , statoNow.velLin);

// Velocità angolare [rad/sec]
statoNow.omega = (statoNow.sigEnc - statoOld.sigEnc) / Tc;

// Segnale encoder con azzeramento [rad]
statoNow.sigEnc0 = statoNow.sigEnc - statoNow.sigEncZERO;

// Posizione lineare con azzeramento [m]
statoNow.posLin0 = statoNow.sigEnc0 * Asse.tau;

// LETTURA SEGNALI DI PRESSIONE (canali AI[0-3])
for (i=0;i<4;i++)
{
    statoFisicoOld.pressP[i] = statoFisicoNow.pressP[i];

    // Pressione P01, P02, P03, P04 [Pa]
    statoFisicoNow.pressP[i] = AI[i] / (SENS_P * 1.0E-6);
}

// LETTURA SEGNALI DI TEMPERATURA (canali AI[4-5])
statoFisicoOld.tempT[T01] = statoFisicoNow.tempT[T01];
statoFisicoOld.tempT[T02] = statoFisicoNow.tempT[T02];

// Temperatura termocoppia T01 [°C]
statoFisicoNow.tempT[T01] = volt2gradiCelsius(AI[4]) + T_AMBIENTE;

// Temperatura termocoppia T02 [°C]
statoFisicoNow.tempT[T02] = volt2gradiCelsius(AI[5]) + T_AMBIENTE;

return res;
}

/*-----*/
/* Funzione - Genera output digitali */
/*-----*/
int output_digital ()
{
    int res=0;

    // SCRITTURA SEGNALI DIGITALI
    res = DAQmxWriteDigitalScalarU32 (digital_output_task_handle , 1, 10.0,
        dig_out , NULL);
    if (res != 0)
    {
        res = 1; return res;
    }
}

```

```
    }
    return res;
}

/*-----*/
/* Funzione - Limitazione corrente azionamento */
/*-----*/
double MyLimitCurrent(Regolatore_data *reg, double val)
{
    double Imax = reg->dati_Asse->I_max_can;

    // Limitazione delle uscite
    if (val > Imax)
        val = Imax;
    else if (val < -Imax)
        val = -Imax;

    return val;
}

/*-----*/
/* Funzione - Genera output analogici */
/*-----*/
int output_analog ()
{
    int i;
    int res=0;
    int32 sampsPerChanWritten;
    double AO;

    // Limitazione dell'uscita
    R.I.comando = MyLimitCurrent(&R, R.I.comando);

    AO = R.I.comando/Asse.I_max*9.76;

    if (task_puliti == OFF)
    {
        // SCRITTURA SEGNALI ANALOGICI
        res = DAQmxWriteAnalogF64 (analog_output_task_handle, 1, 1, 10.0,
            DAQmx_Val_GroupByScanNumber, &AO, &sampsPerChanWritten, NULL);
        if (res != 0)
        {
            res = 1; return res;
        }
    }
    return res;
}
```

```

/*-----*/
/* Funzione di attesa */
/*-----*/
int func_wait(double tempo_attesa)
{
    int ok_attesa;
    static double time0=0.0;

    if (time0 <= tempo_attesa)
    {
        time0 += Tc;
        ok_attesa = OFF;
    }
    else
    {
        time0 = 0.0;
        ok_attesa = ON;
    }
    return ok_attesa;
}

/*-----*/
/* Funzione di calcolo posizione risonatore */
/*-----*/
double calcPosHelmholtz()
{
    double l_ris=0.0;
    double f_ris;

    f_ris = statoCompNow.omega/PI;
    l_ris = pow(C.SOUND/(2.0*PI),2) * pow(Helmholtz.d_collo_su_d,2) /
        (pow(f_ris,2) * Helmholtz.l_eq);

    debug[27]=pow(C.SOUND/(2.0*PI),2);
    debug[28]=pow(Helmholtz.d_collo_su_d,2);
    debug[29]=(pow(f_ris,2) * Helmholtz.l_eq);

    return (1.0 - l_ris + Helmholtz.l_correzione);
}

```

**File “func\_lib.h”**

```

/*-----*/
/* Prototipi di funzione */
/*-----*/
int check_error(struct FaultMacchina qualeFault);
int check_errorFC(struct FaultMacchina qualeFault);
int Creazione_task (double tempo_ciclo);
int Pulizia_task(void);

void write_dig(int mask, int stato);
int readDigitalChannel (void);
int readAnalogChannel (void);

int output_digital(void);
int output_analog(void);

void iniz_filtroContinua(struct filtroContinua* data);
void setup_filtroContinuaPunti(struct filtroContinua* data, int nPunti);
double filtraCalcMedia(struct filtroContinua* data, double in);

void iniz_SecondoOrdine(struct filtroIIord *data);
void setup_SecondoOrdine(struct filtroIIord *data, double freq, double csi);
double filtraIIOrdine (struct filtroIIord *data, double in);
void initFilters(void);

int func_wait(double tempo_attesa);
double MyLimitCurrent(Regolatore_data *reg, double val);

double calcPosHelmholtz(void);

extern TaskHandle analog_input_task_handle;
extern TaskHandle analog_output_task_handle;
extern TaskHandle encoder_task_handle [3];
extern TaskHandle digital_input_task_handle;
extern TaskHandle digital_output_task_handle;

```

**File “func\_moto.c”**

```

/*-----*/
/* Include files */
/*-----*/
#include <NIDAQmx.h>
#include <analysis.h>
#include <ansi_c.h>
#include "myfilter.h"
#include "dati_header.h"
#include "func_statiMAC.h"
#include "func_lib.h"

```

```

/*-----*/
/* Funzione per l'inizializzazione dei parametri degli assi */
/*-----*/
void initDatiAsse()
{
    // PARAMETRI ASSE SISTEMA
    Asse.passo = 0.010; // Passo vite ricircolo
    Asse.tauRid = 1.0/28.0; // Rapporto trasmissione riduttore
    Asse.Kt = 0.96; // Costante di coppia motore [Nm/Arms]
    Asse.I_max = 10.0; // Corrente massima azionamento [A]
    Asse.corsa = MAX_CORSA; // Corsa massima [m]
    Asse.limPosMin = 0.015; // Limite min. pos. ammiss. asse [m]
    Asse.limPosMax = MAX_CORSA - 0.015; // Limite max. pos. ammiss. asse [m]
    Asse.I_max_can = 2.0; // Max. corrente ammessa in uscita [A]
    Asse.vel_max = 0.1; // 0.03333; // Max. velocità ammessa in uscita [m/s]

    // Calcolo rapporto di trasmissione complessivo [m/rad]
    Asse.tau = Asse.tauRid * Asse.passo / (2.0 * PI);

    // Parametri attriti statici e viscosi
    Asse.v1 = 1.0e-3;
    Asse.v2 = 0.002;
    Asse.Fatt_st = 1.1687e3;
    Asse.Fatt_din_pos = 809.5892;
    Asse.Fatt_din_neg = 671.4929;
    Asse.m_pos = 2.2936e3;
    Asse.m_neg = -2.9767e3;
    Asse.coef_pos[0] = 7.2046e11;
    Asse.coef_pos[1] = -3.2409e9;
    Asse.coef_pos[2] = 4.3205e6;
    Asse.coef_pos[3] = -631.3312;
    Asse.coef_neg[0] = -9.9734e11;
    Asse.coef_neg[1] = -4.4865e9;
    Asse.coef_neg[2] = -5.9810e6;
    Asse.coef_neg[3] = -1.3232e3;
}

/*-----*/
/* Funzione per l'inizializzazione dei parametri del risonatore */
/*-----*/
void initHelmholtz()
{
    // PARAMETRI HELMHOLTZ
    Helmholtz.d_collo = 12.4/1000.0;
    Helmholtz.l_collo = 1500.0/1000.0;
    Helmholtz.d = 50.0/1000.0;

    Helmholtz.l_eq = Helmholtz.l_collo + 16.0 * Helmholtz.d_collo / (3.0 * PI);
}

```



```

    Helmholtz.d_collo_su_d = Helmholtz.d_collo/Helmholtz.d;

    Helmholtz.l_correzione = 0.035;
}

/*-----*/
/* Funzioni per i regolatori degli assi. */
/*-----*/
// A. Funzione di inizializzazione dei parametri dei regolatori.
void initRegolatore()
{
    // Inizializzazione struttura dati del regolatore
    memset(&R, 0x0, sizeof(Regolatore_data));

    // Assegnamento dati assi, regolatore asse e stato asse
    R.tipoReg = no_reg;
    R.stato_Asse = &statoNow;
    R.dati_Asse = &Asse;

    // Assegnamento parametri regolatore PID in posizione
    R.REG_pos.Kp = 777.0;
    R.REG_pos.Kd = 0.1;
    R.REG_pos.Ki = 2.0;
    R.REG_pos.N = 125.66;

    R.REG_pos.uMAX = 10.0;
    R.REG_pos.usaFB_Pos = ON;

    // Calcolo dei parametri del regolatore in posizione
    R.REG_pos.Td = R.REG_pos.Kd / R.REG_pos.Kp;
    R.REG_pos.Ti = R.REG_pos.Kp / R.REG_pos.Ki;
    R.REG_pos.a = R.REG_pos.Kp * Tc / R.REG_pos.Ti;
    R.REG_pos.b = R.REG_pos.Td / (R.REG_pos.N * Tc + R.REG_pos.Td);
    R.REG_pos.c = R.REG_pos.Kp * R.REG_pos.N * R.REG_pos.b;
}

// B. Funzione per l'inizializzazione delle memorie del regolatore PID in
// posizione.
void initRegolaPOS(Regolatore_data *reg)
{
    // Inializzo a zero le memorie del regolatore
    reg->REG_pos.ingresso [0] = reg->REG_pos.ingresso [1] = 0.0;
    reg->REG_pos.uP = reg->REG_pos.uD = reg->REG_pos.uI = 0.0;
    reg->REG_pos.uscita = 0.0;
}

```

```

// C. Funzione per la determinazione dell'azione di controllo mediante
// regolatore PID.
double RegolaPOS(Regolatore_data *reg)
{
    double azione = 0.0;

    //-----//
    // Determinazione dell'azione di controllo PID per la posizione //
    //-----//
    if (reg->REG_pos.usaFB.Pos == ON)
    {
        // Errore in (k-1)
        reg->REG_pos.ingresso [1] = reg->REG_pos.ingresso [0];

        // Errore in (k)
        reg->REG_pos.ingresso [0] = reg->SP_Asse.posizione -
            reg->stato_Asse->posLin0;

        reg->REG_pos.uP = reg->REG_pos.Kp * reg->REG_pos.ingresso [0];
        reg->REG_pos.uD = reg->REG_pos.b * reg->REG_pos.uD +
            reg->REG_pos.c * (reg->REG_pos.ingresso [0] -
                reg->REG_pos.ingresso [1]);

        reg->REG_pos.uscita = reg->REG_pos.uP + reg->REG_pos.uI +
            reg->REG_pos.uD;

        if (reg->REG_pos.uscita > reg->REG_pos.uMAX)
            reg->REG_pos.uscita = reg->REG_pos.uMAX;
        else if (reg->REG_pos.uscita < (-reg->REG_pos.uMAX))
            reg->REG_pos.uscita = -reg->REG_pos.uMAX;
        else
            reg->REG_pos.uI += reg->REG_pos.a * reg->REG_pos.ingresso [0];

        azione += reg->REG_pos.uscita;
        reg->REG_pos.comp_FB = reg->REG_pos.uscita;
    }
    return azione;
}

// D. Funzione che fornisce l'azione di controllo a seconda del regolatore
// scelto per il controllo di un determinato asse della macchina.
void RegolaAsse(Regolatore_data *reg)
{
    switch(reg->tipoReg)
    {
        case no_reg:
            {
                reg->I_comando = reg->SP_Asse.corrente;
            }
    }
}

```

```
        break;
    }

    case reg_pos:
    {
        reg->I_comando = RegolaPOS(reg);

        break;
    }
}

// E. Funzione che imposta all'interno della struttura dati regolatore il
// corretto setpoint a seconda del regolatore scelto.
void impostaSetPoint(Regolatore_data *reg, double sp0, double sp1, double sp2)
{
    // Impostazione del Setpoint sul valore desiderato
    switch(reg->tipoReg)
    {
        case no_reg:
        {
            reg->SP_Asse.corrente = sp0;

            break;
        }

        case reg_pos:
        {
            reg->SP_Asse.posizione = sp0;
            reg->SP_Asse.velocita = sp1;
            reg->SP_Asse.accelerazione = sp2;

            break;
        }
    }
}

// F. Funzione che chiama le funzioni di inizializzazione delle memorie dei
// singoli regolatori e imposta il setpoint al valore desiderato.
void inizializzaRegAsse(Regolatore_data *reg, double sp0, double sp1, double sp2)
{
    // Impostazione del Setpoint sullo stato attuale del sistema
    switch(reg->tipoReg)
    {
        default:
        case no_reg:
```

```

    {
    reg->SP_Asse.corrente = sp0;
    reg->SP_Asse.posizione = 0.0;
    reg->SP_Asse.velocita = 0.0;
    reg->SP_Asse.accelerazione = 0.0;

    break;
    }

    case reg_pos:
    {
    initRegolaPOS(reg);

    reg->SP_Asse.corrente = 0.0;
    reg->SP_Asse.posizione = sp0;
    reg->SP_Asse.velocita = sp1;
    reg->SP_Asse.accelerazione = sp2;

    break;
    }
}
}

/*-----*/
/* Funzioni - Leggi di moto */
/*-----*/
// A. Funzione che determina il setpoint adimensionale mediante una legge
// ad accelerazione costante a tratti
void leggeAdimTagl(double eps1, double eps2, double xx, SPAsse_data *data)
{
    double cap, cam, delta, cv;

    cap = 2.0/(eps1*(1.0+eps2));
    cam = 2.0/((1.0+eps2)*(1.0-eps1-eps2));
    delta = (1.0+eps2)/2.0;
    cv = 1.0/delta;

    if (xx <= eps1)
    {
        data->accelerazione = cap;
        data->velocita = (cv/eps1)*xx;
        data->posizione = 1.0/2.0*(cv/eps1)*pow(xx,2);
    }
    else if ((xx > eps1) && (xx <= (eps1+eps2)))
    {
        data->accelerazione = 0.0;
        data->velocita = cv;
        data->posizione = 1.0/2.0*cv*eps1+cv*(xx-eps1);
    }
}

```

```
    }
    else
    {
        data->accelerazione = -cam;
        data->velocita = cv-cam*(xx-eps1-eps2);
        data->posizione = 1.0/2.0*cv*eps1+cv*eps2+cv*(xx-eps1-eps2)-
            cam*pow((xx-eps1-eps2),2)/2.0;
    }
}

// B. Funzione che determina il setpoint adimensionale mediante una legge
// polinomiale di tipo 345
void polinomial345(double xx, SPAsse_data *data)
{
    double c[6] = {0.0, 0.0, 0.0, 10.0, -15.0, 6.0};

    data->posizione = c[3]*pow(xx,3) + c[4]*pow(xx,4) + c[5]*pow(xx,5);
    data->velocita = 3.0*c[3]*pow(xx,2) + 4.0*c[4]*pow(xx,3) + 5.0*c[5]*pow(xx,4);
    data->accelerazione = 6.0*c[3]*xx + 12.0*c[4]*pow(xx,2) + 20.0*c[5]*pow(xx,3);
}

// C. Funzione che determina il setpoint adimensionale mediante una legge
// polinomiale di tipo 357
void polinomial357(double xx, SPAsse_data *data)
{
    double c[7] = {0.0, 0.0, 4.375, 0.0, -5.25, 0.0, 1.875};

    data->posizione = c[2]*pow(xx,3) + c[4]*pow(xx,5) + c[6]*pow(xx,7);
    data->velocita = 3.0*c[2]*pow(xx,2) + 5.0*c[4]*pow(xx,4) + 7.0*c[6]*pow(xx,6);
    data->accelerazione = 6.0*c[2]*xx + 20.0*c[4]*pow(xx,3) + 42.0*c[6]*pow(xx,5);
}

// D. Funzione che determina il setpoint adimensionale mediante una legge
// polinomiale di tipo 369
void polinomial369(double xx, SPAsse_data *data)
{
    double c[9] = {0.0, 0.0, 3.0, 0.0, 0.0, -3.0, 0.0, 0.0, 1.0};

    data->posizione = c[2]*pow(xx,3) + c[5]*pow(xx,6) + c[8]*pow(xx,9);
    data->velocita = 3.0*c[2]*pow(xx,2) + 6.0*c[5]*pow(xx,5) + 9.0*c[8]*pow(xx,8);
    data->accelerazione = 6.0*c[2]*xx + 30.0*c[5]*pow(xx,4) + 72.0*c[8]*pow(xx,7);
}
```

```

// E. Funzione che determina il setpoint adimensionale mediante una legge
// polinomiale di tipo 4567
void polinomial4567(double xx, SPAsse_data *data)
{
    data->posizione = 35.0*pow(xx,4) - 84.0*pow(xx,5) + 70.0*pow(xx,6) -
        20.0*pow(xx,7);
    data->velocita = 140.0*pow(xx,3) - 420.0*pow(xx,4) + 420.0*pow(xx,5) -
        140.0*pow(xx,6);
    data->accelerazione = 420.0*pow(xx,2) - 1680.0*pow(xx,3) + 2100.0*pow(xx,4) -
        840.0*pow(xx,5);
}

// F. Funzione che calcola il seno di ampiezza unitaria
void legge_seno_func(double xx, SPAsse_data *data)
{
    //data->posizione = sin(2.0 * PI * xx);
    data->posizione = cos(2.0 * PI * xx) - 1.0;
    data->velocita = 0.0;
    data->accelerazione = 0.0;
}

// G. Funzione che determina il setpoint adimensionale mediante una legge
// ad accelerazione costante negativa
void leggeDecelCost(double xx, SPAsse_data *data)
{
    double cv, ca;

    ca = -2.0;
    cv = 2.0;

    data->accelerazione = ca;
    data->velocita = cv + ca*xx;
    data->posizione = 1./2.*ca*pow(xx,2) + cv*xx;
}

// H. Funzione di inizializzazione della struttura dati della legge di moto
void init_LeggeMoto2(struct DatiLegge *dataLegge, TipoLeggeMoto tipo_legge,
    double pos_finale, double t_alzata, double pos_init, double t_init)
{
    dataLegge->tipoLegge = tipo_legge;
    dataLegge->t0_ass = t_init;
    dataLegge->pos_init = pos_init;
    dataLegge->pos_end = pos_finale;
    dataLegge->h = pos_finale - pos_init;
    dataLegge->ta = t_alzata;
    dataLegge->index = 0.0;
}

```

```

// Inizializzazione a zero delle memorie dei setpoint
memset(&dataLegge->SP_new,0x0, sizeof(SPAsse_data));
memset(&dataLegge->SP_old,0x0, sizeof(SPAsse_data));

dataLegge->SP_new.posizione = dataLegge->pos_init;
}

// I. Funzione di inizializzazione della struttura dati della legge di moto
void init_LeggeMoto3(struct DatiLegge *dataLegge, TipoLeggeMoto tipo_legge,
double pos_finale, double t_alzata, SPAsse_data *statoInit, double t_init)
{
dataLegge->tipoLegge = tipo_legge;
dataLegge->t0_ass = t_init;
dataLegge->pos_init = statoInit->posizione;
dataLegge->vel_init = statoInit->velocita;
dataLegge->acc_init = statoInit->accelerazione;
dataLegge->pos_end = pos_finale;
dataLegge->h = pos_finale - statoInit->posizione;
dataLegge->ta = t_alzata;
dataLegge->index = 0.0;

// Inizializzazione a zero delle memorie dei setpoint
memset(&dataLegge->SP_new,0x0, sizeof(SPAsse_data));
memset(&dataLegge->SP_old,0x0, sizeof(SPAsse_data));

dataLegge->SP_new.posizione = dataLegge->pos_init;
dataLegge->SP_new.velocita = dataLegge->vel_init;
dataLegge->SP_new.accelerazione = dataLegge->acc_init;
}

// L. Funzione per la scelta della legge di moto desiderata
void calcSetPointAdim(TipoLeggeMoto tipoLegge, double xx, SPAsse_data *dataSP)
{
switch(tipoLegge)
{
case legge345:
{
polynomial345(xx, dataSP);
break;
}

case legge357:
{
polynomial357(xx, dataSP);
break;
}
}
}

```

```

    case legge369:
    {
        polinomial369(xx, dataSP);
        break;
    }

    case legge4567:
    {
        polinomial4567(xx, dataSP);
        break;
    }

    case leggeTagl_unTerzo:
    {
        leggeAdimTagl(1.0/3.0, 1.0/3.0, xx, dataSP);
        break;
    }

    case leggeTagl_unDecimo:
    {
        leggeAdimTagl(1.0/10.0, 8.0/10.0, xx, dataSP);
        break;
    }

    case legge_seno:
    {
        legge_seno_func(xx, dataSP);
        break;
    }

    case legge_decelCost:
    {
        leggeDecelCost(xx, dataSP);
        break;
    }
}

}

// M. Funzione il calcolo del setpoint dimensionale
void calcSetPoint(double tempo, struct DatiLegge *dataLegge, int *stop,
    SPAsse_data *dataSP)
{
    SPAsse_data SPadim;

    if (dataLegge->index <= 1.0)
    {
        // Valore OLD setpoint = Valore NEW
        dataLegge->SP_old.posizione = dataLegge->SP_new.posizione;
    }
}

```



```
dataLegge->SP_old.velocita = dataLegge->SP_new.velocita;
dataLegge->SP_old.accelerazione = dataLegge->SP_new.accelerazione;

// Calcolo dell'indice temporale in cui interrogare la legge di moto
dataLegge->index = (tempo-dataLegge->t0_ass)/dataLegge->ta;

// Calcolo del setpoint adimensionale
calcSetPointAdim(dataLegge->tipoLegge, dataLegge->index, &SPadim);

// Calcolo del setpoint dimensionale di posizione
dataSP->posizione = dataLegge->pos_init + dataLegge->h*SPadim.posizione;

// Calcolo del setpoint dimensionale di velocità e accelerazione
if (dataLegge->index <= 1e-6)
{
    dataSP->velocita = 0.0;
    dataSP->accelerazione = 0.0;
}
else
{
    // Setpoint di velocità
    dataSP->velocita = (dataSP->posizione -
        dataLegge->SP_old.posizione) / Tc;

    // Setpoint di accelerazione
    dataSP->accelerazione = (dataSP->velocita -
        dataLegge->SP_old.velocita) / Tc;
}

dataLegge->SP_new.posizione = dataSP->posizione;
dataLegge->SP_new.velocita = dataSP->velocita;
dataLegge->SP_new.accelerazione = dataSP->accelerazione;

// Flag di raggiungimento alzata: OFF
*stop = 0;
}
else
{
    // Il setpoint in posizione viene mantenuto al valore finale
    dataSP->posizione = dataLegge->pos_init + dataLegge->h;
    //dataLegge->SP_new.posizione = dataSP->posizione;

    // Il setpoint in velocità viene mantenuto a zero
    dataSP->velocita = 0.0;
    //dataLegge->SP_new.velocita = dataSP->velocita;

    // Il setpoint in accelerazione viene mantenuto a zero
    dataSP->accelerazione = 0.0;
    //dataLegge->SP_new.accelerazione = dataSP->accelerazione;
```

```

        // Flag di raggiungimento alzata: ON
        *stop = 1;
    }
}

/*-----*/
/* Funzioni legate al tempo fittizio di esecuzione delle leggi di moto. */
/*-----*/
// A. Funzione per il calcolo del parametro k_eff per la regolazione della
// velocità di esecuzione delle leggi di moto
double calc_K(double k)
{
    double ta=0.5;
    static double index_k=0.0;
    static double k_old=1.0;
    static double h_k=0.0;
    static double k_eff=1.0;
    SPAsse_data s;

    if (k == 0.0)
        k = 0.01;

    if (index_k == 0.0)
        h_k = k-k_old;

    if (h_k == 0.0)
    {
        k_eff = k;
    }
    else
    {
        polinomial345(index_k,&s);
        index_k += Tc/ta;
        k_eff = k_old + h_k*s.posizione;
    }

    if (index_k > 1.0)
    {
        index_k = 0.0;
        k_old = k;
    }

    return k_eff;
}

```

```

// B. Funzione per il calcolo del tempo effettivo t_tilde di esecuzione delle
// leggi di moto
void calcTime(double k_eff)
{
    t_tilde += k_eff*Tc;
}

double calcTime2(double k_eff, double time_old)
{
    return time_old + k_eff*Tc;
}

/*-----*/
/* Funzioni - Verifica dello stato del sistema. */
/*-----*/
TipoEsecuzioneFase func_systOk(TipoEsecuzioneFase faseIn)
{
    TipoEsecuzioneFase stato_subfunc=FaseExec;
    int i;
    int flagPos, flagVel;
    int asse_fermo;
    static int ok_fermo;

    double deltaPos_amm = 0.01;
    double deltaVel_amm = 0.001;

    // Inizializzazione variabili funzione
    if (faseIn == FaseInit)
        ok_fermo = OFF;

    // Esecuzione verifica sistema fermo
    flagPos = (fabs(R.SP_Asse.posizione - R.stato_Asse->posLin0)) <= deltaPos_amm;
    flagVel = (fabs(0.0 - R.stato_Asse->velLin)) <= deltaVel_amm;

    if ((flagPos == ON) && (flagVel == ON))
    {
        if (++ok_fermo > 10)
            asse_fermo = ON;
    }
    else
        ok_fermo = asse_fermo = OFF;

    if (asse_fermo == ON)
        stato_subfunc = FaseEnd;

    return stato_subfunc;
}

```

```

/*-----*/
/* Funzioni - Generazione SP per il controllo del moto. */
/*-----*/
TipoEsecuzioneFase calcSetPointRisonatore(TipoEsecuzioneFase faseIn ,
    Regolatore_data *reg, enumTypeMoto TypeMoto, int startCtrlMoto,
    int stopCtrlMovimento, int stopCtrlMoto, double posLike,
    datiMotoCtrlRisonatore *datiCtrlRis)
{
    TipoEsecuzioneFase stato_funcSPris=FaseExec;

    double RPM_max=4000;           // Max. velocità default x esec. traj [RPM]
    double vel_lin_max=0;
    bool changeMoto;
    static enumTypeMoto TypeMoto_old;
    SPAsse_data *SP=&datiCtrlRis->SP;
    int stop=OFF;                  // Flag: termine generazione SP
    int ok_posEnd=OFF;            // Flag: raggiunte soglie limite sul SP
    double ABSerrPOS;            // Errore in posiz. [m]
    double ABSerrVEL;            // Errore in veloc. [m/s]
    SPAsse_data SP_init;         // Setpoint iniz. prima dell'arresto
    static double pos_end;
    int check_time=OFF;          // Flag temporale

    // Inizializzazione variabili funzione
    if (faseIn == FaseInit)
    {
        // Configurazione della posizione iniziale
        datiCtrlRis->posNow = statoNow.posLin0;
        datiCtrlRis->posInizialeMoto = datiCtrlRis->posNow;

        // Inizializzazione su moto manuale
        TypeMoto_old = MuoviAutomatico;

        // Impostazione passo iniziale
        datiCtrlRis->passo_calcSPris = p_initSPctrl;
    }

    switch(datiCtrlRis->passo_calcSPris)
    {
        case p_initSPctrl:
            {
                // Impostazione dei setpoint iniziali
                SP->posizione = datiCtrlRis->posInizialeMoto;
                SP->velocita = 0.0;
                SP->accelerazione = 0.0;

                pos_end = datiCtrlRis->posInizialeMoto;
            }
    }
}

```

```
// Passo successivo
datiCtrlRis->passo_calcSPris = p_waitUtenteSPctrl;

break;
}

case p_waitUtenteSPctrl:
{
// ON flag attesa comando utente
waitUtenteSPctrl = ON;

// Attesa che l'utente scelga di iniziare il controllo del moto
if (startCtrlMoto == ON)
{
// OFF flag attesa comando utente
waitUtenteSPctrl = OFF;

if (TypeMoto == MuoviAutomatico)
{
// Inizializzazione variabili
datiCtrlRis->passo_calcSPris = p_initMuoviAutoCtrl;
}
else if (TypeMoto == MuoviManuale)
{
// Inizializzazione variabili
pos_end = posLike;

// Passo successivo
datiCtrlRis->passo_calcSPris = p_initMuoviManualeCtrl;
}
else
datiCtrlRis->passo_calcSPris = p_waitUtenteSPctrl;
}
else
datiCtrlRis->passo_calcSPris = p_waitUtenteSPctrl;

// Verifica se l'utente vuole uscire dal controllo del moto
if (stopCtrlMoto == ON)
datiCtrlRis->passo_calcSPris = p_endSPctrl;

break;
}

case p_initMuoviAutoCtrl:
{
// Tipo legge di moto
datiCtrlRis->legge = leggeTagl_unDecimo;
```

```

// Determinazione della posizione finale
pos_end = calcPosHelmholtz();
debug[30] = pos_end;

// Controllo posizione desiderata utente
if (pos_end > reg->dati_Asse->limPosMax)
    datiCtrlRis->posEnd = reg->dati_Asse->limPosMax - 0.002;
else if (pos_end < reg->dati_Asse->limPosMin)
    datiCtrlRis->posEnd = reg->dati_Asse->limPosMin + 0.002;
else
    datiCtrlRis->posEnd = pos_end;

// Calcolo della velocità lineare massima
vel_lin_max = (2.0 * PI) * RPM_max / 60.0 * Asse.tau;

// Calcolo del nuovo tempo di alzata
datiCtrlRis->t_alzata = (3.0/2.0)*fabs(datiCtrlRis->posEnd -
    datiCtrlRis->posInizialeMoto)/vel_lin_max;
if (datiCtrlRis->t_alzata < 1.0)
    datiCtrlRis->t_alzata = 1.0;

// Inizializzazione legge di moto
init_LeggeMoto2(&datiCtrlRis->dataLegge, datiCtrlRis->legge,
    datiCtrlRis->posEnd, datiCtrlRis->t_alzata,
    datiCtrlRis->posInizialeMoto, t_tilde);

// Passo successivo
datiCtrlRis->passo_calcSPris = p_execMuoviAutoCtrl;

break;
}

case p_execMuoviAutoCtrl:
{
if (stopCtrlMovimento == ON)
    datiCtrlRis->passo_calcSPris = p_initFermaSystCtrl;
else
{
// Generazione SP
calcSetPoint(t_tilde, &datiCtrlRis->dataLegge, &stop, SP);

// Verifica termine generazione SP
if ((stop == ON))
{
ABSerrPOS = fabs(datiCtrlRis->dataLegge.pos_end -
    reg->stato_Asse->posLin0);
ABSerrVEL = fabs(0.0 - reg->stato_Asse->velLin);
if (ABSerrPOS <= 0.002)
    if (ABSerrVEL <= 0.005)

```

```

        ok_posEnd = ON;

        if (ok_posEnd == ON)
            datiCtrlRis->passo_calcSPris = p_waitRestartMuoviAutoCtrl;
    }
}
break;
}

case p_initMuoviManualeCtrl:
{
    // Tipo legge di moto
    datiCtrlRis->legge = leggeTagl_unDecimo;

    // Controllo posizione desiderata utente
    if (pos_end > reg->dati_Asse->limPosMax)
        datiCtrlRis->posEnd = reg->dati_Asse->limPosMax - 0.002;
    else if (pos_end < reg->dati_Asse->limPosMin)
        datiCtrlRis->posEnd = reg->dati_Asse->limPosMin + 0.002;
    else
        datiCtrlRis->posEnd = pos_end;

    // Calcolo del tempo di alzata
    vel_lin_max = (2.0 * PI) * RPM_max / 60.0 * Asse.tau;
    datiCtrlRis->t_alzata = (3.0/2.0)*fabs(datiCtrlRis->posEnd -
        datiCtrlRis->posInizialeMoto)/vel_lin_max;
    if (datiCtrlRis->t_alzata < 1.0)
        datiCtrlRis->t_alzata = 1.0;

    // Inizializzazione legge di moto
    init.LeggeMoto2(&datiCtrlRis->dataLegge, datiCtrlRis->legge,
        datiCtrlRis->posEnd, datiCtrlRis->t_alzata,
        datiCtrlRis->posInizialeMoto, t_tilde);

    // Passo successivo
    datiCtrlRis->passo_calcSPris = p_execMuoviManualeCtrl;

    break;
}

case p_execMuoviManualeCtrl:
{
    if (stopCtrlMovimento == ON)
        datiCtrlRis->passo_calcSPris = p_initFermaSystCtrl;
    else
    {
        // Generazione SP
        calcSetPoint(t_tilde, &datiCtrlRis->dataLegge, &stop, SP);
    }
}

```

```

    // Verifica termine generazione SP
    if ((stop == ON))
    {
        ABSerrPOS = fabs(datiCtrlRis->dataLegge.pos_end -
            reg->stato_Asse->posLin0);
        ABSerrVEL = fabs(0.0 - reg->stato_Asse->velLin);
        if (ABSerrPOS <= 0.002)
            if (ABSerrVEL <= 0.005)
                ok_posEnd = ON;

        if (ok_posEnd == ON)
            datiCtrlRis->passo_calcSPris = p_restartSPctrl;
    }
}
break;
}

case p_initFermaSystCtrl:
{
    if (SP->velocita == 0.0)
        // Passo successivo
        datiCtrlRis->passo_calcSPris = p_waitUtenteSPctrl;
    else
    {
        SP_init.posizione = SP->posizione;
        SP_init.velocita = SP->velocita;
        SP_init.accelerazione = 0.0;
        pos_end = SP_init.posizione + fabs(SP_init.velocita)/
            SP_init.velocita*0.005;
        init.LeggeMoto3(&datiCtrlRis->dataLegge, legge_decelCost,
            pos_end, 1.0, &SP_init, Time);

        // Passo successivo
        datiCtrlRis->passo_calcSPris = p_FermaSystCtrl;
    }
}
break;
}

case p_FermaSystCtrl:
{
    // Generazione SP
    calcSetPoint(Time, &datiCtrlRis->dataLegge, &stop, SP);

    // Verifica termine generazione SP
    if ((stop == ON))
    {
        ABSerrPOS = fabs(datiCtrlRis->dataLegge.pos_end -
            reg->stato_Asse->posLin0);
        ABSerrVEL = fabs(0.0 - reg->stato_Asse->velLin);
    }
}

```



```
        if (ABSerrPOS <= 0.002)
            if (ABSerrVEL <= 0.005)
                ok_posEnd = ON;

        if (ok_posEnd == ON)
            datiCtrlRis->passo_calcSPris = p_restartSPctrl;
    }
    break;
}

case p_restartSPctrl:
{
    // Aggiorno posizione iniziale moto
    datiCtrlRis->posInizialeMoto = SP->posizione;

    // Passo successivo
    datiCtrlRis->passo_calcSPris = p_initSPctrl;

    break;
}

case p_waitRestartMuoviAutoCtrl:
{
    check_time = func_wait(2.0);

    if (check_time == ON)
        // Passo successivo
        datiCtrlRis->passo_calcSPris = p_restartMuoviAutoCtrl;

    break;
}

case p_restartMuoviAutoCtrl:
{
    // Aggiorno posizione iniziale moto
    datiCtrlRis->posInizialeMoto = SP->posizione;

    // Passo successivo
    datiCtrlRis->passo_calcSPris = p_initMuoviAutoCtrl;

    break;
}

case p_endSPctrl:
{
    stato_funcSPris = FaseEnd;

    break;
}
```

```

    }
    TypeMoto_old = TypeMoto;

    return stato_funcSPris;
}

```

### File “func\_moto.h”

```

/*-----*/
/* Prototipi di funzione */
/*-----*/
void initDatiAsse(void);
void initRegolatore(void);
void initHelmholtz(void);
void RegolaAsse(Regolatore_data *reg);
void impostaSetPoint(Regolatore_data *reg, double sp0, double sp1, double sp2);
void inizializzaRegAsse(Regolatore_data *reg, double sp0, double sp1, double sp2);

void init_LeggeMoto(struct DatiLegge *dataLegge, TipoLeggeMoto tipo_legge,
    double pos_finale, double t_alzata, double pos_init);
void init_LeggeMoto2(struct DatiLegge *dataLegge, TipoLeggeMoto tipo_legge,
    double pos_finale, double t_alzata, double pos_init, double t_init);
double calc_K(double k);
void calcTime(double k_eff);
double calcTime2(double k_eff, double time_old);
double calc_Amp(double amp);
void calcSetPoint(double tempo, struct DatiLegge *dataLegge, int *stop,
    SPAsse_data *dataSP);
void calcSetPoint2(double tempo, struct DatiLegge *dataLegge, SPAsse_data *dataSP,
    double amp);

TipoEsecuzioneFase func_systOk(TipoEsecuzioneFase faseIn);

TipoEsecuzioneFase calcSetPointRisonatore(TipoEsecuzioneFase faseIn,
    Regolatore_data *reg, enumTypeMoto TypeMoto, int startCtrlMoto,
    int stopCtrlMovimento, int stopCtrlMoto, double posLike,
    datiMotoCtrlRisonatore *datiCtrlRis);

```

### File “func\_statimac.c”

```

/*-----*/
/* Include files */
/*-----*/
#include <NIDAQmx.h>
#include <analysis.h>
#include <ansi_c.h>
#include <utility.h>
#include "dati_header.h"

```

```

#include "func_lib.h"
#include "func_moto.h"

/*-----*/
/* Definizione delle costanti */
/*-----*/
// Definizione parametri canali PXI e sensori
const double min_Vin=-5.0; // Tensione minima ammessa in input [V]
const double max_Vin=5.0; // Tensione massima ammessa in input [V]
const double min_Vout=-10.0; // Tensione minima ammessa in output [V]
const double max_Vout=10.0; // Tensione massima ammessa in output [V]

/*-----*/
/* Definizione e inizializzazione variabili globali */
/*-----*/
// Variabili temporali di ciclo macchina
double Tc = 0.001; // Tempo ciclo [s]
int tickCount = 0; // Contatore cicli esecuzione loop principale
double t_tilde = 0; // Tempo ciclo virtuale [s]
int passoStatoMAC = 0; // Passo interno alla funzione stato macchina

// Variabile per debug errori
double debug[100];
int res = 0; // Errore sui task

// Variabili controllo FAULTS
struct FaultMacchina tipoFault;
int errorHW=0, errorSW=0;
int errorHW_FC=0;
int flag_FAULT=0;
int flag_FAULT_FC=0;

// Variabili lettura-scrittura digitale
uInt32 dig_out=0; // Uscite digitali
int stAlto[7]; // Identificazione segnale digitale alto
int stBasso[7]; // Identificazione segnale digitale alto
int statoDig[7]; // Ingressi digitali rielaborati

// Variabili configurazione attuale sistema
StatoAsse_data statoNow, statoOld;
StatoComp_data statoCompNow, statoCompOld;
StatoFisica_data statoFisicoNow, statoFisicoOld;
struct digitalStatus digStatus;

// Strutture dati per dati Asse, Setpoint e regolatori
Asse_data Asse; // Dati asse risonatore
Regolatore_data R; // Parametri per regolatore
Helmholtz_data Helmholtz; // Parametri risonatore

```

```

// Struttura dati esecuzione controllo del moto
datiMotoCtrlRisonatore motoCtrl;

// Altre variabili
int task_puliti = ON;           // Flag: task puliti
int waitUtenteSPctrl = OFF;    // Flag: attesa input comando utente
int bassobasso=0;

/*-----*/
/* Definizione e inizializzazione variabili enum per exec stati macchina */
/*-----*/
// Enum per esecuzione ATTESA OK HARDWARE
enum passi_exec_WaitOkHw
{
    p_initWaitOkHw=0,
    p_execWaitOkHw ,
    p_waitErrorDrive ,
    p_endWaitOkHw ,
} passo_WaitOkHw=0;

// Enum per esecuzione ATTESA ESECUZIONE ZERO
enum passi_exec_OkZero
{
    p_initOkZero=0,
    p_execOkZero ,
    p_waitXexecZero ,
    p_removeMeanAccOkZero ,
    p_endOkZero ,
} passo_OkZero=0;

// Enum per esecuzione ZERO MACCHINA
enum passi_exec_ZERO
{
    p_initZero=0,
    p_goToFC ,
    p_waitXgoToPos ,
    p_initGoToPos ,
    p_posLikeGoToPos ,
    p_checkFermoZero ,
    p_waitEndZero ,
    p_endZero ,
} passo_Zero=0;

// Enum per esecuzione PREREADEY
enum passi_exec_PREREADEY
{
    p_initPR=0,
    p_initLeggeMotoPR ,
    p_posLikePR ,

```

```
    p_checkFermoPR ,
    p_endPR ,
} passo_PreReady=0;

// Enum per esecuzione READY
enum passi_exec_READY
{
    p_initReady=0,
    p_waitUserOk ,
    p_waitEndReady ,
    p_endReady ,
} passo_Ready=0;

// Enum per esecuzione CONTROLLO DEL MOTO
enum passi_exec_CONTMOT
{
    p_initCM=0,
    p_initLeggeMotoCM ,
    p_execCM ,
    p_waitCM ,
    p_endCM ,
} passo_ContMoto=0;

// Enum per esecuzione Fault hardware: emergenza, limiti software
enum passi_exec_FAULT_HW
{
    p_initFHW ,
    p_waitEmergOffFHW ,
    p_emergOffFHW ,
    p_utenteFHW ,
    p_enableFHW ,
    p_endFHW ,
} passo_FaultHW=0;

// Enum per esecuzione Fault hardware: finecorsa elettrici
enum passi_exec_FAULT_HW_FC
{
    p_initFHW_FC=0,
    p_waitUtenteFHW_FC ,
    p_utenteFHW_FC ,
    p_endFHW_FC ,
} passo_FaultHW_FC=0;

// Enum per esecuzione TWHALT
enum passi_exec_TWHALT
{
    p_initTH=0,
    p_waitDisablePowerTH ,
    p_disablePowerTH ,
```

```

    p_waitEndTH,
    p_endTH,
} passo_TwHalt=0;

/*-----*/
/* SOTTOFUNZIONI DEGLI STATI MACCHINA */
/*-----*/

/*-----*/
/* FUNZIONE INIZIALIZZA DATI - func_InizializzaDati() */
/* */
/* HELP: */
/* 1. Questa funzione inizializza tutte le variabili e crea i task. */
/*-----*/
TipoEsecuzioneFase func_InizializzaDati(double tempo_ciclo)
{
    // Inizializzazione a zero degli stati di emergenza
    memset(&tipoFault, 0x0, sizeof(struct FaultMacchina));

    // Inizializzazione variabili globali
    Tc = tempo_ciclo;

    // Inizializzazione array di debug
    memset(debug, 0x0, 100*sizeof(double));

    // Inizializzazione array stato della macchina
    memset(&statoNow, 0x0, sizeof(StatoAsse_data));
    memset(&statoOld, 0x0, sizeof(StatoAsse_data));
    memset(&statoCompNow, 0x0, sizeof(StatoComp_data));
    memset(&statoCompOld, 0x0, sizeof(StatoComp_data));

    // Inizializzazione array segnali digitali
    memset(&digStatus, 0x0, sizeof(struct digitalStatus));
    memset(&stAlto, 0x0, 7*sizeof(int));
    memset(&stBasso, 0x0, 7*sizeof(int));
    memset(&statoDig, 0x0, 7*sizeof(int));

    // Attenzione alla sequenza: prima "initDatiAsse()" poi "initRegolatore()"
    // Inizializzazione parametri asse
    initDatiAsse();

    // Inizializzazione parametri regolatori
    initRegolatore();

    // Inizializzazione filtri
    initFilters();

    // Inizializzazione Helmholtz

```

```
    initHelmholtz();

    return FaseEnd;
}

/*-----*/
/* FUNZIONE STATO MACCHINA - func_OkAvvio */
/* */
/* HELP: */
/* 1. Questa funzione riceve in ingresso dal pannello una variabile che */
/* dice se l'utente ha premuto il pulsante OK AVVIO. */
/*-----*/
TipoEsecuzioneFase func_OkAvvio(int push_OkAvvio)
{
    TipoEsecuzioneFase stato=FaseExec;
    static int ttl=OFF;

    // Accensione degli stati di emergenza
    tipoFault.checkEMERG_error = OFF;
    tipoFault.checkFC_error = OFF;
    tipoFault.checkVELMAX_error = OFF;
    tipoFault.checkPOSMAX_error = OFF;

    if (ttl == OFF)
    {
        // Abilito segnale TTL sempre alto
        write_dig(TTL5Volt, ON);
        ttl = ON;
    }

    if (push_OkAvvio == ON)
    {
        // Stato funzione: termine
        stato = FaseEnd;
    }

    //return ((push_OkAvvio == 1) ? FaseEnd : FaseExec);
    return stato;
}
```

```

/*-----*/
/* FUNZIONE STATO MACCHINA - func_WaitOkHw */
/* */
/* HELP: */
/* 1. Questa funzione controlla se il quadro è acceso e pronto per eseguire */
/* gli stati macchina successivi. */
/*-----*/
TipoEsecuzioneFase func_WaitOkHw ()
{
    TipoEsecuzioneFase stato=FaseExec;
    int check_time=OFF;

    switch (passo_WaitOkHw)
    {
        case p_initWaitOkHw:
            {
                // Accensione degli stati di emergenza
                tipoFault.checkEMERG_error = OFF;
                tipoFault.checkFC_error = OFF;
                tipoFault.checkVELMAX_error = OFF;
                tipoFault.checkPOSMAX_error = OFF;

                passo_WaitOkHw = p_execWaitOkHw;

                break;
            }

        case p_execWaitOkHw:
            {
                if (digStatus.Em.IN == ON) // Emergenza spenta
                    passo_WaitOkHw = p_waitErrorDrive;

                break;
            }

        case p_waitErrorDrive:
            {
                check_time = func_wait (5.0);

                if (check_time == ON)
                    passo_WaitOkHw = p_endWaitOkHw;

                break;
            }

        case p_endWaitOkHw:
            {
                stato = FaseEnd;
            }
    }
}

```



```

        break;
    }
}
return stato;
}

/*-----*/
/* FUNZIONE STATO MACCHINA - func_OkZero */
/*
/* HELP: */
/* 1. Questa funzione riceve in ingresso dal pannello una variabile che */
/* dice se l'utente ha premuto il pulsante OK ESEGUI ZERO. */
/* 2. Abilita l'azionamento: Drive Enable=ON. */
/* 3. Attende 5 sec. per assestamento del sistema. */
/*-----*/
TipoEsecuzioneFase func_OkZero()
{
    TipoEsecuzioneFase stato=FaseExec;
    int i;
    int check_time=OFF;

    passoStatoMAC = passo_OkZero;

    switch(passo_OkZero)
    {
        case p_initOkZero:
            {
                // Accensione degli stati di emergenza
                tipoFault.checkEMERG_error = ON;
                tipoFault.checkFC_error = OFF;
                tipoFault.checkVELMAX_error = OFF;
                tipoFault.checkPOSMAX_error = OFF;

                // Passo successivo
                passo_OkZero = p_execOkZero;

                break;
            }

        case p_execOkZero:
            {
                // Abilito la linea di potenza
                write_dig(PowerEnable, ON);

                // Passo successivo
                passo_OkZero = p_waitXexecZero;

                break;
            }
    }
}

```

```

    }

    case p_waitXexecZero:
    {
        check_time = func_wait(5.0);

        if (check_time == ON)
            passo_OkZero = p_removeMeanAccOkZero;

        break;
    }

    case p_removeMeanAccOkZero:
    {
        // Abilito enable drive
        write_dig(DriveEnable, ON);

        // Passo successivo
        passo_OkZero = p_endOkZero;

        break;
    }

    case p_endOkZero:
    {
        stato = FaseEnd;

        break;
    }
}
return stato;
}

/*-----*/
/* FUNZIONE STATO MACCHINA - func_RunZero */
/* */
/* HELP: */
/* 1. Questa funzione esegue l'azzeramento dell'asse del risonatore: */
/* a) Porta gli assi verso i fincorsa con un controllo in velocità. */
/* b) Sposta gli assi in una posizione iniziale nota mediante un */
/* controllo in posizione. */
/*-----*/
TipoEsecuzioneFase func_RunZero(TipoEsecuzioneFase faseIn)
{
    int i, res=0;
    TipoEsecuzioneFase stato=FaseExec;
    static TipoEsecuzioneFase stato_subfunc;
    SPAsse_data SP; // Struttura contenente i SP

```

```

// VARIABILI PER CONTROLLO IN POSIZIONE
// POSIZIONE EFFETTIVA DEL SISTEMA NON NOTA
double SPvel=-0.010;           // Setpoint velocità motore [m/s]
double errSP=0.0;             // Errore sul setpoint in velocità [m/s]
static double time0;         // Tempo iniziali di inizio procedura zero
static int azzeroEnc;        // Flag che indica encoder azzerato
static int endZero;         // Flag fine procedura di ZERO

// VARIABILI PER CONTROLLO IN POSIZIONE
TipoLeggeMoto tipoLegge=leggeTagl_unTerzo;
int stop=0, ok_posEnd=0;
double posEnd=0.030;        // Posizione finale di riferimento [m]
double t_alzata=1.0;       // Tempo di esecuzione alzata [s]
static struct DatiLegge dataLegge; // Struttura contenente i dati della legge
double errPOS;              // Errore in posiz. [m]
double ABSerrPOS;          // Errore in posiz. [m]
double ABSerrVEL;          // Errore in veloc. [m/s]

int check_time=OFF;

// Inizializzazione variabili funzione
if (faseIn == FaseInit)
    passo_Zero = p_initZero;

passoStatoMAC = passo_Zero;

switch(passo_Zero)
{
    case p_initZero:
        {
            // Accensione degli stati di emergenza
            tipoFault.checkEMERG_error = ON;
            tipoFault.checkFC_error = OFF;
            tipoFault.checkVELMAX_error = ON;
            tipoFault.checkPOSMAX_error = OFF;

            // Attivo regolatore in posizione
            R.tipoReg = reg_pos;
            inizializzaRegAsse(&R, statoNow.posLin0, 0.0, 0.0);

            // Inizializzazione flag per esecuzione zero
            azzeroEnc = 0;
            endZero = 0;

            // Inizializzazione t_0
            time0 = Time;
        }
}

```

```

// Passo successivo
passo_Zero = p_goToFC;

break;
}

case p_goToFC:
{
//-----//
// Azzeramento ASSE //
//-----//
if ((digStatus.FC_Sx == OFF) && (endZero == OFF))
{
// Calcolo il setpoint in posizione in funz. della veloc.
SP.posizione = SPvel * (Time - time0);

// Impostazione valore setpoint
impostaSetPoint(&R, SP.posizione, SPvel, 0.0);
}
else
{
if (azzeroEnc == OFF)
{
// Azzeramento l'encoder
statoNow.sigEncZERO = statoNow.sigEnc;

// Spegnimento regolatore posizione
R.tipoReg = no_reg;
inizializzaRegAsse(&R, 0.0, 0.0, 0.0);

// Accensione flag encoder azzerato
azzeroEnc = ON;
}
else if (azzeroEnc == ON)
{
if (endZero == OFF)
{
// Riaccensione regolatore posizione
R.tipoReg = reg_pos;
inizializzaRegAsse(&R, statoNow.posLin0, 0.0, 0.0);

// Accensione flag fine procedura zero
endZero = ON;
}
}
}
}

```

```
//-----//  
// Controllo fine azzeramento ASSI //  
//-----//  
if ((endZero == ON))  
    passo_Zero = p_waitXgoToPos;  
  
break;  
}  
  
case p_waitXgoToPos:  
{  
    check_time = func_wait(0.5);  
  
    if (check_time == ON)  
        passo_Zero = p_initGoToPos;  
  
    break;  
}  
  
case p_initGoToPos:  
{  
    tipoLegge = leggeTagl_unTerzo;  
    posEnd = 0.030; t_alzata=5.0;  
  
    init_LeggeMoto2(&dataLegge, tipoLegge, posEnd, t_alzata,  
        statoNow.posLin0, Time);  
  
    // Passo successivo  
    passo_Zero = p_posLikeGoToPos;  
  
    break;  
}  
  
case p_posLikeGoToPos:  
{  
    calcSetPoint(Time, &dataLegge, &stop, &SP);  
    impostaSetPoint(&R, SP.posizione, SP.velocita, SP.accelerazione);  
  
    // Verifica raggiungimento setpoint  
    if ((stop == ON))  
    {  
        ABSerrPOS = fabs(dataLegge.pos_end - statoNow.posLin0);  
        ABSerrVEL = fabs(0 - statoNow.velLin);  
        if (ABSerrPOS <= 0.050)  
            if (ABSerrVEL <= 0.01)  
                ok_posEnd = ON;  
  
        if (ok_posEnd == ON)  
            passo_Zero = p_waitEndZero;
```

```

    }
    break;
}

case p_checkFermoZero:
{
    stato_subfunc = func_systOk(stato_subfunc);

    if (stato_subfunc == FaseEnd)
        passo_Zero = p_waitEndZero;

    break;
}

case p_waitEndZero:
{
    check_time = func_wait(2.0);

    if (check_time == ON)
        passo_Zero = p_endZero;

    break;
}

case p_endZero:
{
    stato = FaseEnd;

    break;
}
}
return stato;
}

/*-----*/
/* FUNZIONE STATO MACCHINA - func_PreReady */
/* */
/* HELP: */
/* 1. Questa funzione porta il risonatore nella posizione di corsa/2, */
/* attraverso un controllo in posizione. */
/*-----*/
TipoEsecuzioneFase func_PreReady(TipoEsecuzioneFase faseIn)
{
    TipoEsecuzioneFase stato=FaseExec;

    int i=0, stop=0, ok_posEnd=0;
    static TipoEsecuzioneFase stato_subfunc;
    SPAsse_data SP; // Struttura contenente i SP

```

```

TipoLeggeMoto tipoLegge;           // Variabile leggi di moto
double posEnd;                     // Posizione finale di riferimento [m]
double t_alzata=2.0;               // Tempo di esecuzione alzata [s]
static struct DatiLegge dataLegge; // Dati della legge di moto
double ABSerrPOS;                  // Errore in posiz. [m]
double ABSerrVEL;                  // Errore in veloc. [m/s]

double RPM_max=4000;
double vel_lin_max=0;

// Inizializzazione variabili funzione
if (faseIn == FaseInit)
    passo_PreReady=p_initPR;

passoStatoMAC = passo_PreReady;

switch(passo_PreReady)
{
    case p_initPR :
        {
            // Accensione degli stati di emergenza
            tipoFault.checkEMERG_error = ON;
            tipoFault.checkFC_error = ON;
            tipoFault.checkVELMAX_error = ON;
            tipoFault.checkPOSMAX_error = ON;

            // Attivo regolatore in posizione
            R.tipoReg = reg_pos;
            inizializzaRegAsse(&R, statoNow.posLin0, 0.0, 0.0);

            // Passo successivo
            passo_PreReady = p_initLeggeMotoPR;

            break;
        }

    case p_initLeggeMotoPR:
        {
            tipoLegge = leggeTagl_unTerzo;
            posEnd = Asse.corsa/2.0;

            // Calcolo del tempo di alzata
            vel_lin_max = (2.0 * PI) * RPM_max / 60.0 * Asse.tau;
            t_alzata = (3.0/2.0)*fabs(posEnd - statoNow.posLin0)/vel_lin_max;

            //posEnd = Asse.limPosMin+0.035; t_alzata=5.0;
            init_LeggeMoto2(&dataLegge, tipoLegge, posEnd, t_alzata,
                statoNow.posLin0, Time);
        }
}

```

```
        // Passo successivo
        passo_PreReady = p_posLikePR;

        break;
    }

    case p_posLikePR:
    {
        calcSetPoint(Time, &dataLegge, &stop, &SP);
        impostaSetPoint(&R, SP.posizione, SP.velocita, SP.accelerazione);

        // Verifica raggiungimento setpoint
        if ((stop == ON))
        {
            ABSerrPOS = fabs(dataLegge.pos_end - statoNow.posLin0);
            ABSerrVEL = fabs(0 - statoNow.velLin);
            if (ABSerrPOS <= 0.050)
                if (ABSerrVEL <= 0.01)
                    ok_posEnd = ON;

            if (ok_posEnd == ON)
                passo_PreReady = p_endPR;
        }
        break;
    }

    case p_checkFermoPR:
    {
        stato_subfunc = func_systOk(stato_subfunc);

        if (stato_subfunc == FaseEnd)
            passo_PreReady = p_endPR;

        break;
    }

    case p_endPR:
    {
        stato = FaseEnd;

        break;
    }
}
return stato;
}
```



```

/*-----*/
/* FUNZIONE STATO MACCHINA - func_Ready */
/*-----*/
/*
/* HELP:
/* 1. Questa funzione mantiene l'asse del risonatore nella posizione
/* desiderata, permettendo all'utente di effettuare scelte dal menu
/* di controllo
/*-----*/
TipoEsecuzioneFase func_Ready(TipoEsecuzioneFase faseIn , int user_ok)
{
    TipoEsecuzioneFase stato=FaseExec;
    int i=0;
    int check_time=OFF;

    // Inizializzazione variabili funzione
    if (faseIn == FaseInit)
        passo_Ready = p_initReady;

    passoStatoMAC = passo_Ready;

    switch(passo_Ready)
    {
        case p_initReady:
            {
                // Accensione degli stati di emergenza
                tipoFault.checkEMERG_error = ON;
                tipoFault.checkFC_error = ON;
                tipoFault.checkVELMAX_error = ON;
                tipoFault.checkPOSMAX_error = ON;

                passo_Ready = p_waitUserOk;

                break;
            }

        case p_waitUserOk:
            {
                if (user_ok == ON)
                    passo_Ready = p_waitEndReady;

                break;
            }

        case p_waitEndReady:
            {
                check_time = func_wait(1.0);

                if (check_time == ON)
                    passo_Ready = p_endReady;
            }
    }
}

```

```

        break;
    }

    case p_endReady:
    {
        stato = FaseEnd;

        break;
    }
}
return stato;
}

/*-----*/
/* FUNZIONE STATO MACCHINA - func_ContMoto */
/* */
/* HELP: */
/* Questa funzione muove sposta il sistema nella posizione desiderata */
/*-----*/
TipoEsecuzioneFase func_ContMoto(TipoEsecuzioneFase faseIn, int CtrlStartMoto,
    int CtrlStopMoto, int CtrlTypeMoto, int CtrlStopMovimento, double CtrlPosLike)
{
    TipoEsecuzioneFase stato=FaseExec;
    static TipoEsecuzioneFase stato_subfunc;

    int i=0, stop=0, ok_posEnd=0;
    int check_time=OFF; // Flag di attesa
    static int conta_cicli=0;

    static double posEnd; // Posizione finale di riferimento [m]
    double t_alzata=2.0; // Tempo di esecuzione alzata [s]
    static struct DatiLegge dataLegge; // Dati della legge di moto
    double errPOS; // Errore in posiz. [m]
    double ABSerrPOS; // Errore in posiz. [m]
    double ABSerrVEL; // Errore in veloc. [m/s]

    // Inizializzazione variabili funzione
    if (faseIn == FaseInit)
        passo_ContMoto = p_initCM;

    passoStatoMAC = passo_ContMoto;

    switch(passo_ContMoto)
    {
        case p_initCM:
        {
            // Accensione degli stati di emergenza
            tipoFault.checkEMERG_error = ON;

```

```
    tipoFault.checkFC_error = ON;
    tipoFault.checkVELMAX_error = ON;
    tipoFault.checkPOSMAX_error = ON;

    // Inizializzazione stato sottofunzione
    stato_subfunc = FaseInit;

    // Passo successivo
    passo_ContMoto = p_execCM;

    break;
}

case p_execCM:
{
    // Determinazione del setpoint
    stato_subfunc = calcSetPointRisonatore(stato_subfunc, &R,
        CtrlTypeMoto, CtrlStartMoto, CtrlStopMovimento, CtrlStopMoto,
        CtrlPosLike, &motoCtrl);

    // Impostazione del setpoint all'interno del regolatore
    impostaSetPoint(&R, motoCtrl.SP.posizione, motoCtrl.SP.velocita,
        motoCtrl.SP.accelerazione);

    // Passo successivo
    passo_ContMoto = (stato_subfunc == FaseEnd) ? p_waitCM : p_execCM;

    break;
}

case p_waitCM:
{
    check_time = func_wait(0.5);

    if (check_time == ON)
        passo_ContMoto = p_endCM;

    break;
}

case p_endCM:
{
    stato = FaseEnd;

    break;
}
}
return stato;
}
```

```

/*-----*/
/* FUNZIONE STATO MACCHINA - func_FaultHW */
/* */
/* HELP: */
/* 1. Si arriva in questo stato macchina se si è verificata una condizione */
/* di FAULT hardware: emergenza premuta o limitazioni software. */
/*-----*/
TipoEsecuzioneFase func_FaultHW(TipoEsecuzioneFase faseIn , int push_OkAvvio)
{
    TipoEsecuzioneFase stato=FaseExec;
    int i;
    int check_time=OFF;

    // Inizializzazione variabili funzione
    if (faseIn == FaseInit)
        passo_FaultHW = p_initFHW;

    passoStatoMAC = passo_FaultHW;

    switch (passo_FaultHW)
    {
        case p_initFHW:
            {
                // Accensione degli stati di emergenza
                tipoFault.checkEMERG_error = OFF;
                tipoFault.checkFC_error = OFF;
                tipoFault.checkVELMAX_error = OFF;
                tipoFault.checkPOSMAX_error = OFF;

                // Impostazione a zero delle uscite analogiche
                R.tipoReg = no_reg;
                inizializzaRegAsse(&R, 0.0, 0.0, 0.0);

                // Impostazione a OFF enable drive
                write_dig(DriveEnable, OFF);

                // Passo successivo
                passo_FaultHW = p_waitEmergOffFHW;

                break;
            }

        case p_waitEmergOffFHW:
            {
                check_time = func_wait(5.0);

                // Passo successivo
                if (check_time == ON)
                    passo_FaultHW = p-emergOffFHW;
            }
    }
}

```

```
        break;
    }

    case p_emergOffFHW:
    {
        // Passo successivo
        if (digStatus.Em_IN == ON) // Emergenza spenta
            passo_FaultHW = p_utenteFHW;

        break;
    }

    case p_utenteFHW:
    {
        if (push.OkAvvio == ON)
        {
            // Passo successivo
            passo_FaultHW = p_enableFHW;
        }
        break;
    }

    case p_enableFHW:
    {
        check_time = func_wait(5.0);

        if (check_time == ON)
        {
            // Abilito drive enable
            write_dig(DriveEnable, ON);

            // Passo successivo
            passo_FaultHW = p_endFHW;
        }
        break;
    }

    case p_endFHW:
    {
        // Spegnimento flag errori HW
        errorHW = flag_FAULT = OFF;

        stato = FaseEnd;
        break;
    }
}
return stato;
}
```

```

/*-----*/
/* FUNZIONE STATO MACCHINA - func_FaultHW_FC */
/*
/* HELP: */
/* 1. Si arriva in questo stato macchina se si è verificata una condizione */
/* di FAULT hardware: raggiungimento FC elettrici. */
/* 2. In questo caso è necessario rieffettuare azzeramento macchina. */
/*-----*/
TipoEsecuzioneFase func_FaultHW_FC(TipoEsecuzioneFase faseIn , int push_OkAvvio)
{
    TipoEsecuzioneFase stato=FaseExec;
    int i;
    int check_time=OFF;

    // Inizializzazione variabili funzione
    if (faseIn == FaseInit)
        passo_FaultHW_FC = p_initFHW_FC;

    passoStatoMAC = passo_FaultHW_FC;

    switch (passo_FaultHW_FC)
    {
        case p_initFHW_FC:
            {
                // Accensione degli stati di emergenza
                tipoFault.checkEMERG_error = OFF;
                tipoFault.checkFC_error = OFF;
                tipoFault.checkVELMAX_error = OFF;
                tipoFault.checkPOSMAX_error = OFF;

                // Impostazione a zero delle uscite analogiche
                R.tipoReg = no_reg;
                inizializzaRegAsse(&R, 0.0, 0.0, 0.0);

                // Impostazione a OFF enable drive
                write_dig(DriveEnable , OFF);

                // Passo successivo
                passo_FaultHW_FC = p_waitUtenteFHW_FC;

                break;
            }

        case p_waitUtenteFHW_FC:
            {
                check_time = func_wait(5.0);

                // Passo successivo
                if (check_time == ON)

```

```

        passo_FaultHW_FC = p_utenteFHW_FC;

        break;
    }

    case p_utenteFHW_FC:
    {
        if (push_OkAvvio == ON)
        {
            // Abilito drive enable
            write_dig(DriveEnable, ON);

            // Passo successivo
            passo_FaultHW_FC = p_endFHW_FC;
        }
        break;
    }

    case p_endFHW_FC:
    {
        // Spegnimento flag errori HW
        errorHW_FC = flag_FAULT_FC = OFF;

        stato = FaseEnd;

        break;
    }
}

return stato;
}

/*-----*/
/* FUNZIONE STATO MACCHINA - func_FaultSW */
/*-----*/
/*
/* HELP: */
/* 1. Si arriva in questo stato macchina se si è verificata una condizione */
/* di FAULT software: ad esempio un errore nell'inizializzazione dei task. */
/* Se si entra in questo stato macchina è necessario riavviare il sistema */
/*-----*/
TipoEsecuzioneFase func_FaultSW()
{
    TipoEsecuzioneFase stato=FaseEnd;

    return stato;
}

```

```

/*-----*/
/* FUNZIONE STATO MACCHINA - func_TwHalt */
/* */
/* HELP: */
/* 1. Funzione che prepara la macchina alla condizione di arresto sicuro. */
/*-----*/
TipoEsecuzioneFase func_TwHalt()
{
    TipoEsecuzioneFase stato=FaseExec;
    int i;
    int check_time=OFF;
    static int endTwHalt;

    passoStatoMAC = passo_TwHalt;

    switch (passo_TwHalt)
    {
        case p_initTH:
            {
                // Accensione degli stati di emergenza
                tipoFault.checkEMERG_error = OFF;
                tipoFault.checkFC_error = OFF;
                tipoFault.checkVELMAX_error = OFF;
                tipoFault.checkPOSMAX_error = OFF;

                // Impostazione a zero delle uscite analogiche
                R.tipoReg = no_reg;
                inizializzaRegAsse(&R, 0.0, 0.0, 0.0);

                // Impostazione a OFF enable drive
                write_dig(DriveEnable, OFF);

                // Passo successivo
                passo_TwHalt = p_waitDisablePowerTH;

                break;
            }

        case p_waitDisablePowerTH:
            {
                check_time = func_wait(5.0);

                if (check_time == ON)
                    // Passo successivo
                    passo_TwHalt = p_disablePowerTH;

                break;
            }
    }
}

```



```

    case p_disablePowerTH:
    {
        // Impostazione a OFF power drive
        write_dig(PowerEnable, OFF);

        // Passo successivo
        passo_TwHalt = p_waitEndTH;

        break;
    }

    case p_waitEndTH:
    {
        check_time = func_wait(0.2);

        if (check_time == ON)
            // Passo successivo
            passo_TwHalt = p_endTH;

        break;
    }

    case p_endTH:
    {
        stato = FaseEnd;

        break;
    }
}
return stato;
}

/*-----*/
/* FUNZIONE STATO MACCHINA - func_Halt */
/*-----*/
/* HELP: */
/* 1. Funzione che indica che la macchina si trova in condizioni di */
/* arresto sicuro. */
/*-----*/
TipoEsecuzioneFase func_Halt()
{
    TipoEsecuzioneFase stato=FaseEnd;

    return stato;
}

```

## File “func\_statiMAC.h”

```

/*-----*/
/* Prototipi di funzione */
/*-----*/
TipoEsecuzioneFase func_InizializzaDati(double tempo_ciclo);
TipoEsecuzioneFase func_OkAvvio(int push_OkAvvio);
TipoEsecuzioneFase func_WaitOkHw(void);
TipoEsecuzioneFase func_OkZero(void);
TipoEsecuzioneFase func_RunZero(TipoEsecuzioneFase faseIn);
TipoEsecuzioneFase func_PreReady(TipoEsecuzioneFase faseIn);
TipoEsecuzioneFase func_Ready(TipoEsecuzioneFase faseIn, int user_ok);
TipoEsecuzioneFase func_ContMoto(TipoEsecuzioneFase faseIn, int CtrlStartMoto,
    int CtrlStopMoto, int CtrlTypeMoto, int CtrlStopMovimento, double CtrlPosLike);
TipoEsecuzioneFase func_FaultHW(TipoEsecuzioneFase faseIn, int push_OkAvvio);
TipoEsecuzioneFase func_FaultHW_FC(TipoEsecuzioneFase faseIn, int push_OkAvvio);
TipoEsecuzioneFase func_FaultSW(void);
TipoEsecuzioneFase func_TwHalt(void);
TipoEsecuzioneFase func_Halt(void);

/*-----*/
/* Definizione delle costanti */
/*-----*/
// Definizione parametri canali PXI e sensori
extern const double min_Vin; // Tensione minima ammessa in input [V]
extern const double max_Vin; // Tensione massima ammessa in input [V]
extern const double min_Vout; // Tensione minima ammessa in output [V]
extern const double max_Vout; // Tensione massima ammessa in output [V]

/*-----*/
/* Definizione e inizializzazione variabili globali */
/*-----*/
// Variabili temporali di ciclo macchina
extern double Tc; // Tempo ciclo [s]
extern int tickCount; // Contatore cicli esecuzione loop
extern double t_tilde; // Tempo ciclo virtuale [s]
extern int passoStatoMAC; // Passo interno alla fz. stato macchina

// Variabile per debug errori
extern double debug[100];
extern int res; // Errore sui task

// Variabili controllo FAULTS
extern struct FaultMacchina tipoFault;
extern int errorHW, errorSW;
extern int errorHW_FC;
extern int flag_FAULT;
extern int flag_FAULT_FC;

```

```

// Variabili lettura-scrittura digitale
extern uInt32 dig_out;           // Uscite digitali
extern int stAlto [7];          // Identificazione segnale digitale alto
extern int stBasso [7];         // Identificazione segnale digitale alto
extern int statoDig [7];        // Ingressi digitali rielaborati

// Variabili configurazione attuale sistema
extern StatoAsse_data statoNow, statoOld;
extern StatoComp_data statoCompNow, statoCompOld;
extern StatoFisica_data statoFisicoNow, statoFisicoOld;
extern struct digitalStatus digStatus;

// Strutture dati per dati Asse, Setpoint e regolatori
extern Asse_data Asse;          // Dati asse risonatore
extern Regolatore_data R;       // Parametri regolatore
extern Helmholtz_data Helmholtz; // Parametri risonatore

// Struttura dati esecuzione controllo del moto
extern datiMotoCtrlRisonatore motoCtrl;

// Altre variabili
extern int task_puliti;         // Flag: pulizia task
extern int waitUtenteSPctrl;   // Flag: attesa input comando utente
extern int bassobasso;

```

## File “dati\_header.h”

```

#include "myfilter.h"

/*-----*/
/* Define */
/*-----*/
#define TASK_ANALOG_INPUT "ANALOG_INPUT"
#define TASK_ANALOG_OUTPUT "TASK_ANALOG_OUTPUT"
#define TASK_COUNTER_ANGULAR_ENC_RIS "ENCODER_RISONATORE"
#define TASK_COUNTER_ANGULAR_ENC_COMP "ENCODER_COMPRESSORE"
#define TASK_DIGITAL_INPUT "DIGITAL_INPUT"
#define TASK_DIGITAL_OUTPUT "DIGITAL_OUTPUT"

// Definizione delle costanti
#define ACC_GRAV 9.806
#define PI 3.141593
#define C_SOUND 343.4
#define ON 1
#define OFF 0

// Definizione della distanza tra i finecorsa elettrici
#define MAX_CORSO 0.910

```

```

// Definizione delle risoluzioni degli encoder
#define RISOLUZIONE_ENC_RIS 1024
#define RISOLUZIONE_ENC_COMP 1000

// Definizione della temperatura ambiente
#define T_AMBIENTE 23.0;          // [°C]

// Definizione etichette per sensori di pressione e temperatura
#define P01 0                    // Indice sensore pressione P01
#define P02 1                    // Indice sensore pressione P02
#define P03 2                    // Indice sensore pressione P03
#define P04 3                    // Indice sensore pressione P04
#define T01 0                    // Indice termocoppia T01
#define T02 1                    // Indice termocoppia T02

// Definizione delle sensibilità dei trasduttori di pressione (Uv/Pa)
#define SENS_P 316.228          // [uV/Pa]

// Definizione posizione array per variabili encoder
#define ENC_RIS 0
#define ENC_COMP 1

// Maschere input/output digitali
#define DriveEnable 0x1 //digital channel 0
#define TTL5Volt 0x2 //digital channel 1
#define StartTrigger 0x4 //digital channel 2
#define PowerEnable 0x8 //digital channel 3
#define EMERG_IN 4 //digital channel 4
#define FC_SX 5 //digital channel 5
#define FC_DX 6 //digital channel 6

// Errori di FAULT macchina
#define EMERG_error -1
#define FC_error -2
#define VELMAX_error -3
#define POSMAX_error -4

// Definizione della variabile tempo assoluto
#define Time tickCount*Tc

// Definizione variabile booleana
typedef int bool;

// Struttura faults macchina
struct FaultMacchina
{
    int checkEMERG_error;
    int checkFC_error;
    int checkVELMAX_error;

```

```
    int checkPOSMAX_error;
};

// Struttura input digitali
struct digitalStatus
{
    int Em_IN;           // Digital channel 0
    int FC_Sx;          // Digital channel 1
    int FC_Dx;          // Digital channel 2
};

// Struttura filtro II ordine
struct filtroIIord
{
    double k1;
    double k2;
    double alpha;
    double beta;
    double gamma;
    double K;
    double a0;
    double a1;

    double ingresso [3];
    double uscita [3];
};

// Struttura filtro continua
#define MAX_PT.FILTRO.CONTINUA 3001
struct filtroContinua
{
    int idx;
    int npt;
    double buffer [MAX_PT.FILTRO.CONTINUA];
};

// Stati macchina
typedef enum
{
    sM_InizializzaDati=0,
    sM_OkAvvio,
    sM_WaitOkHw,
    sM_OkZero,
    sM_initZero,
    sM_Zero,
    sM_initPreReady,
    sM_PreReady,
    sM_initReady,
    sM_Ready,
};
```

```
sM_initContmoto ,
sM_ContMoto ,
sM_initFaultHW ,
sM_FaultHW ,
sM_initFaultHW_FC ,
sM_FaultHW_FC ,
sM_FaultSW ,
sM_TwHalt ,
sM_Halt ,
} TypePasso;

// Output delle funzioni
typedef enum
{
    FaseInit=0,           // Inizializzazione variabili stato macchina
    FaseExec,            // Esecuzione in corso funzione stato macchina
    FaseEnd,              // Termine esecuzione funzione stato macchina
    FaseInError,         // Errore durante esecuzione stato macchina
} TipoEsecuzioneFase;

// Struttura Setpoint
typedef struct
{
    double posizione;
    double velocita;
    double accelerazione;
    double corrente;
} SPasse_data;

// Struttura contenente i dati della legge di moto
struct DatiLegge
{
    int tipoLegge;
    double t0_ass;
    double pos_init;
    double vel_init;
    double acc_init;
    double pos_end;
    double h;
    double ta;
    double index;

    // Memoria dei setpoint
    SPasse_data SP_new, SP_old;
};
```

```

// Enum delle diverse leggi di moto
typedef enum
{
    legge345=0,
    legge357 ,
    legge369 ,
    legge4567 ,
    leggeTagl_unTerzo ,
    leggeTagl_unDecimo ,
    legge_seno ,
    legge_decelCost ,
} TipoLeggeMoto;

// Struttura per legge di moto sinusoidale
typedef struct
{
    double spPos_old;
    double spVel_old;
    double t_init;
} setPointSeno;

// Struttura contenente lo stato dell'asse
typedef struct
{
    double ENC; // Posizione angolare misurata da encoder [rad]
    double sigEncZERO; // Posizione angolare di azzeramento encoder [rad]
    double sigEnc; // Posizione angolare filtrata [rad]
    double sigEnc0; // Posizione angolare con azzeramento [rad]
    double posLin; // Posizione lineare [m]
    double posLin0; // Posizione lineare con azzeramento [m]
    double velLin; // Velocità lineare [m/s]
    double velLin_filt; // Velocità filtrata per check FAULT max. vel. [m/s]
    double omega; // Velocità angolare [rad/s]
} StatoAsse_data;

// Struttura contenente lo stato del compressore
typedef struct
{
    double ENC; // Posizione angolare misurata da encoder [rad]
    double sigEnc; // Posizione angolare filtrata [rad]
    double omega; // Velocità angolare [rad/s]
} StatoComp_data;

// Struttura contenente le variabili fisiche del sistema (pressioni e temperature)
typedef struct
{
    double pressP [4]; // Pressione [bar]
    double tempT [2]; // Temperatura [°C]
} StatoFisica_data;

```

```

// Struttura dati ASSE
typedef struct
{
    double passo;           // Passo della vite a ricircolo [m]
    double tau;             // Rapporto di trasmissione [m/rad]
    double tauRid;         // Rapporto di riduzione del riduttore
    double Kt;              // Costante di coppia del motore [Nm/A]
    double I_max;          // Corrente massima drive
    double corsa;          // Corsa asse (distanza tra FC elettrici) [m]
    double limPosMin;      // Posizione minima ammessa (limite sw) [m]
    double limPosMax;      // Posizione massima ammessa (limite sw) [m]
    double I_max_can;      // Corrente massima di comando sull'asse [A]
    double vel_max;        // Velocità massima ammessa sull'asse [m/s]
    unsigned int impulsi_enc; // Numero impulsi a giro encoder
} Asse_data;

// Struttura dati ASSE
typedef struct
{
    double d_collo;
    double l_collo;
    double d;

    double l_eq;
    double d_collo_su_d;

    double l_correzione;
} Helmholtz_data;

// Struttura per REGOLATORE POSIZIONE
struct REG_pos_param
{
    double a;               // Parametro regolatore
    double b;               // Parametro regolatore
    double c;               // Parametro regolatore
    double Kp;              // Guadagno proporzionale
    double Kd;              // Guadagno derivativo
    double Ki;              // Guadagno integrale
    double Ti;              // Costante di tempo integrale
    double Td;              // Costante di tempo derivativa
    double N;               // Posizione polo filtro azione derivativa
    double uMAX;            // Limite saturazione azione FB

    // Flag per la scelta di utilizzo delle azioni di FF
    int usaFB_Pos;          // Variabile booleana ON/OFF azione di Feed Back

    // Dati da dampare
    double comp_FB, comp_FF_Attriti, comp_FF_Dyn;

```



```

// Memorie
double ingresso [2]; // Memoria per l'errore in ingresso al regolatore
double uP, uD, uI; // Azioni di controllo PID
double uscita; // Somma delle singole azioni di controllo [A]
};

// Struttura dati REGOLATORE
typedef struct
{
    struct REG_pos_param REG_pos; // Regolatore: controllo del moto

    enum
    {
        no_reg=0,
        reg_pos,
    } tipoReg; // Tipo di regolatore da utilizzare

    Asse_data *dati_Asse; // Puntatore ai dati dell'asse
    StatoAsse_data *stato_Asse; // Puntatore allo stato dell'asse
    SPAsse_data SP_Asse; // SetPoint da fornire all'asse

    double I_comando; // Uscita in corrente da fornire al motore [A]
} Regolatore_data;

// Enum per esecuzione funzione calcola SP nel controllo del moto
typedef enum
{
    p_initSPctrl=0,
    p_waitUtenteSPctrl,
    p_initMuoviAutoCtrl,
    p_execMuoviAutoCtrl,
    p_initMuoviManualeCtrl,
    p_execMuoviManualeCtrl,
    p_initFermaSystCtrl,
    p_FermaSystCtrl,
    p_restartSPctrl,
    p_waitRestartMuoviAutoCtrl,
    p_restartMuoviAutoCtrl,
    p_endSPctrl,
} passi_exec_SPrisonatore;

// Struttura dati parametri moto per posizionamento risonatore
typedef struct
{
    // Parametri per la generazione del setpoint
    double posNow;
    double posEnd;
    double t_alzata;

```

```
TipoLeggeMoto legge;  
  
double posInizialeMoto; // Posizione di partenza per effettuare l'alzata  
struct DatiLegge dataLegge; // Struttura contenente i dati della legge  
SPAsse_data SP;  
  
passi_exec_SPrisonatore passo_calcSPris;  
} datiMotoCtrlRisonatore;  
  
// Definizione tipo movimentazione asse risonatore  
typedef enum  
{  
    MuoviManuale=0,  
    MuoviAutomatico,  
} enumTypeMoto;
```

# BIBLIOGRAFIA

- [1] AA. VV., “SOUND PRESSURE LEVEL”  
*sito web*, [http://en.wikipedia.org/wiki/Sound\\_pressure](http://en.wikipedia.org/wiki/Sound_pressure)
- [2] C. Drioli, N. Orio, “ELEMENTI DI ACUSTICA E PSICOACUSTICA”,  
*Ampliamento delle dispense del Prof. G. B. Debiassi*, *sito web*:  
<http://www.dei.unipd.it/~musica/IM/>, 1999
- [3] D. Burr, “UDITO: FISIOLOGIA E PSICOACUSTICA”,  
*dispense del corso di Psicologia Generale e Sperimentale*, *sito web*:  
<http://www.pisavisionlab.org/teaching/burr/>, 2008
- [4] I. L. Vér, L. L. Beranek, “NOISE AND VIBRATION CONTROL ENGINEERING”,  
*Wiley & Sons*, *cap. 1 e 4*, 2006
- [5] S. Salsa, “EQUAZIONI A DERIVATE PARZIALI: METODI, MODELLI E APPLICAZIONI”,  
*Springer*, *pag. 173*, 2004
- [6] P. A. Nelson, S. J. Elliot, “ACTIVE CONTROL OF SOUND”,  
*Academic Press*, *pag. 116-160*, 1995
- [7] Trinder, M. C. Nelson, “ACTIVE NOISE CONTROL IN FINITE LENGTH DUCT”,  
*Journal of Sound and Vibration*, 1983
- [8] J. Tichy, Warnaka, “A STUDY OF ACTIVE CONTROL OF NOISE IN DUCTS”,  
*Journal of Vibration, Acoustics, Stress and Reliability in Design*, 1984
- [9] J. Kazakia, “A STUDY OF ACTIVE ATTENUATION OF BROADBAND NOISE”,  
*Journal of Vibration, Acoustics, Stress and Reliability in Design*, 1986
- [10] L. E. Kinsler, “FOUNDAMENTALS OF ACOUSTICS”,  
*John Wiley, New York*, 1982

- [11] M. Berengier, A. Roure, "BROAD-BAND ACTIVE SOUND ABSORPTION IN A DUCT CARRYING UNIFORMLY FLOWING FLUID", *Journal of Sound and Vibration*, 1980
- [12] I. L. Ver, L. L. Beranek, "NOISE AND VIBRATION CONTROL ENGINEERING", *Wiley & Sons*, pag. 30-35, 2006
- [13] AA. VV., "DATASHEET CASSA ACUSTICA FR8", *Visaton*, 2002
- [14] AA. VV., "DATASHEET CASSA ACUSTICA R10S 8 OHM", *Visaton*, 2002
- [15] AA. VV., "THE NEXUS RANGE OF CONDITIONING AMPLIFIERS", *Bruel & Kjaer*, Technical documentation, <http://www.bksv.com>, 1993
- [16] AA. VV., "DATASHEET LM3875", *National Semiconductor*, sito web: <http://www.national.com>, 1999
- [17] A. C. Neve, "INTRODUZIONE DIGITAL SIGNAL PROCESSING", *S & Z*, pag. 40-50, 2002
- [18] A. C. Neve, "INTRODUZIONE DIGITAL SIGNAL PROCESSING", *S & Z*, pag. 71-74, 2002
- [19] A. C. Neve, "INTRODUZIONE DIGITAL SIGNAL PROCESSING", *S & Z*, pag. 103-108, 2002
- [20] AA. VV. "RECIPROCATING COMPRESSORS FOR PETROLEUM, CHEMICAL, AND GAS INDUSTRY SERVICES", *EnergyAPI Standard*, 2007
- [21] M. A. Corbo, C. F. Stearns, "PRACTICAL DESIGN AGAINST PUMP PULSATIONS", *TURBOLAB: Turbomachinery Laboratory, Texas University*, pag. 22-137, 2005
- [22] K.T. Chen, Y.H. Chen, K.Y. Lin, C.C. Weng "THE IMPROVEMENT ON THE TRANSMISSION LOSS OF A DUCT BY ADDING HELMHOLTZ RESONATORS", *Elsevier, Applied Acoustics, Vol. 54, N. 1, pp. 71-82*, 1998
- [23] G. Diana, F. Cheli, "DINAMICA E VIBRAZIONE DEI SISTEMI", *UTET Libreria*, pag. 146-152, 2005
- [24] AA. VV., "FISICA, ONDE MUSICA" sito web, [http://fisicaondemusica.unimore.it/Risuonatori\\_di\\_Helmholtz.html](http://fisicaondemusica.unimore.it/Risuonatori_di_Helmholtz.html)

- [25] AA. VV., “UNI EN ISO 11690-2: ACUSTICA - RACCOMANDAZIONI PRATICHE PER LA PROGETTAZIONE DI AMBIENTI DI LAVORO A BASSO LIVELLO DI RUMORE CONTENENTI MACCHINARIO - PROVVEDIMENTI PER IL CONTROLLO DEL RUMORE” *Norma UNI EN ISO* , 1996
- [26] L. Kela, “RESONANT FREQUENCY OF AN ADJUSTABLE HELMHOLTZ RESONATOR IN A HYDRAULIC SYSTEM”, *Springer-Verlag, Applied Mechanics Vol. 79, pag. 1115*, 2009
- [27] J. Kim, “APPLICATION OF FOUR POLE PARAMETERS FOR GAS PULSATION ANALYSIS OF MULTI-CYLINDER COMPRESSORS WITH SYMMETRICALLY ARRANGED GAS CAVITIES”, *Proceedings of International Compressor Engineering Conference, pag. 1487*, 1992
- [28] J. Kim, W. Soedel, “ANALYTICAL FOUR POLE PARAMETERS FOR GAS FILLED CYLINDRICAL ANNULAR CAVITIES”, *Proceedings of Midwestern Mechanics Conference, pag. 813*, 1987
- [29] J. Kim, W. Soedel, “GENERAL FORMULATION OF FOUR POLE PARAMETERS FOR THREE-DIMENSIONAL CAVITIES”, *Journal of Sound and Vibration*, 1989
- [30] J. Kim, W. Soedel, “FORMULATION OF FOUR POLES OF THREE-DIMENSIONAL ACOUSTICS CAVITIES USING PRESSURE RESPONSE FUNCTIONS WITH SPECIAL ATTENTION TO SOURCE MODELING”, *Proceedings of International Compressor Engineering Conference, pag. 549*, 1998
- [31] W. Soedel, “SOUND AND VIBRATIONS OF POSITIVE DISPLACEMENT COMPRESSORS”, *CRC Press, pag. 195* , 2007
- [32] AA.VV., “HI-WIN LINEAR GUIDEWAY: TECHNICAL INFORMATION”, *Datasheet del costruttore*
- [33] AA.VV., “HI-WIN BALLSCREWS: TECHNICAL INFORMATION”, *Datasheet del costruttore*
- [34] G. Diana, F. Resta, “CONTROLLO DI SISTEMI MECCANICI”, *Polipress, pag. 25-147*, 2007

- [35] G. Magnani, G. Ferretti, P. Rocco, "TECNOLOGIE DEI SISTEMI DI CONTROLLO", *McGraw-Hill*, pag. 191-243, 2007