

Università degli studi di Bergamo

Dipartimento di Ingegneria

Scuola di dottorato in Meccatronica, Informazione, Tecnologie

Innovative e Metodi Matematici



**DESIGN AND IMPLEMENTATION
OF GUIDANCE CONTROL SYSTEM
FOR AUTONOMOUS LIGHT ELECTRIC VEHICLES**

Doctoral dissertation of
Matteo Giacomo MADASCHI

Advisor: Prof. **Fabio PREVIDI**
Prof. **Sergio M. SAVARESI**

2013 – XXV edition

Index

Chapter 1	7
1.1 Thesis structure	11
Chapter 2	13
2.1 A survey of AGV	13
2.1.1 Navigation	14
2.1.2 Path decision	17
2.1.3 Steering system	18
2.1.4 Traffic congestion control	19
2.2 Concluding remarks	20
Chapter 3	21
3.1 Hardware architecture	23
3.1.1 Driving group	23
3.1.2 Marker board	24
3.1.3 Guidance board	25
3.1.4 Multipurpose board	27
3.1.5 Wi-Fi board	29
3.1.6 Input-Output board	31
3.1.7 Touch panel	32
3.1.8 Optical sensor	34
3.2 Signals routing	35
3.3 Software architecture	38

3.3.1 Marker board	38
3.3.2 Guidance board.....	40
3.3.3 Multipurpose board	42
3.3.4 Wi-Fi board	44
3.3.5 Input Output board	45
3.3.6 Touch panel	49
3.4 Concluding remarks.....	51
Chapter 4.....	53
4.1 The AGV dynamic model.....	53
4.2 Concluding remarks.....	61
Chapter 5.....	63
5.1 A parameter space approach for the description of H_∞ region	64
5.2 Vehicle motion control architecture	68
5.3 Parameter analysis for controller design	69
5.3.1 Sampling for stability domain for polynomials.....	70
5.3.2 Robust stability domain.....	71
5.3.3 Limitations on control action.....	74
5.3.4 Controlled system speed performance requirements.....	75
5.3.5 H_∞ performance evaluation.....	76
5.3.6 Dependence on the mass value of the control current limitation and the transient time speed	77
5.4 PI discretization	79
5.5 Concluding remarks.....	80
Chapter 6.....	83
6.1 Open-end windings brushless motor drive	83
6.2 Basis of current control	87
6.3 SVM standard.....	90
6.4 SVM and current loop for brushless motors with open-end windings	101
6.5 Experimental results	103

6.6 Concluding remarks	107
Chapter 7	109
7.1 Description of the test	109
7.2 Performances analysis	110
7.3 Concluding remarks	116
Chapter 8	117
8.1 Future works	119
Bibliography	121

Chapter 1

Introduction

Nowadays Automated Guided Vehicles (AGVs) are used in many production environments and warehouses such as the automotive industry, logistics or container harbors. In general, AGVs are highly expensive and often not flexible, mainly because of their large dimensions and weight. In fact, they are designed to move loads whose weight is not too large in comparison with the vehicle weight. So, they are not used in small or medium size enterprises, where the investment cost could have a high impact on the cost structure of the handled products or the warehouse dimensions are reduced.

In this work, a light AGV developed by Scaglia Indeva S.P.A. (Brembilla, Italy) is presented. Light AGVs are usually designed to move small payloads and their distinctive feature is their high flexibility and re-configurability in the load handling. However, they have usually limited load weight capacity. In fact, large load weight values have a high impact on the guidance performances, even affecting the vehicle mechanical stability and strong speed limitations are usually applied. Moreover, when designed to move loads in a large range of weight values, from small light plastic objects to heavy metal components, the performance are conservatively chosen on the basis of the largest expected payload.

In this work, we will focus our attention to the vehicle guidance control problem and aim to design controllers that guarantees desirable specifications for a wide range of possible load mass values (from 50kg to 1000kg). Hence, we assume that the payload mass is an uncertainty in the model and resort to robust control design methods. Moreover, we'd like not only to find a single suitable controller but to analyze the stability domain in the controller parameter space, so that we can evaluate the effects on the guidance performances produced by changes in the controller parameters, preserving control system stability. In order to obtain such results we will pay special attention to randomized techniques. The variety of randomized algorithms for control can be found in [1]; special techniques for fixed order controller design are described in [2] and [3]; the problem of static output control feedback is discussed in [4]. We prefer this parameter space approach because it allows not only to compute controller parameters but also to compute a discrete set of stabilizing controllers and to obtain a configuration of the admissible domain in the parameter space, see Figure 1.1.

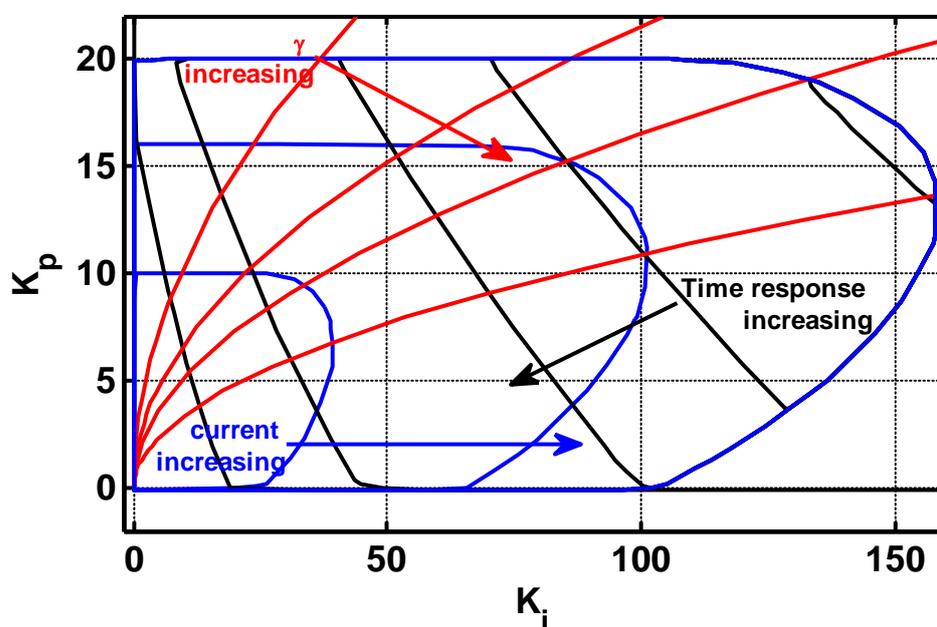


Figure 1.1 Domains of current (blue), time-response (black) and H_∞ norm (red) projected in the parameter space

Moreover, it is demonstrated in [3] how randomized techniques are oriented to deal with basic notions for any engineering characteristics – gain or phase margin, overshoot or other time-response characteristics, robustness margin – as well as mathematical objectives such as H_2 or H_∞ norm.

Specifically, randomized method are applied to the tuning of PI controllers [5], taking into account control action limitations, due to the limited current supply values; system speed specifications, defined by desired values for settling time; H_∞ performances.

The control loops are implemented in Matlab/Simulink and then automatically translated in C code, see Figure 1.2, ready for the compilation with firmware and middleware files and then the complete user software is ready to download on microcontroller.

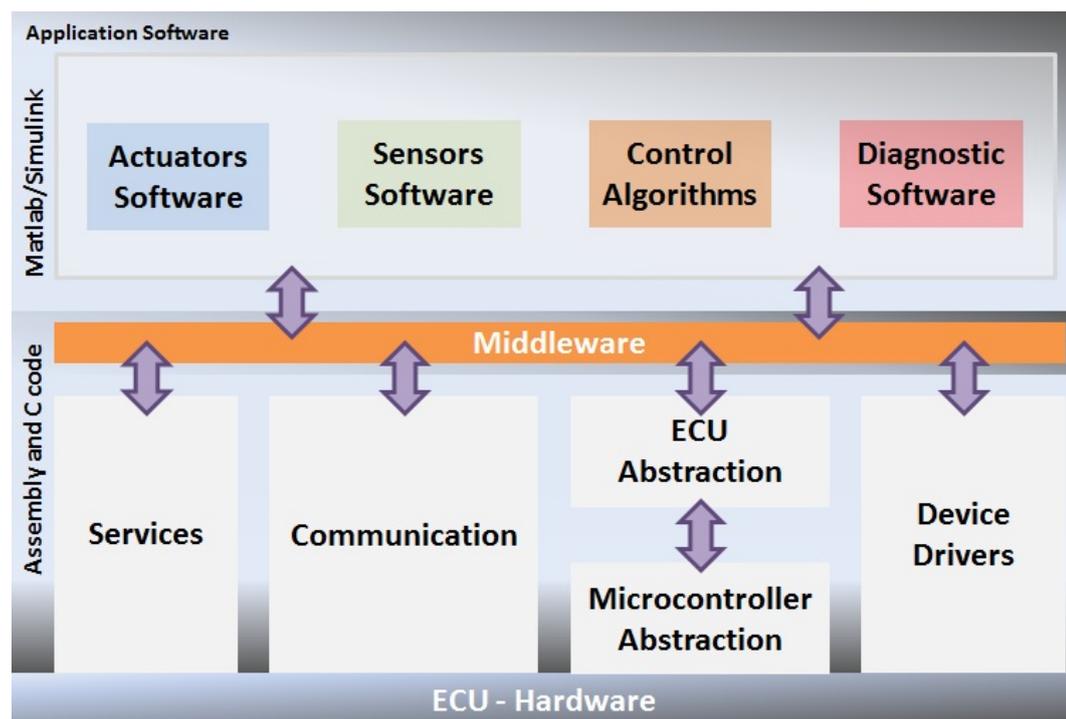


Figure 1.2 Structure of user software that manages the AGV

Two controller have been chosen with different expected performances about time-response and current consumption but both have the H_∞ norm lower than a fixed γ .

The controllers performances will be tested with different types of load (50, 200 and 1000kg) on an AGV used on a commercial vehicle engines assembly line.

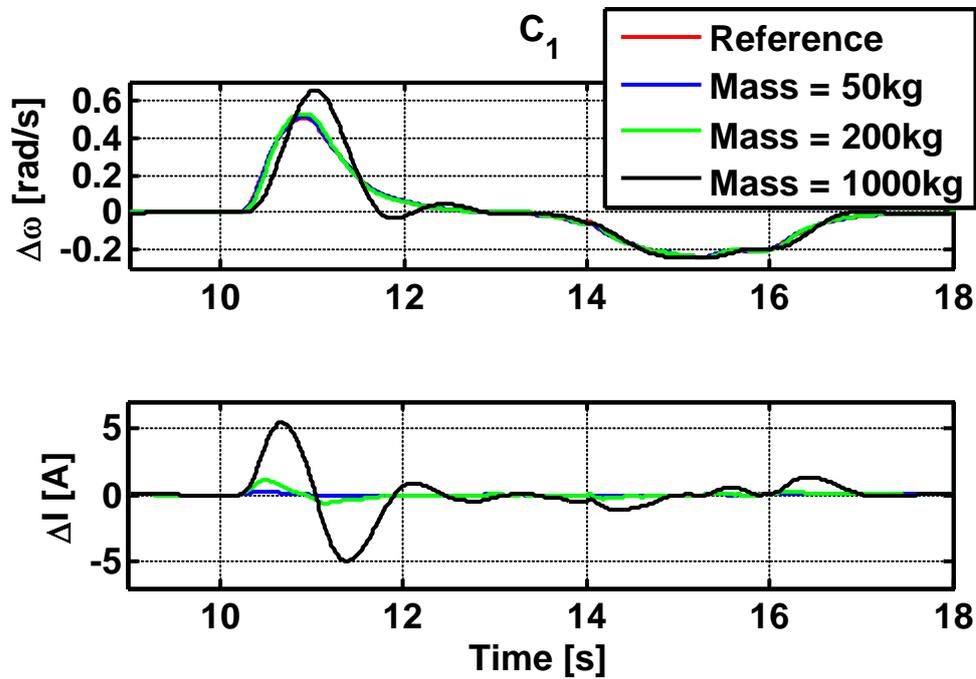


Figure 1.3 Closed-loop steering control $\Delta\omega$ in the experimental test with different loads, C_1 controller

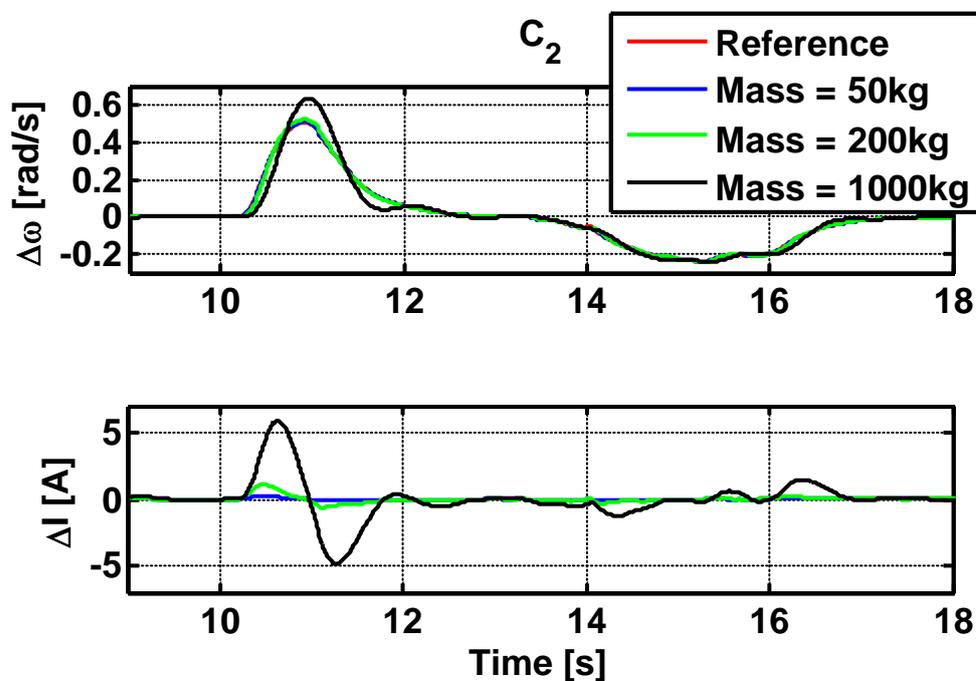


Figure 1.4 Closed-loop steering control $\Delta\omega$ in the experimental test with different loads, C_2 controller

Using the same path for all the tests it will be possible to compare the results for each controller, see Figure 1.3 and Figure 1.4.

1.1 Thesis structure

The topics covered in this thesis are organized in the following section:

CHAPTER 2: AGVs architecture

It describes, through a survey, the features navigation, path decision, steering system and traffic congestion of the most common AGVs.

CHAPTER 3: Prototype description

It describes deeply the prototype used starting from the hardware components (electronic boards and brushless motors) passing through the signals routing and finishing with the description of the software architecture.

CHAPTER 4: AGV mechanical model

It describes the dynamic model of the AGV prototype considering the type of steering system and all the forces acting on the vehicle. The MIMO system obtained is decoupled by means of a static matrix gain.

CHAPTER 5: Design of path control

It describes the techniques used to design the path controller like parameter space approach for the description of H_∞ region, the parameter analysis like limitation on control action and speed performance and finally the discretization of the desired controller.

CHAPTER 6: Current loop of brushless motor with open-end windings

It describes the standard Space Vector Modulation (SVM) for motor with star connection and then introduces the direct and quadrature current control for motor with open-end windings using the standard SVM.

CHAPTER 7: Experimental results

It describes the experimental results obtained from the tests performed on a circuit with a predefined vehicle longitudinal speed and with different loads: no load (50kg), small load (200kg) and full load (1000kg); finally the results are analyzed.

Chapter 2

AGVs architecture

Automatic Guided Vehicle (AGV) is a kind of Wheeled Mobile Robot (WMR) that is used to convey material and supplies in unmanned production workshop. In this chapter will be described an investigation of the principal types of navigation, path decision, steering system and traffic congestion control of the AGVs.

2.1 A survey of AGV

An AGV (Automatic Guidance Vehicle) is an unmanned mobile robot which is able to move materials in autonomous mode thanks to a navigation system. AGVs industrial application are meant to increase the products manufacturing, in fact the automation of materials transportations from/to the warehouse and/or through the production plant can guarantee the reduction of the operating costs and the minimization of the loss of time due to transportation speed.

2.1.1 Navigation

As the acronym AGV states, AGVs are autonomous vehicles and the system which is dedicated to realize their autonomy in the movements through a production plant is the Navigation System.

The AGVs navigation system can be implemented in four different ways:

- *Guide tape*: many AGVs, also called Automatic Guided Carts, use a tape as guide path. The tape can be of two types: magnetic or colored. An advantage in the use of the tape is that it can be easily removed and replaced if the path has to be changed. A disadvantage of this type of navigation is that intersection between two or more paths can be damaged or become dirty because of high traffic over the tape. In order to reduce this kind of damages the magnetic tape can be embedded in the floor, but, doing this, the advantage of the easy replacement is lost. The guide tape is considered a “passive” system since it does not require to be energized.

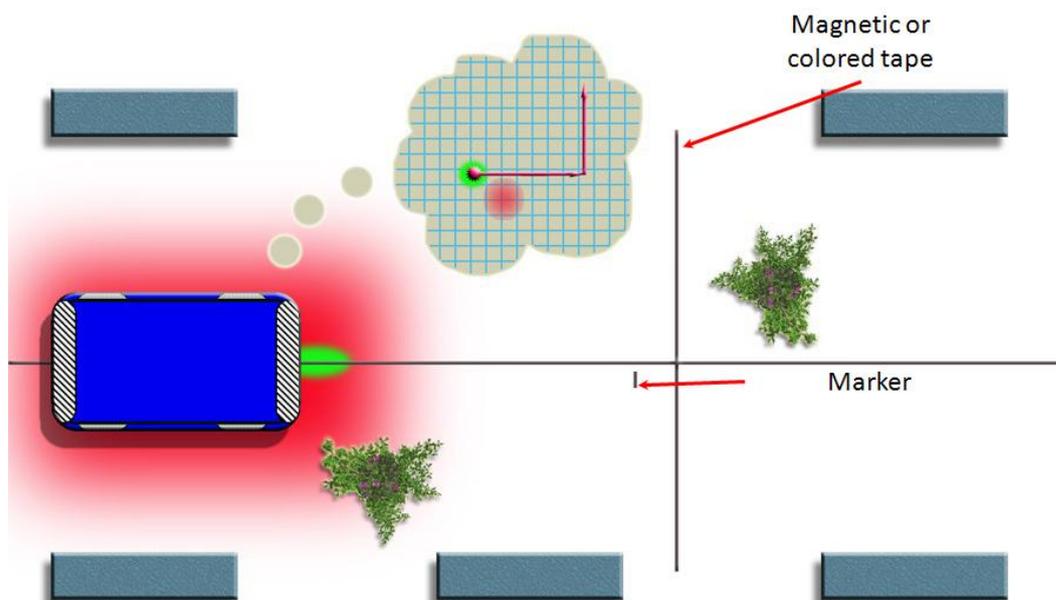


Figure 2.1 Example of guide tape AGV operation

- *Wired*: a slot is dug-out in the ground and a wire is positioned inside it, Figure 2.2. A sensor positioned on the AGV detects the radio frequency transmitted from the wire. This type of navigation has two big disadvantages: it can't be easily removed if a re-routing is needed and there is the possibility of a wire breaking. It is considered an "active" system since it requires to be energized.
- *Laser*: a laser transmitter and receiver are mounted on the AGV on a rotating turret, while retro-reflective tape is positioned on walls and machines, Figure 2.3. The laser is emitted and received so that the angle and the distance from the reflectors are calculated and saved in the AGV. The AGV stores in an internal memory the position of all reflectors and, calculating the error between the expected position and the measurements, is able to correct its position. This type of navigation has the advantage that can be easily removed and replaced, but the complexity of the guidance algorithm increases. It is considered a "passive" system since it doesn't require to be energized.
- *Inertial*: AGVs use an inertial guidance to accurately maintain the intended path and speed and to perform required tasks. Transponders are embedded in the floor of the production site. The AGV uses these transponders to verify that the vehicle is in the correct location, Figure 2.4. A gyroscope is able to detect the slightest change in the direction of the vehicle and corrects it in order to keep the AGV on its path. The disadvantage of this type of navigation is the positioning of the transponder in the ground, but this type of navigation can operate in any environment and has a long lifespan. It can be considered "passive" or "active" depends on the type of transponder. If the transponder generates signals when questioned the system is considered "active" otherwise "passive".

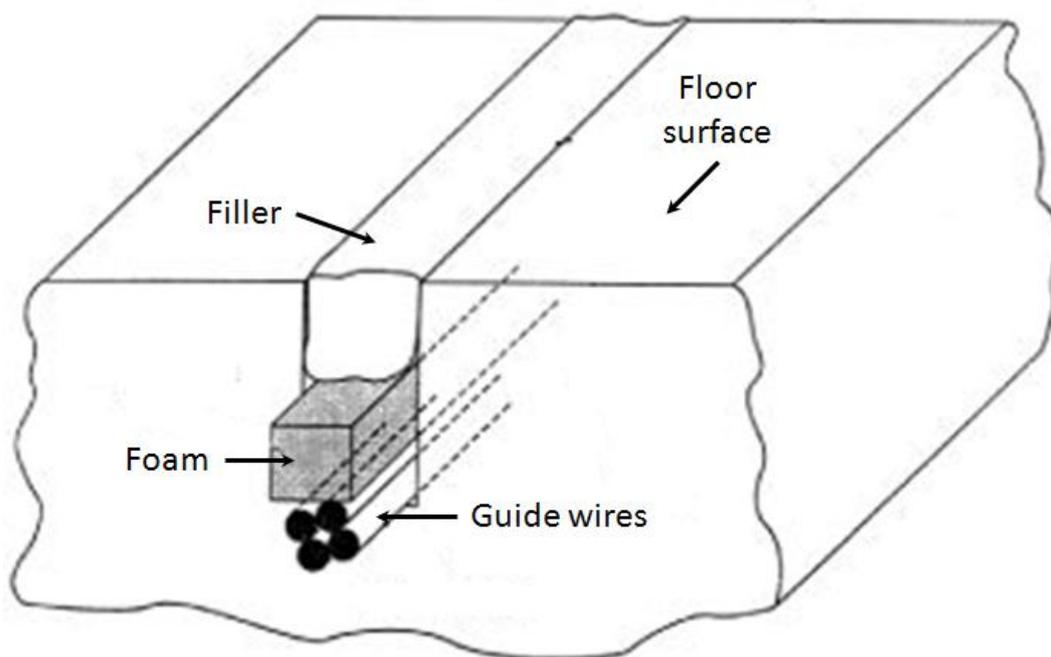


Figure 2.2 Floor section for wired navigation

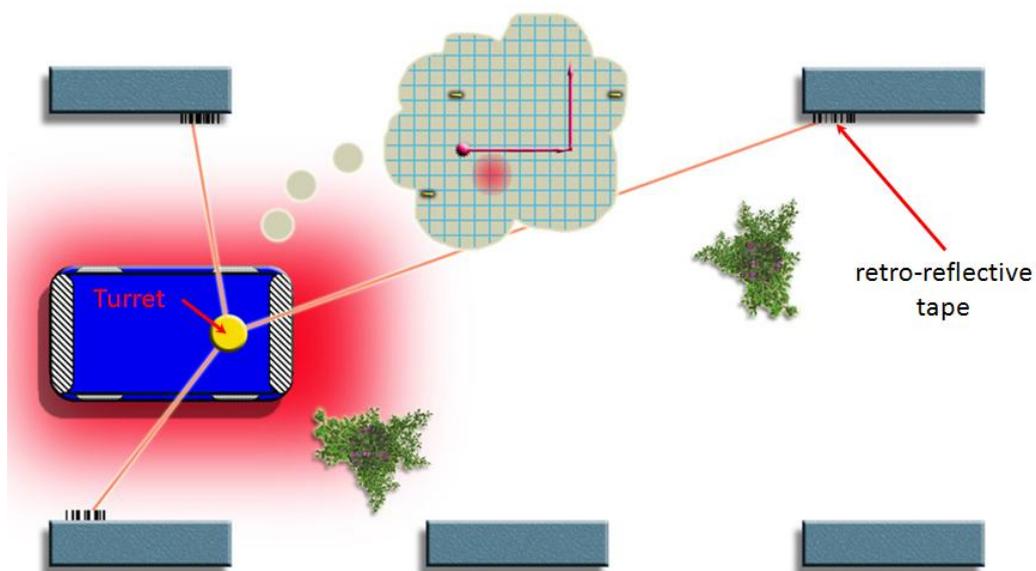


Figure 2.3 Example of laser AGV operation

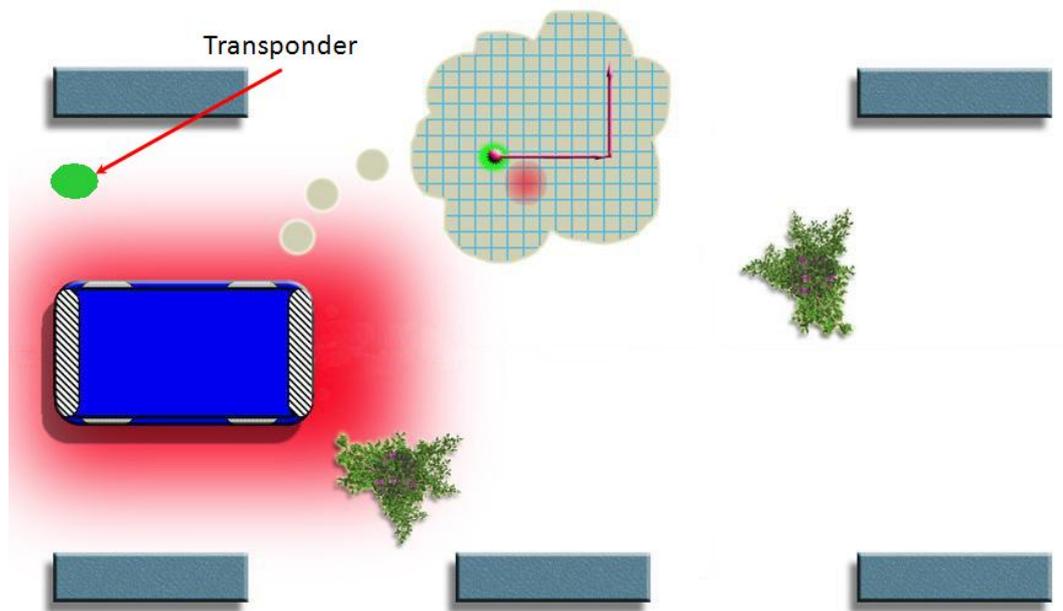


Figure 2.4 Example of inertial AGV operation

2.1.2 Path decision

In order to satisfy the demand of new materials from different stations in a production plant the AGVs have to be able to “choose” the correct path towards the destination. There are different solutions to solve this problem:

- *Frequency select mode:* used by wired AGVs that measure the frequency emitted from the wire drowned in the floor. At each decision point there is a change of the emitted frequency, so when an AGV approaches a decision point it detects different frequencies. The AGV chooses the correct path through a program stored in its memory. This method is not easily expandable and it's very expensive to implement.
- *Path select mode:* the AGV chooses a path among a set of preprogrammed paths. It uses the measures from the on board sensors and compare them to the values provided by programmers. When an AGV approaches a decision point it has to decide which path to follow and the decision is very simple since it already knows its path from its programs. The

implementation of this method can increase the production cost because of the need of specialized programmers to correctly program the AGVs.

- *Magnetic tape mode*: a magnetic tape (magnetic field North) is stuck on the floor and provides the path to the AGV. Adding auxiliaries tapes (magnetic field South), called markers, near to the main navigation tape it is possible to give auxiliaries commands like speed up, slow down, stop, turn left or turn right to the AGV, see Figure 2.1. The markers are read by the AGV on board sensors and their information is used to get the correct path using a program stored in AGV memory. This method is very simple and cheap.

2.1.3 Steering system

The AGVs steering system is very different from the one realized for cars, since AGVs use the differential drive system. This is a very simple system: two motorized wheels provide the power to move the vehicle and changing the speed of each single wheel the AGV can modify its direction as a tank. In fact the right wheel speed (blue arrow) lower than the left wheel speed causes a right curve (red arrow) of the vehicle and vice versa, Figure 2.5; the same speed for each wheel makes the AGV to go straight.

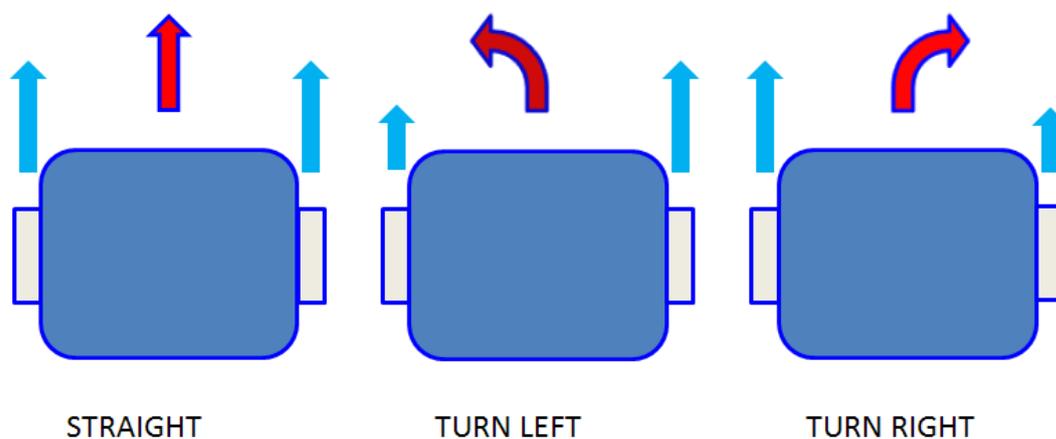


Figure 2.5 Differential drive operation

2.1.4 Traffic congestion control

Usually in a production site there are a lot of AGVs following different paths but using the same bunch of ways, this can potentially lead to frequent collisions. To avoid this problem, the AGV traffic can be controlled implementing different methods:

- *Zone control*: it uses a wireless technology to transmit a signal in a prefixed area like an intersection. Each AGV is equipped by sensing wireless device to receive and transmit back signals to the area transmitter. If the intersection area is empty the incoming AGV can pass through, otherwise it will be stopped until the AGV inside the controlled area will leave. Otherwise each AGV can be equipped with a transmitter and a receiver, so that approaching an intersection it can check if the crossroad is empty: if so, it sends a signal to inform that the intersection is occupied to all the near AGVs; otherwise it will wait for the intersection to be cleared. The first type of zone control has a problem: if one zone goes down the AGVs can collide with any other vehicle. These two methods are easy to install, easy to expand and very simple to control.
- *Forward sensing control*: it uses collision avoidance sensors to detect other AGVs and to avoid collisions. These sensors are: radar, optic and physical contact like bumpers. Radar sensors are able to determine if an object is in front of the AGV and consequently the AGV takes the correct actions to avoid the obstacle. Optic sensors send an infrared signal which gets reflected back and works as radar sensor. Bumper sensors usually open the electrical power circuit when activated. The problem with these sensors is that they can protect the AGV only along one side.
- *Hybrid control*: it is a mix between zone control and forward sensing control. The combination of two controls avoids collision in all situations.

2.2 Concluding remarks

In this chapter are described the various types of features that distinguish AGV.

The characteristics can be summarize in:

Navigation systems

- Guide tape
- Wired
- Laser
- Inertial

Path decision

- Frequency select mode
- Path select mode
- Magnetic tape mode

Steering system

- Differential drive

Traffic congestion control

- Zone control
- Forward sensing control
- Hybrid

Chapter 3

Prototype description

The prototype of AGV herein considered was provided by Scaglia INDEVA SpA. This is a light weight AGV (it can carry or drag up to 750 kg), which has a modular composition thanks to INDEVA Lean System®. This system allows AGV to assume particular shapes or dimensions according to different requirements of different plants, changing only the chassis, see Figure 3.1 and Figure 3.2.

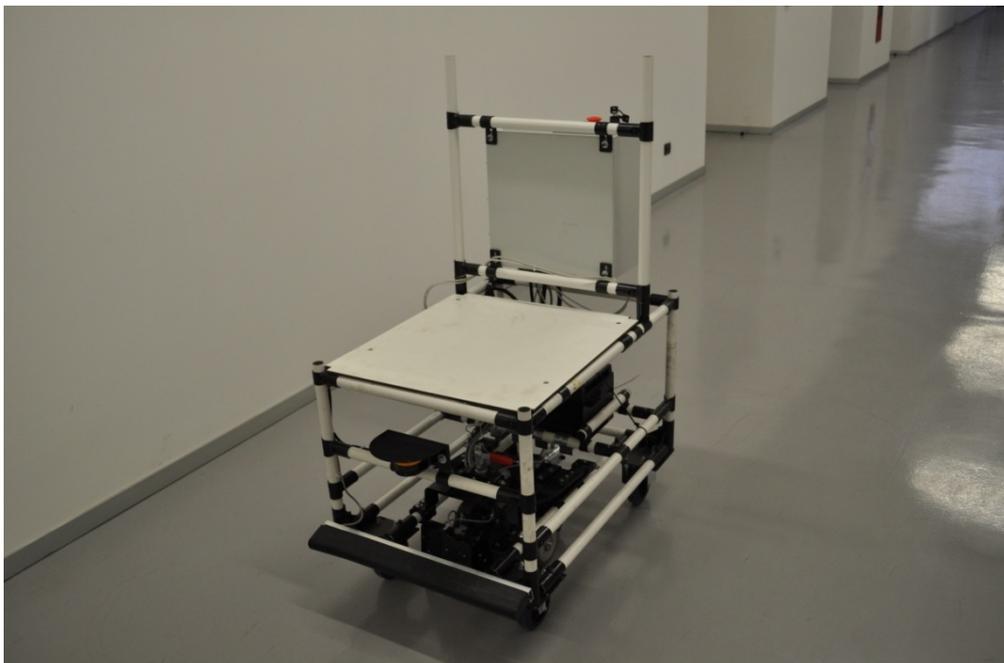


Figure 3.1 AGV prototype

Based on the AGV classification described above, the AGV prototype Navigation system is magnetic guide tape type, its Path decision is based on magnetic tape mode, its Traffic congestion control is based on hybrid control and the Steering system is a differential drive. In fact the AGV is equipped with a bumper, an optical sensor and a wireless board in order to minimize collisions and keep the area, near the vehicle, safe.



Figure 3.2 AGV, the same of Figure 3.1, with another type of chassis

The AGV stands on four wheels: two fixed on the rear of the vehicle and two castor wheels on the front of the vehicle. The driving wheels are placed in the middle of AGV and are pushed to the ground by a spring, see Figure 3.2.

In the next paragraphs the vehicle will be described using the following subdivision in three macro areas: hardware architecture, signal routing and software architecture.

3.1 Hardware architecture

In this paragraph are described all the electronic and mechatronic devices that make up the prototype.

3.1.1 Driving group

The driving group is made up of two brushless motors, two gear reductions, two chains drive, two wheels with solid tires (driving wheels) and a lift motor, see Figure 3.3.

Each brushless motor has a power of 100W and it's possible to change the motor speed rotation thanks to a dedicated driver that receives as input the speed reference from the Input-Output board. The driver provides an alarm in case of failure and it has the following input commands:

- Start or Stop
- Direction of rotation: clockwise or counterclockwise
- Run or Brake
- Alarm Reset

The motor is connected to the wheel by a series of two reductions: a gear reduction and a chain drive.

The lift motor is a brushed motor that raises the entire driving group from ground making the AGV similar to a shopping trolley. This motor is operated by a multipurpose board based on the commands coming from the Input-Output board.

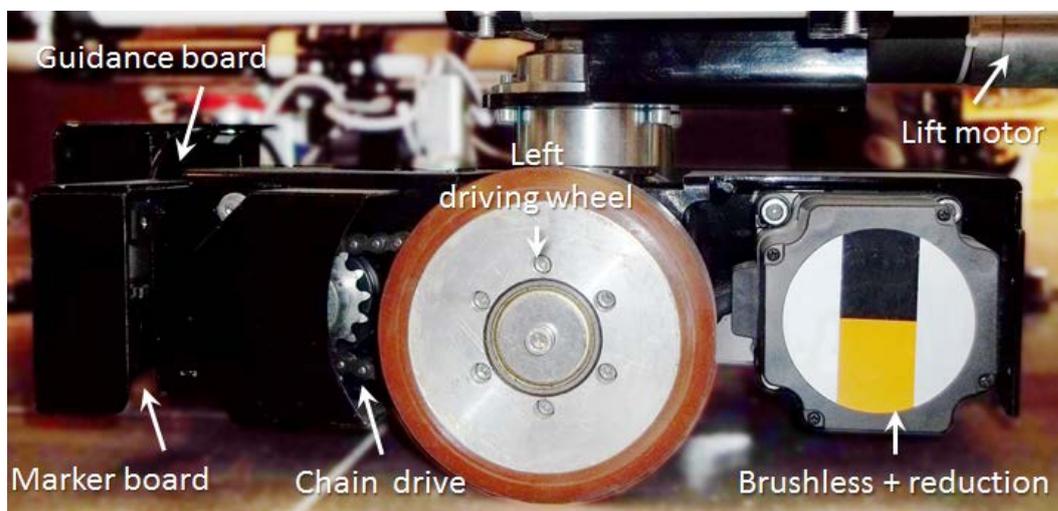


Figure 3.3 Driving group, left side

3.1.2 Marker board

Two marker boards are placed on the front of the driving group, one left and one right, and they measure the presence or the lack of the magnetic marker along the path, see Figure 3.5. To detect the presence of the magnetic tape, the magnetic field measure is compared to a fixed threshold, so it is possible to obtain a digital value: 1 if the marker is present and 0 if the marker is not present. The magnetic field which has to be detected by the marker boards is the South. The Marker board is equipped with a microcontroller, see Figure 3.4, that manages the magnetic field sensor and sends the information about markers presence to the guidance board by a digital line. The microcontroller has to follow a strict procedure in order to get a correct magnetic field measure, for more information see the sensor datasheet.

Technical specifications:

- Microcontroller: Microchip DSPIC33FJ128GP802, architecture 16 bit, CPU speed 40 MIPS, flash memory, 128 KB of program memory and RAM 16,384 Bytes
- Magnetic sensor: Honeywell HMC1021S, 1- and 2- axis magnetic sensor, field range up to ± 6 gauss (earth's field is 0.5 gauss), low cost, small dimensions to reduce board assembly costs.

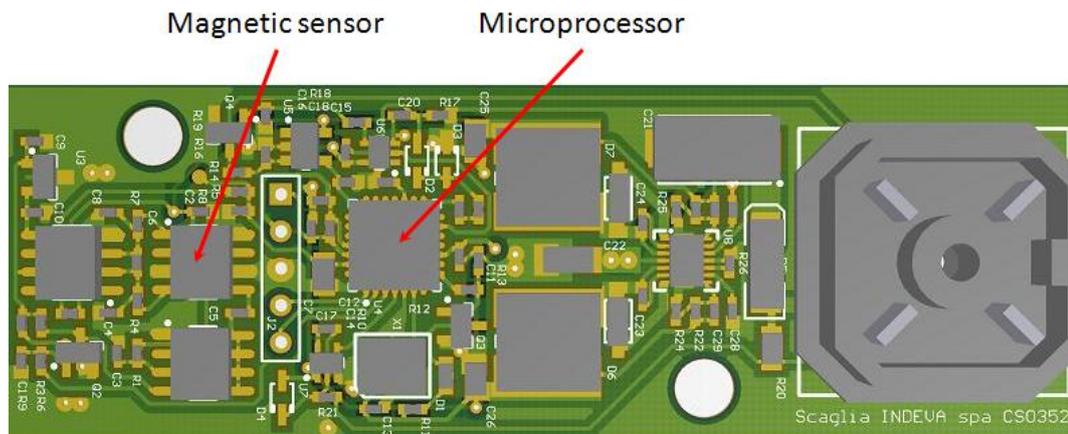


Figure 3.4 Marker sensor rendering

3.1.3 Guidance board

The guidance board is placed in the front center of the driving group, see Figure 3.3 and Figure 3.5, it is equipped with three magnetic field sensors and a microcontroller. The microcontroller reads the internal sensors outputs and the digital lines from marker boards and sends all to the Input-Output board via CAN message with five digital data: 3 for path control and 2 for marker control.

The three sensors are positioned in order to guarantee a perfect reading of the magnetic field over the magnetic tape, Figure 3.6. To calculate the relative positioning between AGV and the magnetic tape, the magnetic field measure is compared to a settable threshold, so it is possible to obtain the reference of the steering speed $\Delta\omega^*$. Exactly like the marker sensor the microcontroller has to follow a strict procedure in order to get a correct magnetic field measure.

The settable threshold is sent via CAN message from Input-Output board based on the position of the driving group:

- Driving group up: the threshold is set to 2.2 Gauss
- Driving group down: the threshold is set to 3.3 Gauss

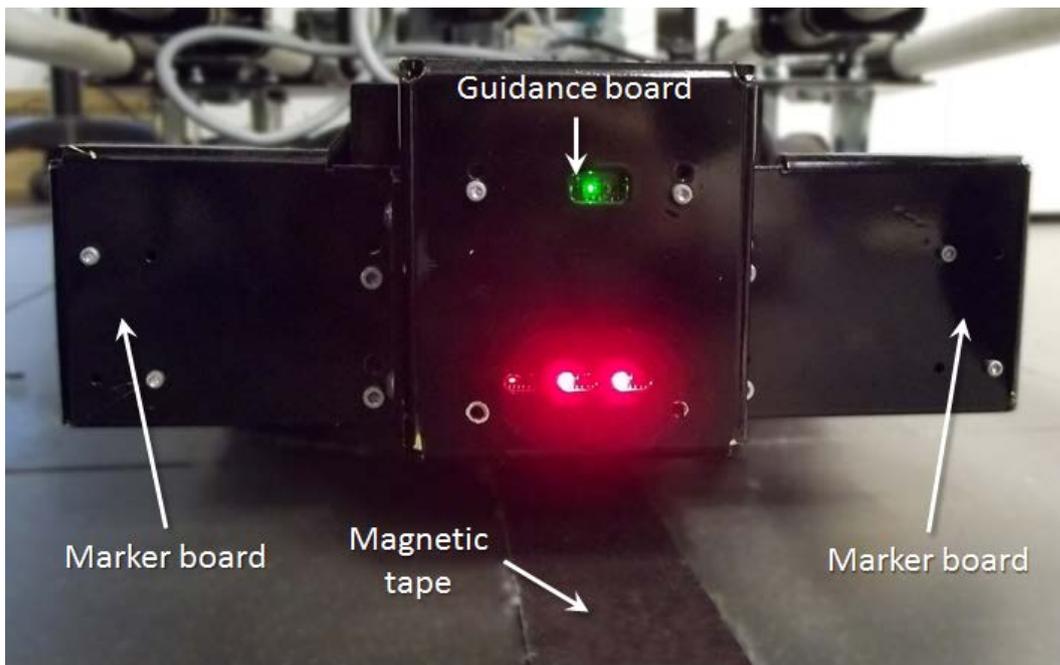


Figure 3.5 Driving group, frontal view

These are the default parameters but it's possible to change them through a specific command from the touch panel.

This option is very useful because there are a lot of type of magnetic tape with different magnetic field intensity, in this way it is possible to calibrate the threshold based on the magnetic field intensity of the magnetic tape type.

Technical specifications:

- Microcontroller: Microchip DSPIC33FJ128GP706, architecture 16 bit, CPU speed 40 MIPS, flash memory, 128 KB of program memory and RAM 16384 Bytes

- Magnetic sensor (3): Honeywell HMC1021S, 1- and 2- axis magnetic sensor, field range up to ± 6 gauss (earth's field is 0.5 gauss), low cost, small dimensions to reduce board assembly costs.
- CAN bus interface

3.1.4 Multipurpose board

The multipurpose board is placed in the electrical panel and it manages the lift motor and an additional motor used to perform an extra motion (usually to up/download goods from the AGV), Figure 3.7.

The multipurpose board has two H-bridges, see Figure 3.17, to command the lift motor and the additional motor both brushed motors.

The multipurpose board receives commands via CAN message from the Input-Output board.

Technical specifications:

- Microcontroller: Microchip DSPIC33FJ128GP802, architecture 16 bit, CPU speed 40 MIPS, flash memory, 128 KB of program memory and RAM 16384 Bytes
- Bridge (2): two IRF7907 dual MOSFET, Very Low $R_{DS(on)}$ at 4.5V V_{GS} , fully characterized avalanche voltage and current and improved body diode reverse recovery.
- CAN bus interface
- Lift motor: Siboni 30PC series, nominal voltage 24V, nominal current 4.1A, nominal speed 3000rpm, nominal power 66W, continuous stall current 0.3Nm
- Additional motor: depends on the type of application. It can be identical to the lift motor for heavy load or it can be a KAG motor M32x30, nominal voltage 24V, nominal current 0.6A, nominal speed 3300rpm, rated power 66W, continuous stall current 2.5Ncm for light load.

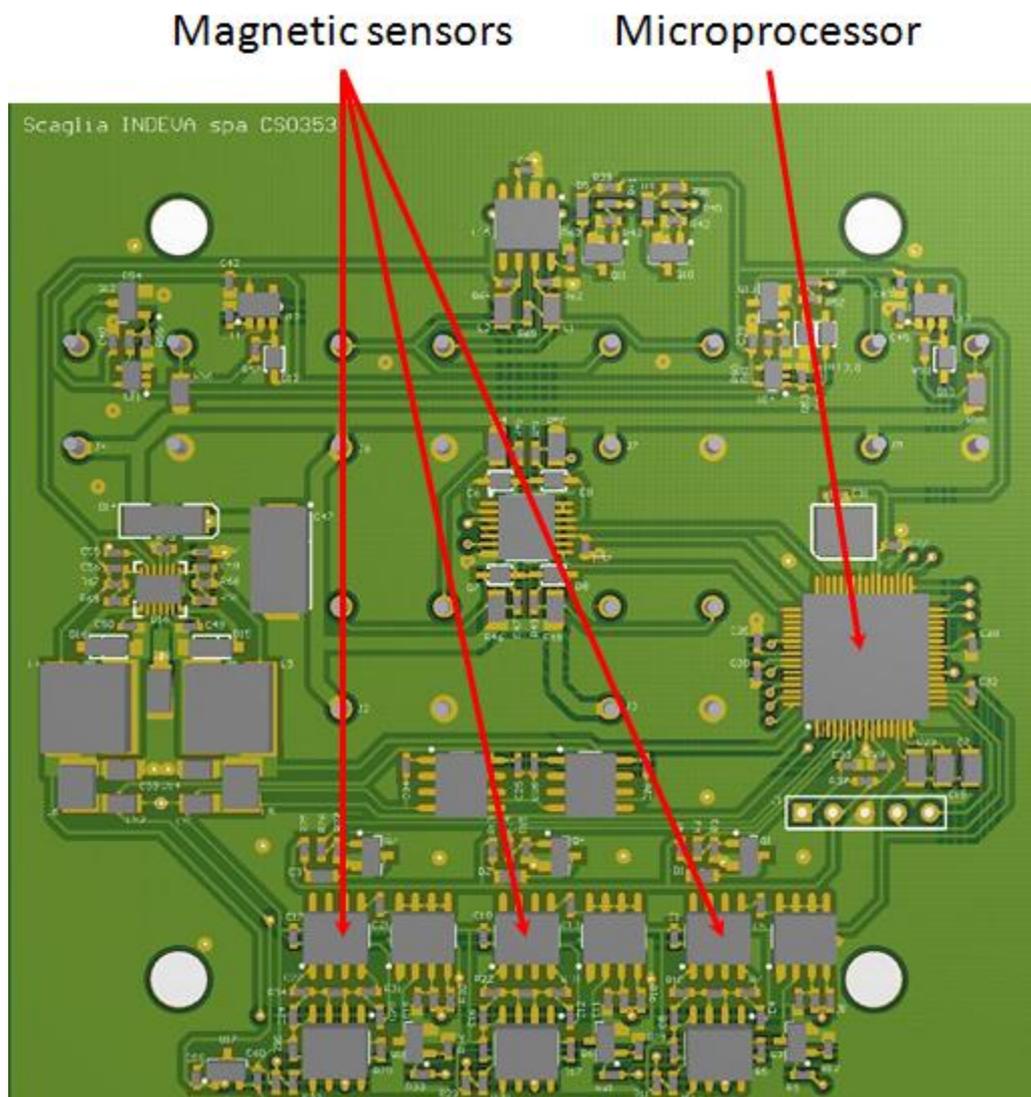


Figure 3.6 Guidance board rendering

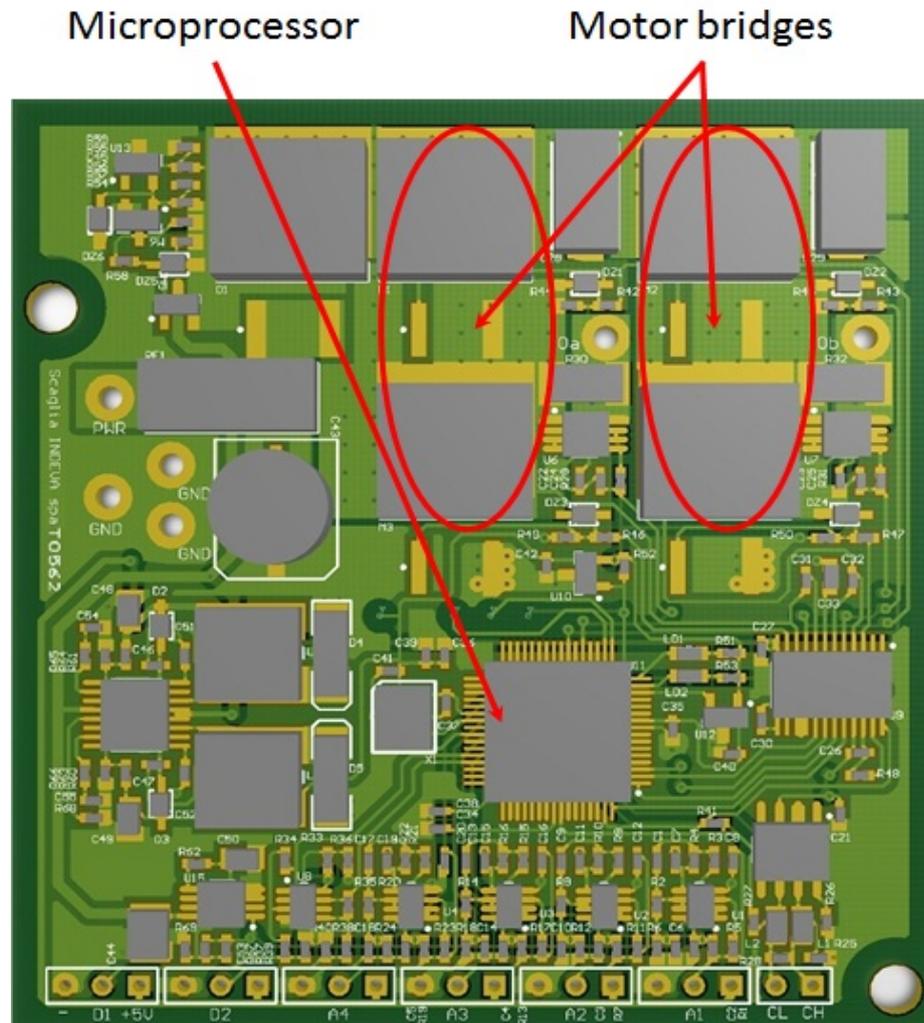


Figure 3.7 Multipurpose board rendering

3.1.5 Wi-Fi board

The Wi-Fi board is placed in the electrical panel and manages all the critical situations between AGVs, like path intersections, and between AGV and smart roller conveyors.

The board is made mainly of a microcontroller and a Wi-Fi module. It communicates with Input-Output board through the CAN bus. To operate correctly, this board receives some information from the Input-Output board like:

- Actual program number: the number of the program that AGV is performing
- Actual step number: the number of the step that AGV is performing
- Generic 1, 2 and 3: program commands with generic orders for the correct execution of the assigned task

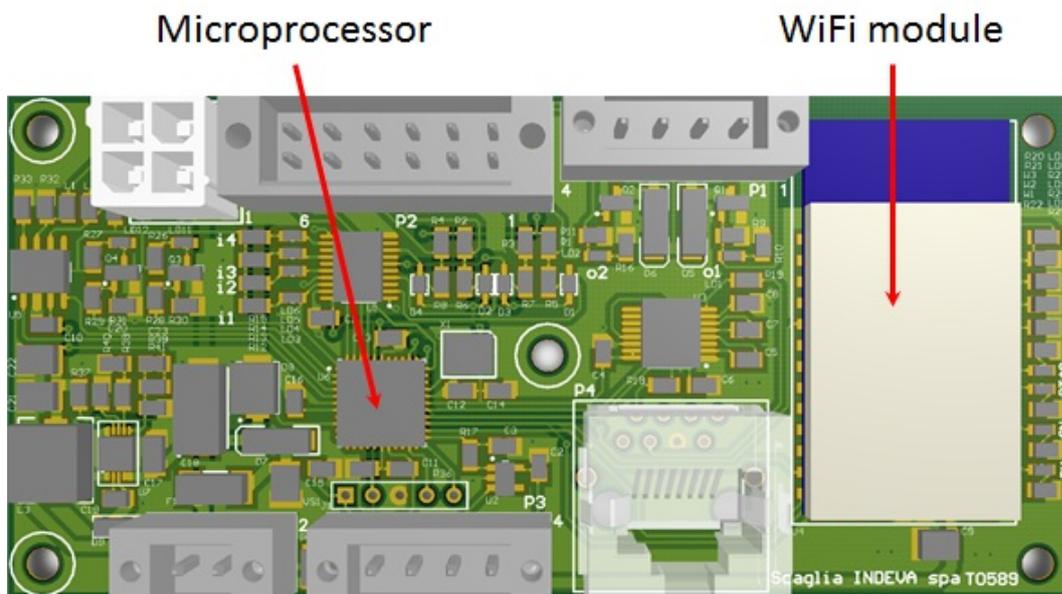


Figure 3.8 Wi-Fi board rendering

Technical specifications:

- Microcontroller: Microchip DSPIC33FJ128GP802, architecture 16 bit, CPU speed 40 MIPS, flash memory, 128 KB of program memory and RAM 16,384 Bytes
- Wi-Fi module: Roving Networks RN-131, ultra-low power design for battery powered application, UART and SPI slave are its hardware interfaces, configuration over serial or wireless interface using simple ASCII commands, secure Wi-Fi authentication schemes (WEP / WPA / WPA2) and full onboard TCP/IP stack (no external drivers required).
- CAN bus interface

3.1.6 Input-Output board

The Input-Output board is placed in the electrical panel and it drives the AGV along the path through the path control manager, moreover it performs all the commands introduced in the programs by operators like turn on/off lamps, indicators and melody unit; in other words this board can be considered the AGV brain, Figure 3.9.

This board elaborates the information coming from CAN bus:

- Guidance board: values of steering speed reference, $\Delta\omega^*$, and marker sensors
- Wi-Fi board: number of program and step to be performed after a command reception from a keypad equipped with a Wi-Fi module; remote start and stop buttons; diagnostic of the presence of CAN bus connection
- Multipurpose board: the state of 8 remote digital inputs and the status of lift motor and additional motor. The status of the motors can be the following: motor is not supplied, motor blocked and supplied (with current limitation before the maximum time for torque control) and the driver has performed the torque control

Technical specifications:

- Microcontroller: Microchip DSPIC33FJ256MC710, architecture 16 bit, CPU speed 40 MIPS, flash memory, 256 KB of program memory and RAM 30706 Bytes
- Input-Output expanders (2): configurable 16 bit I/O expanders via I²C with three hardware address pins to allow up to eight devices on the bus.
- Safety relay (2): close the redundant hardware safety chain
- Safety Inputs (8): 8 analog inputs used to read digital inputs in order to improve the safety and perform AGV diagnostic
- Safety Outputs (8): 8 digital outputs enabled from the microprocessor watchdog, so in case of microcontroller failure the outputs will be disabled

- Serial interface: to communicate with PC
- CAN bus interface: to communicate with all boards
- Modbus interface: to communicate with touch panel
- External memory: CMOS EEPROM with I²C interface, 1024 KB. In this memory are stored all the AGV settings and the AGV programs.

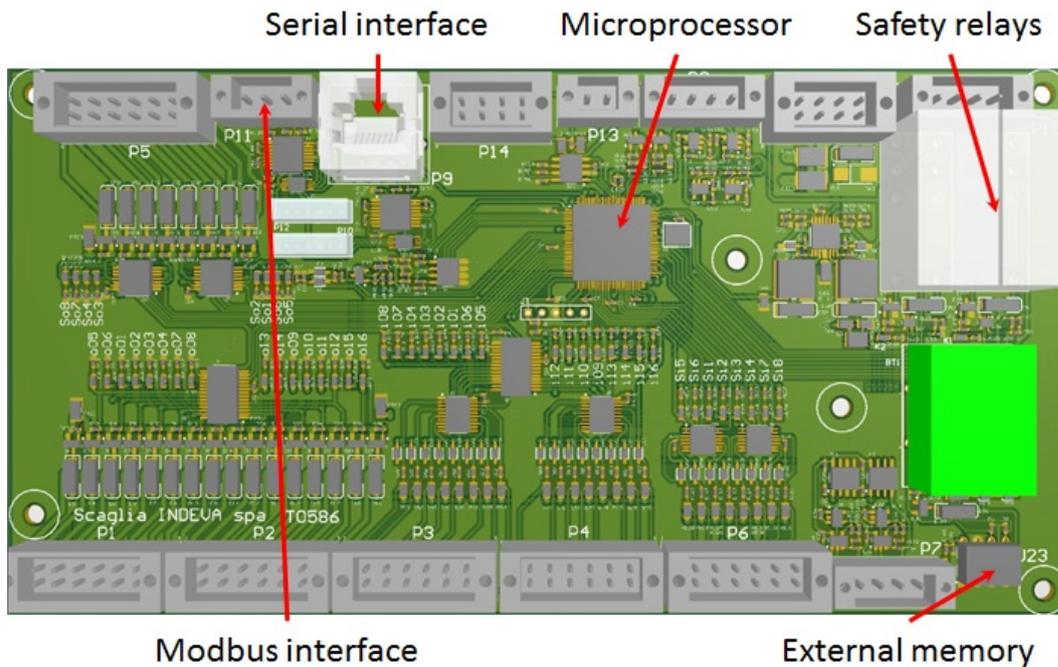


Figure 3.9 Input-Output board rendering

3.1.7 Touch panel

The Human Machine Interface (HMI) is realized through the touch panel AS57 produced by Delta Electronics , Figure 3.10.

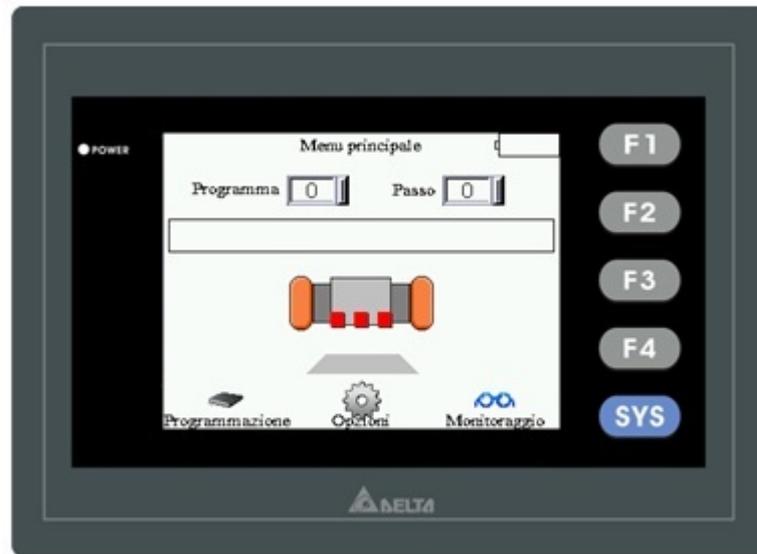


Figure 3.10 Touch panel with home screen

By the touch panel is possible to insert programs with all the commands, to change the settings parameters and to monitor the internal signals. These operations are performed via Modbus communication between the touch panel and the Input-Output board.

The screenshots are developed with the HMI software editor “Screen Editor”, downloadable for free from the Delta Electronics web site. This software is very user friendly and it’s very helpful in creating screenshots using pre-existing images.

Technical specifications:

- Full color display
- Resolution 320x240 pixel
- Size 5.7”
- USB interface for a rapid software download
- Industrial communication interfaces

3.1.8 Optical sensor

The AGV is equipped with an optical sensor which allows to avoid collisions. This sensor monitors the area in the travel direction of AGV and switches its outputs to the OFF state to stop the vehicle (this operation opens the redundant hardware safety chain and the brakes are activated) as soon as there is an object in the protective field.

The sensor is a SICK product, S300 expert, and it's consisting of three components, Figure 3.11:

- the sensor with the opto-electronic acquisition system, the LEDs and the 7-segment display
- the optics cover with the window for the light output
- the system plug with the configuration memory (the system plug contains all the electrical connections)

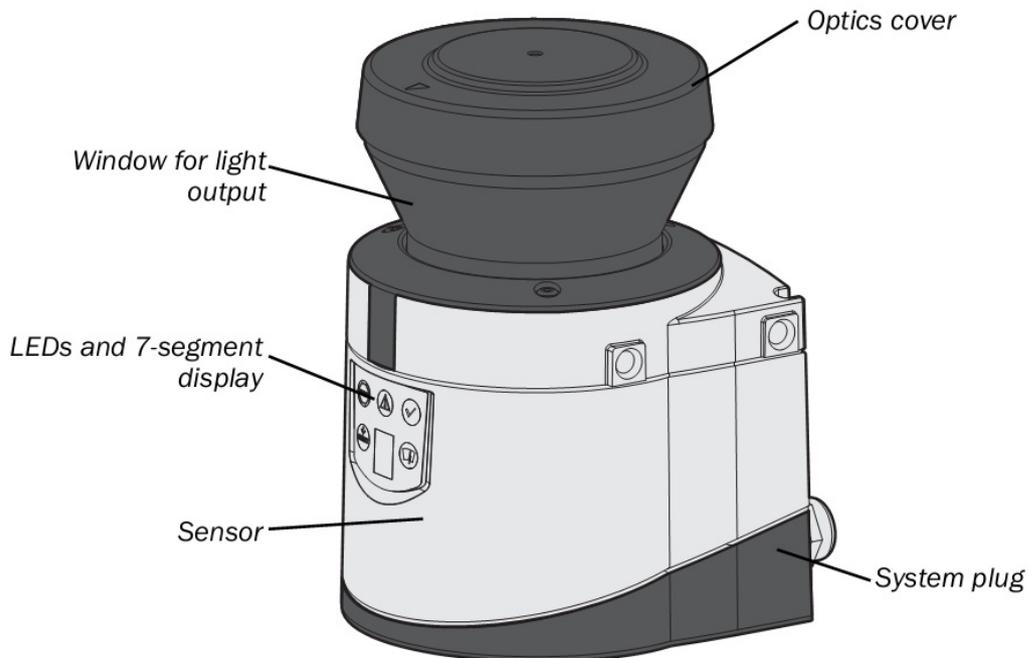


Figure 3.11 Optical sensor: device components

Technical specifications:

- 270° scanning angle
- 8 different contour areas can be monitored
- minimum response time 80 ms
- configuration using PC or notebook with SICK Configuration & Diagnostic Software
- The safety level corresponds to category 3 in accordance with EN ISO 13849-1, type 3 in accordance with CLC/TS 61 496-3 as well as SIL2 in accordance with IEC 61508/EN 61508.

3.2 Signals routing

The signals flow in several direction among all the devices on board the vehicle: the boards, the motors drivers and the touch panel. In Figure 3.12 is possible to see the variety of means of communication: CAN bus, Modbus, air, digital lines and analogical lines.

As can be noticed in Figure 3.12 and in Table 3.1 the Input-Output Board is the brain of the vehicle because all the measurement signals go to this board and all the control signals to the actuators are sent from this board

The CAN bus provides a means of communication among the boards. A lot of signals pass through the bus inside various messages, see Table 3.1. Each message sent from a transmitter on the CAN bus is received from all the devices connected to the bus, but only the receiver who has the correct filter can decode the message and assign the correct value to the signals.

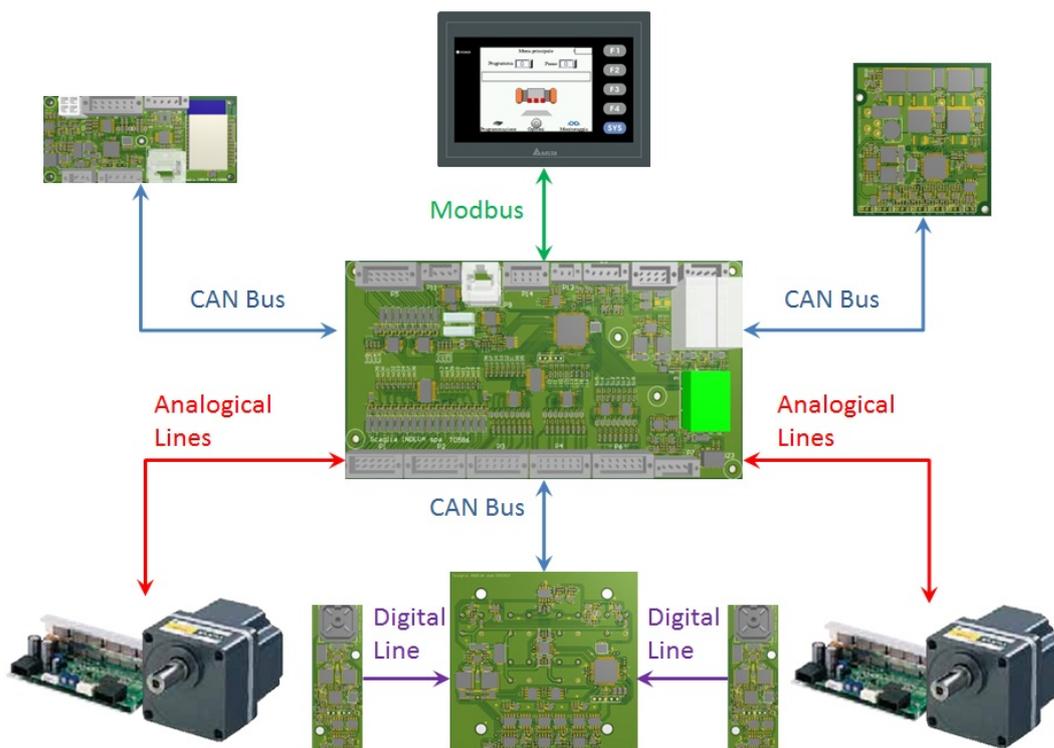


Figure 3.12 AGV signals routing

CAN message	Transmitter	Receiver
Marker and Track Values (MTV)	Guidance Board	Input-Output Board
Change Threshold Guidance (CTG)	Input-Output Board	Guidance Board
Up-Down Lift Motor (UDM)	Input-Output Board	Multipurpose Board
Lift Motor Acknowledgment (ALM)	Multipurpose Board	Input-Output Board
Wi-Fi commands (WFCMD)	Wi-Fi Board	Input-Output Board
Wi-Fi Acknowledgment (WFA)	Input-Output Board	Wi-Fi Board

Table 3.1 Description of CAN messages with Transmitters and Receivers

Inside these messages there are signals useful for the management of the vehicle, see the below tables:

Marker and Track Values (MTV)	Signal name
	Left Marker
	Right Marker
	Steering Speed Reference

Table 3.2 MTV signals

Change Threshold Guidance (CTG)	Signal name
	Threshold value

Table 3.3 CTG signals

Up-Down Lift Motor (UDM)	Signal name
	Lift motor command
	Lift motor current limit
	Lift motor time max
	Additional motor command
	Additional motor current limit
	Additional motor time max

Table 3.4 UDM signals

Lift Motor Acknowledgment (ALM)	Signal name
	Lift motor state
	Additional motor state
	First limit switch
	Second limit switch
	Multi digital inputs

Table 3.5 ALM signals

Wi-Fi commands (WFCMD)	Signal name
	Start/stop button emulation
	New program number
	New step number
	Multi digital inputs

Table 3.6 WFCMD signals

Wi-Fi Acknowledgment (WFA)	Signal name
	Actual program number
	Actual step number
	Generic commands
	Safety chains state

Table 3.7 WFA signals

3.3 Software architecture

The software distribution among the various boards meets the needs of computational and simplicity features.

In fact each board manages its competences perimeter sending out only the information that are useful to the other boards.

In detail, it is possible to obtain a division that reflects the hardware architecture.

3.3.1 Marker board

To the software of this board it's assigned the task of identifying markers. First of all, as described in paragraph 3.1.2, the software has to instruct the magnetic sensor through a strictly procedure to read correctly the magnetic field. The procedure is shown in Figure 3.13 for the circuit showed in Figure 3.14.

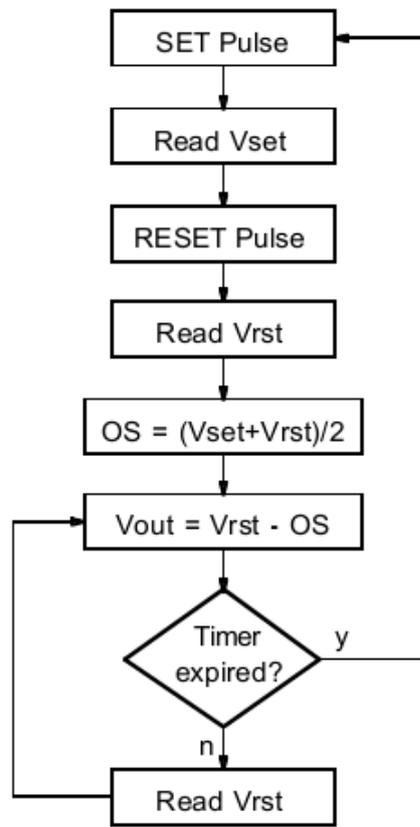


Figure 3.13 Flow chart procedure to prepare magnetic sensor

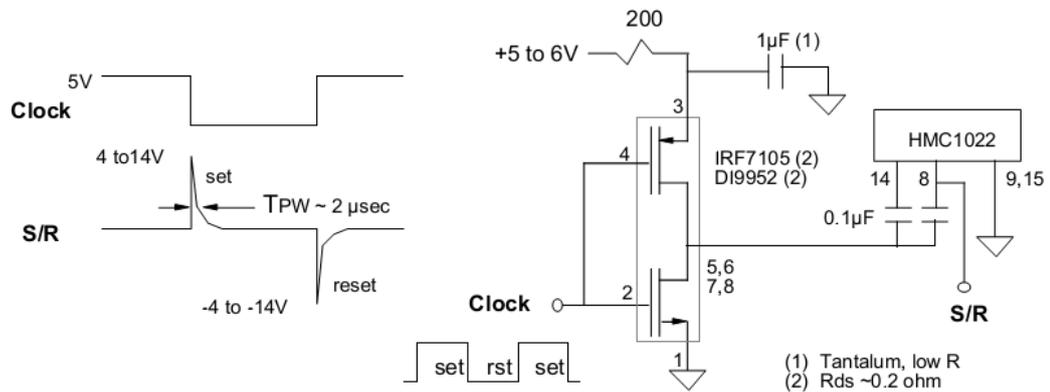


Figure 3.14 Set and Reset Pulse with control clock

The software sends to the sensor a sequence of commands, Set and Reset, that permit to remove temperature drift, non-linearity and errors cross-axis effects aligning the sensor magnetic dipoles. After that it is possible to calculate the offset voltage which has to be subtracted to the voltage measured. Then, thanks to the

sensitivity value provided by datasheet (1.0mV/V/G), it's possible to calculate exactly the magnetic field intensity (South). The presence of marker tape is determined through a hysteresis comparator, see Figure 3.15: if the magnetic field intensity is smaller than -3.3 Gauss continuously for 20ms then the digital output changes its value (e.g. from 0 to 1). To detect the lack of marker tape the procedure is the same but the value of magnetic field intensity is 3.0 Gauss. The procedure is evaluated every 1ms.

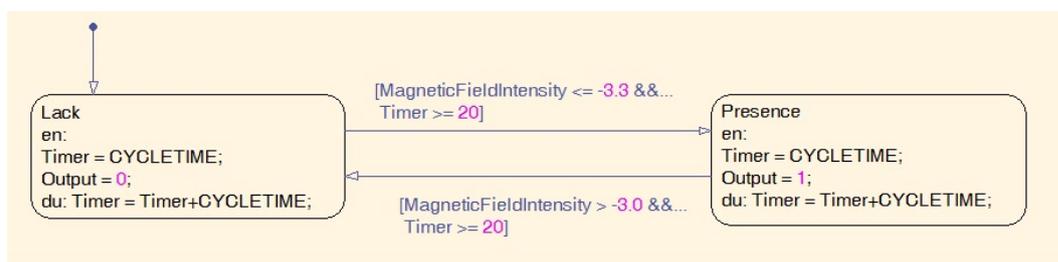


Figure 3.15 Hysteresis comparator to detect the presence of marker tape

3.3.2 Guidance board

It's possible to define the guidance board as an enhanced marker board. In fact the guidance board has three magnetic sensors like the one mounted on the marker board, moreover it has a CAN bus interface whereby communicate with the Input Output board, see Figure 3.12.

Since the sensors are the same, the procedure to read the magnetic field intensity is the same, see Figure 3.14, but the differences are how the presence on magnetic tape is detected, then how the steering speed reference $\Delta\omega^*$ is calculated, see Figure 3.16, and the direction of magnetic field, in this case North. The magnetic tape has a particular shape, see Figure 3.16, like a Gaussian distribution. Thanks to this feature it's possible to reconstruct through the three measurements a similar shape in order to compare them and obtain the angle between the magnetic field measured and the reference magnetic field that the AGV has to follow. Due to the

control scheme that will be adopted it's necessary to derive the angle obtained in order to provide the steering speed reference, $\Delta\omega^*$, to the Input Output board.

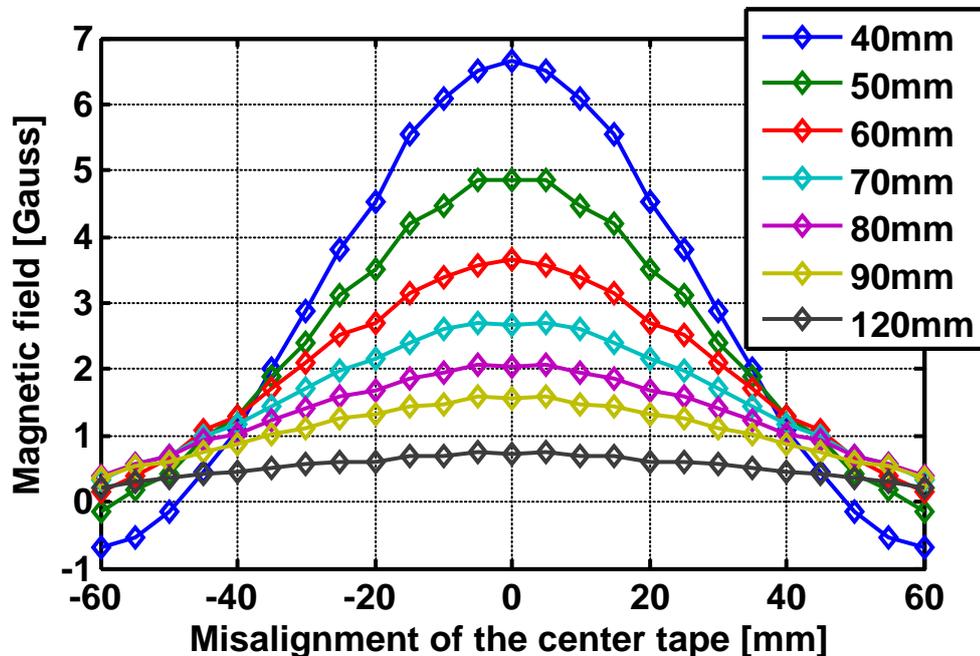


Figure 3.16 Shape of the magnetic field

There isn't checking time in order to get the steering speed reference, $\Delta\omega^*$, immediately. The threshold value has mainly two values:

- Low value when the driving group is up, 2.0Gauss
- High value when the driving group is down and the vehicle can follow the path, 3.3Gauss

They can be changed via touch screen when the vehicle is stopped.

This parameter allows to change the shape of the reconstructed normal distribution, in this way it's possible to use the same structure of signals processing for different types of magnetic tape.

After the reading of digital input, from the two marker boards, and the three magnetic sensor, the calculation of the steering speed reference, the guidance board sent the MTV CAN message, Table 3.2, to the Input Output board on the CAN bus.

All these routines are evaluated every 10ms.

3.3.3 Multipurpose board

The software that manages this board is very simple. In fact it receives the commands via CAN bus and provides power to two DC brushed motors through two H-bridges, see Figure 3.17.

Thanks to the H-bridge configurations it's possible to obtain both clockwise and counterclockwise rotation, see Figure 3.18. Furthermore it is possible to stop the motor by two different braking configurations, see Figure 3.19, thus the software to command the two H-bridges has to avoid only contemporary closing of transistor 1 and 2 or 3 and 4 otherwise it will create a short-circuit, see Figure 3.17.

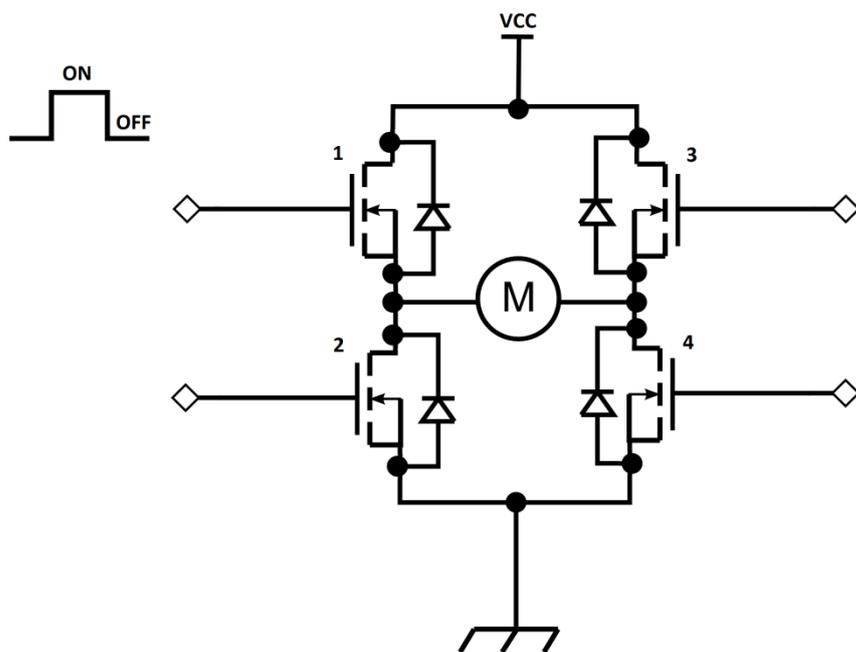


Figure 3.17 H-Bridge: basic diagram

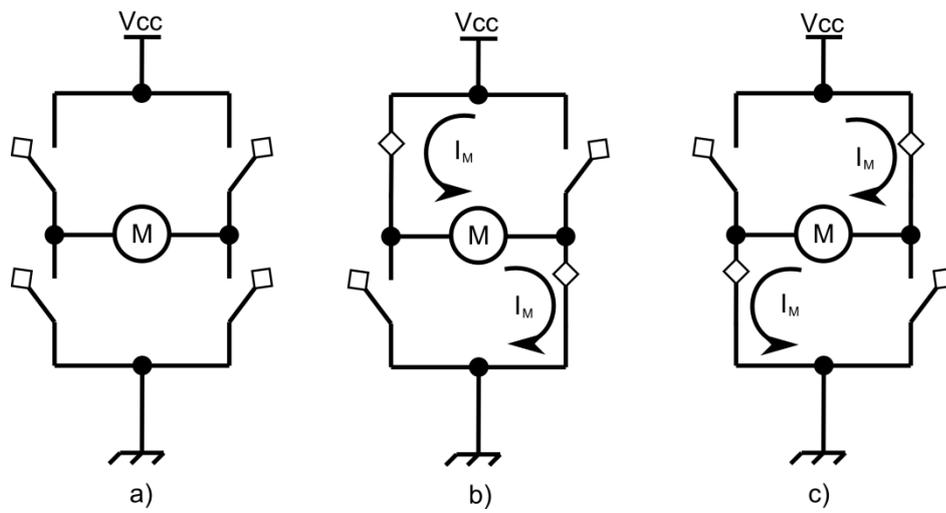


Figure 3.18 H-bridge running configurations: a) free running, b) clockwise rotation, c) anticlockwise rotation

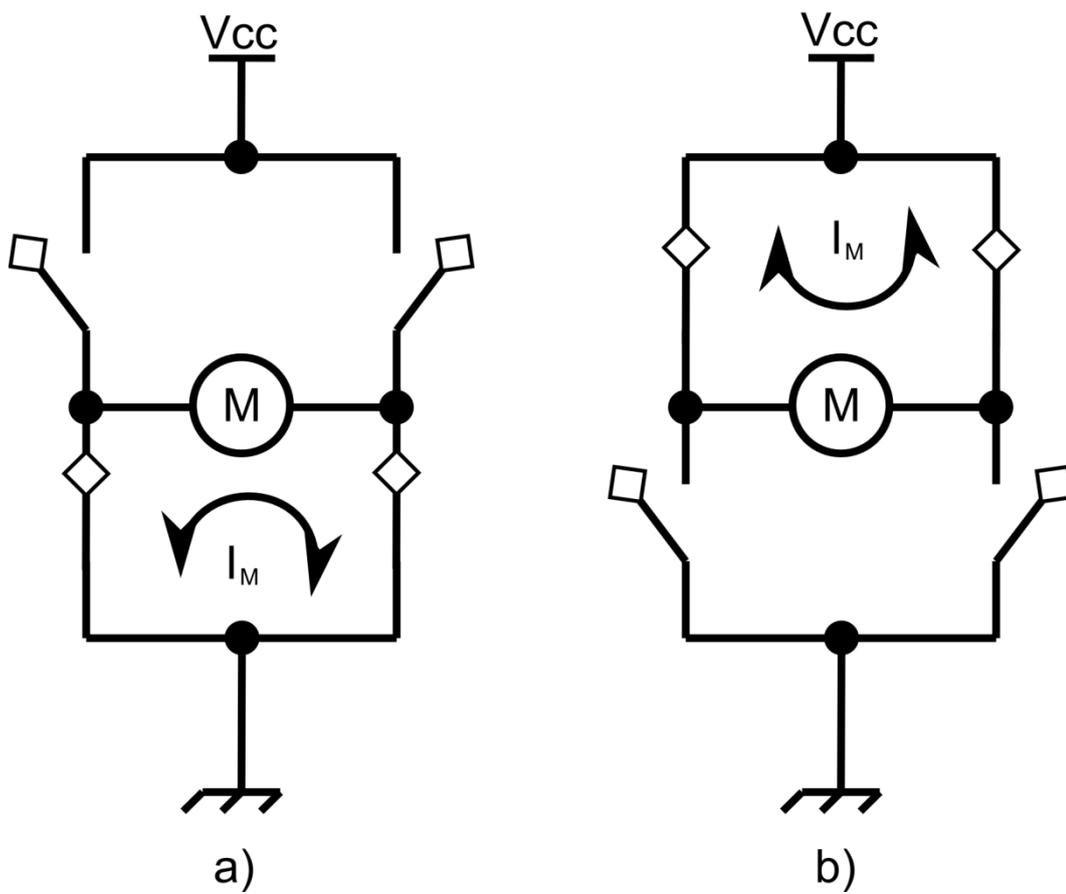


Figure 3.19 H-bridge braking configurations

3.3.4 Wi-Fi board

The software implemented in the Wi-Fi board has to guarantee safety in some critical situations. For example when two AGVs approach the same path intersections the software has to be able to decide which AGV has the priority to pass the cross and to avoid a possible deadlock, see Figure 3.20 a) .

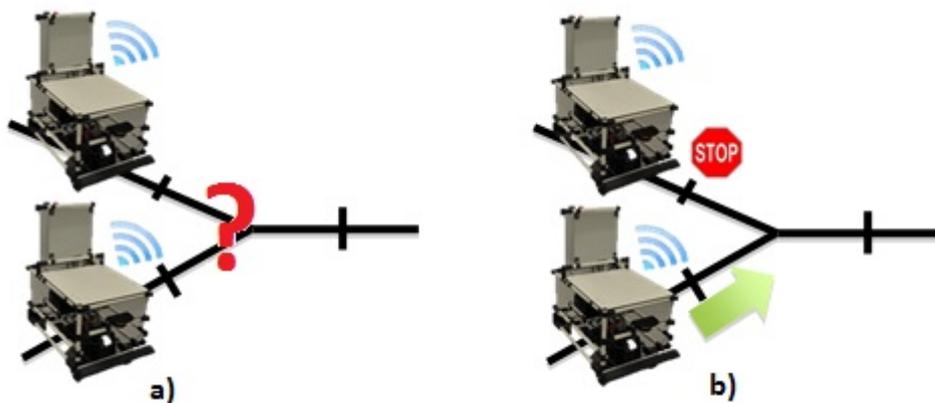


Figure 3.20 Path intersection: a) no decision, possible deadlock b) Wi-Fi board software resolves the possible deadlock

The Wi-Fi board software, with the aid of three parameters, creates an ad-hoc Wi-Fi network when it reaches the marker before the intersection. The three parameter are:

- Activation intersection mode command
- Number of marker of branch where the AGV is situated
- Number of marker in concurrent branch

In this way when an AGV creates an ad-hoc Wi-Fi network and no other AGV joins the network the first one can occupy the intersection. Otherwise the AGV that joins the Wi-Fi network stops and waits for the clearance of the intersection, see Figure 3.20 b). The intersection becomes free, after an occupation, when the first AGV reaches the marker after the intersection and turn-off the ad-hoc Wi-Fi.

Another critical situation is the interface between AGV and a smart roller conveyor, see Figure 3.21. A smart roller conveyor is a roller conveyor equipped

with a PLC that activates or deactivates the conveyors motors according to the command coming from AGV.

On the AGV and on the destination bench there are powered conveyors which rotate at the same speed in order to allow a smooth and controlled product transfer. To rotate at the same speed and with synchrony a Wi-Fi connection is set between the AGV and the PLC that manages the smart roller conveyor. In this case the Wi-Fi board software has to establish a connection and pass information and commands between AGV and smart roller conveyor.

This is an easy solution for the transfer of fragile or very heavy loads.



Figure 3.21 AGV with smart roller conveyor

3.3.5 Input Output board

The Input Output board software is certainly more complicated than the ones implemented in the other boards. To make easier the implementation of several algorithms that compose the software management of the vehicle, the logics are written in Matlab Simulink and then automatically converted in C code to allow the integration with firmware and create the entire software to download on the

microcontroller. This is the procedure normally used in automotive, aerospace and railways industry to save time and money: doing this way it's possible to forget about pure coding problems and to focus only on the problem to be solved, link the code to system requirements and simulate the code before field test.

The structure used to target the software on microcontroller is showed in Figure 3.22.

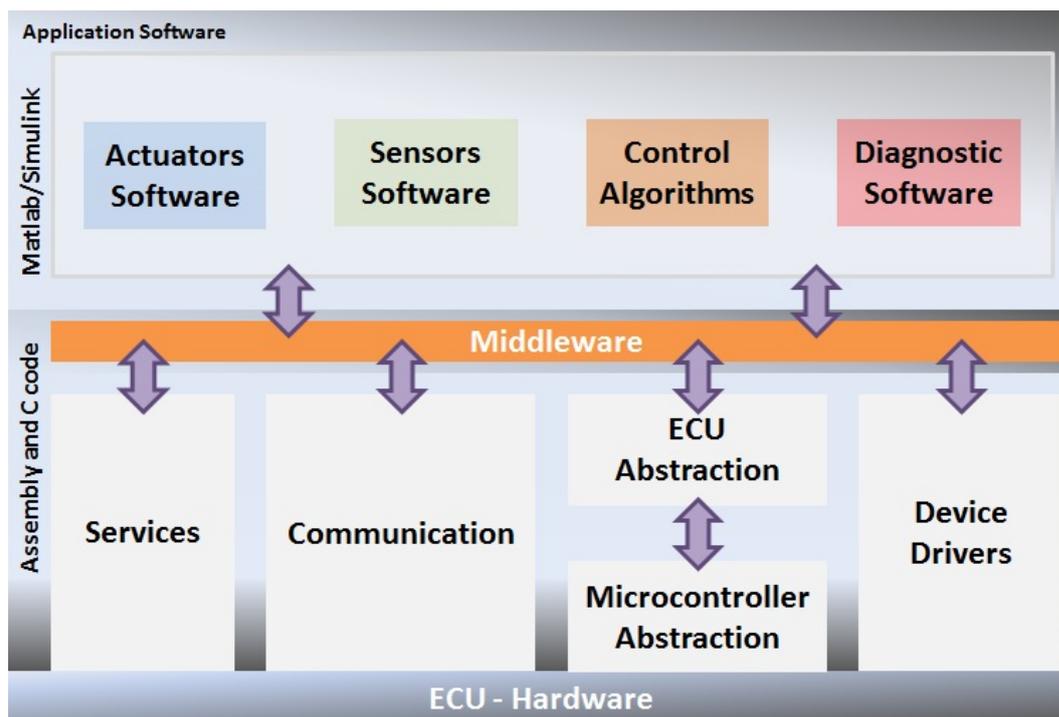


Figure 3.22 Software architecture for Input Output board

The middleware is an interconnection layer between application software and firmware. In this layer, every cycle time (here 10ms) the input of application software are updated, the main function of application software is called (in this case `Frame()`) and then the output are calculated and performed outside the Input Output board, see the middleware structure in Figure 3.23.

The application software is organized in macro blocks: System manager, AGV control, Input manager and Diagnostics.

Each block is divided in other sub-blocks, e.g. see Figure 3.24, thanks to the modularity of Matlab Simulink [6]. Since a lot of logics can be modeled as finite state machines, they have been implemented with the aid of Stateflow a product of

Matlab Simulink. Stateflow is very useful because it's possible to implement the control logics graphically using flow diagrams, as already shown in Figure 3.15.

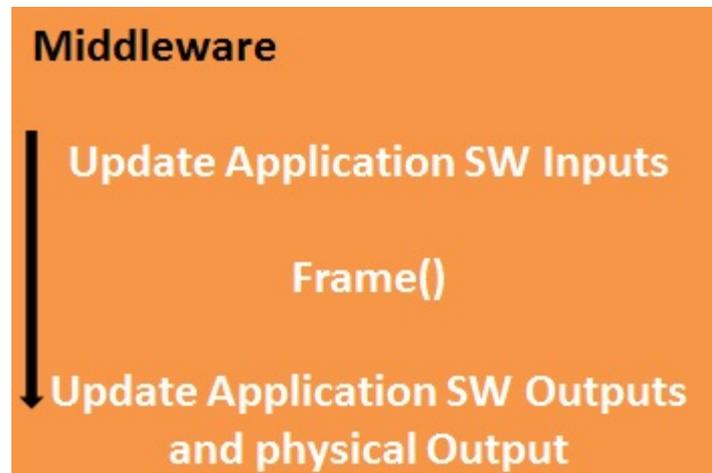


Figure 3.23 Middleware structure

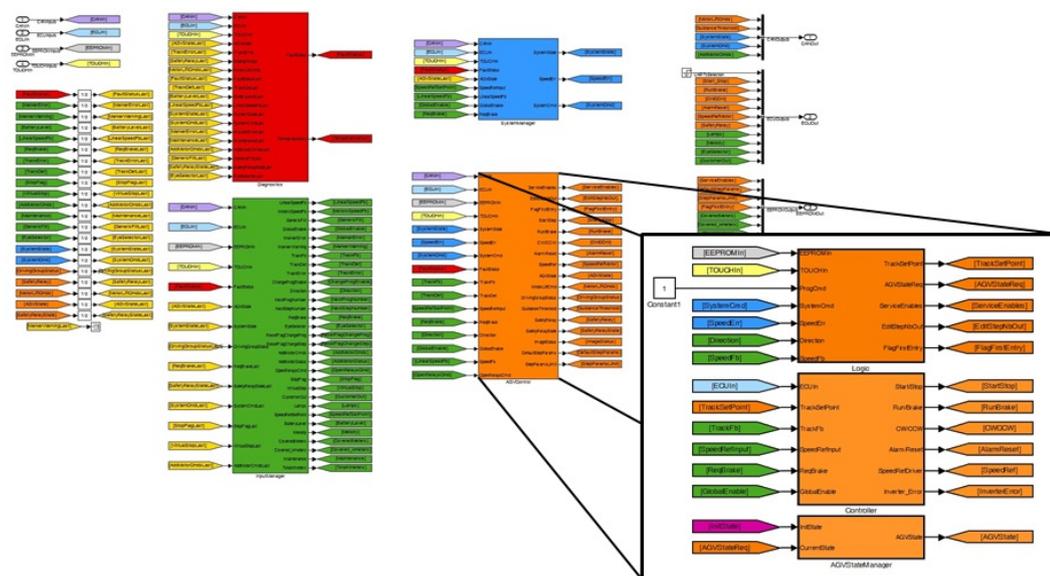


Figure 3.24 Example of sub-blocks division

In order to create a simple project management for each sub-block the relevant files .c and .h (header file) are generated with the same name of the sub-block. In this way is very simple to identify the auto-generated code and to verify it.

The macro-block *System manager* handles all the states in which the vehicle can be found like recovery after a failure or service if the vehicle is in programming mode, moreover this macro-block gives all the commands to the others blocks about the type of services.

The macro-block *Input Manager* treats all the inputs of the application software through filters and logics. The most important block is *Global Enable* which, depending on the operator request, the system state and programs loaded, provides the signal to start, stop and perform work cycle to the AGV, see Figure 3.25.

The macro-block *AGV Control* manages the control path, the lift motor, the lamps, the melody unit, the threshold for the guidance board, the safety relays and the optical sensor.

In the *Controller* sub-block, the core of the vehicle, is implemented the path control, for this topic please refer to Chapter 5.

The macro-block *Diagnostic* handles all the failure passing through two step: the first is the fault detection and the second is the fault validation. These are some failures that can be isolated:

- Battery low
- Bumper pressed
- Derailment
- Inverters
- Additional motor
- Over temperature
- Fuse breaking
- No CAN connection with one or more boards
- Optical sensor
- Obstacle on the trip direction

According to the number and the severity of the failure(s) the vehicle reduces gradually its functionality up to the impossibility to perform any commands before the intervention of an operator.

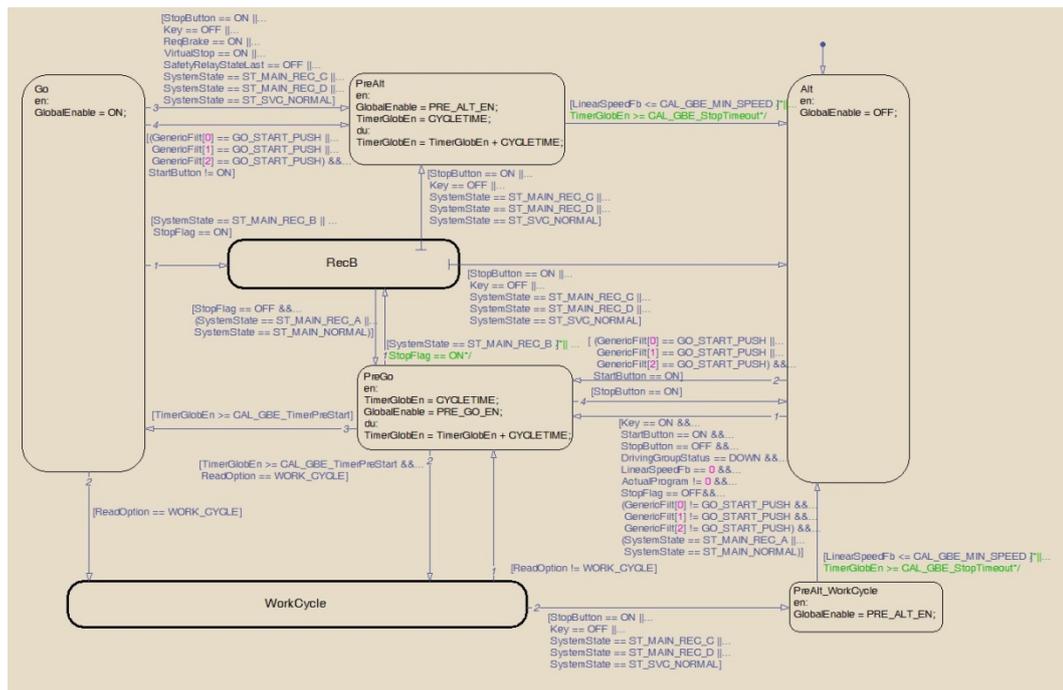


Figure 3.25 Global enable state chart

3.3.6 Touch panel

The touch panel has a dedicated software to create the screens, called “Screen Editor”, downloadable for free from the Delta Electronics web site. “Screen Editor” is a Windows-based and user-friendly software. It’s very simple to use and to associate the functionalities to the virtual buttons.

By using this software the user can quickly edit images and graphs and set suitable communication protocol through Macro command. A simple screenshot of “Screen Editor” is shown in Figure 3.26.

In Figure 3.27 some example of implemented screens. A very helpful feature of this software is the possibility to use the same screens with different languages, in this way the project to develop is only one and for the user is very easy to add another language to the project.

In Figure 3.28 is shown the screenshot relevant to the current failures on a vehicle. The sentence identifying the type of the faults flows on the red part of the touch

panel continuously till the pressure of the reset button. If the fault is still present after the reset the touch panel will automatically show the failure screen.

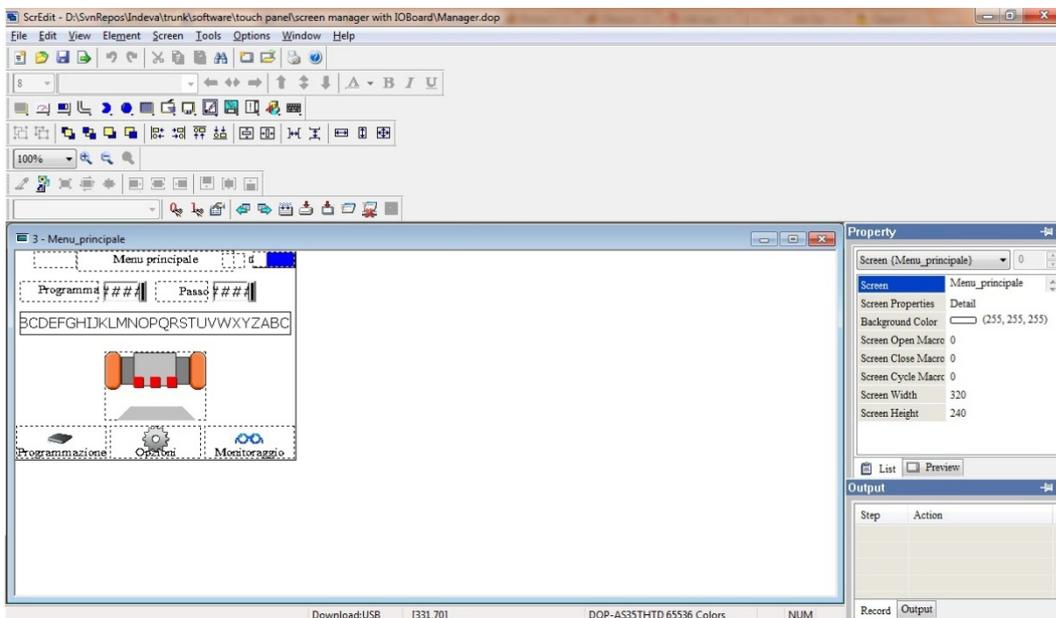


Figure 3.26 Screenshot of “Screen Editor” software

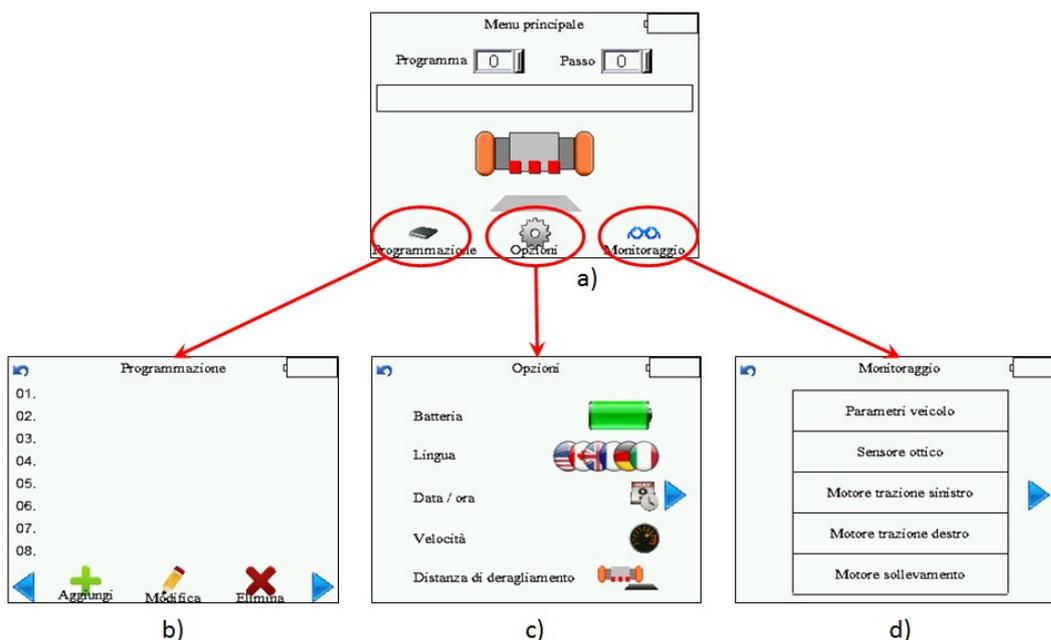


Figure 3.27 Touch panel example screens: a) main screen, b) first programming screen (list of programs), c) options screen and d) monitoring screen



Figure 3.28 Screenshot showing a failure on the vehicle

When it's necessary to create a lot of different screens, like in this case, it's very important to link the screens in an intuitive way. Moreover it's necessary to maintain a tree structure of the software, like show in Figure 3.29, in order not to duplicate the screenshots.

To update all the variables shown on the screen it is essential the communication with the Input Output board by Modbus. In fact for each screenshot the touch panel asks to the Input Output board the value of each variable displayed every 300ms. It's of capital importance to preserve the correspondence between variables value inside the Input Output board and touch panel to display always the correct values, in order to help the operator job and to facilitate a fast problems identification and solution.

3.4 Concluding remarks

In this chapter was described the prototype considered in this work. The first description was relevant to the hardware architecture like electronic boards and

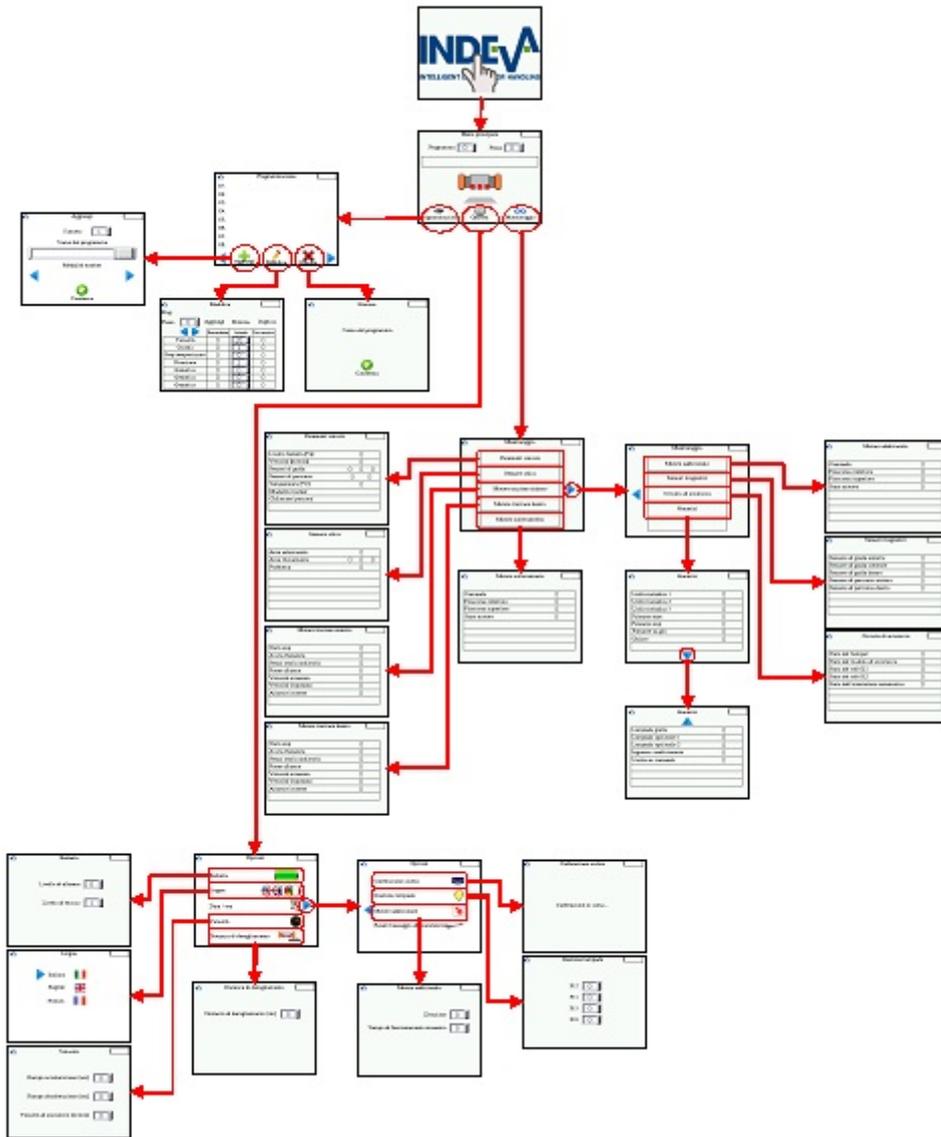


Figure 3.29 Tree structure touch panel software

driving group with brushless motors. The second one was relevant to the signal routing inside the AGV among the several boards. The last one was relevant to the software architecture, in general and for each electronic board.

Chapter 4

AGV mechanical model

It's very important to describe mathematically the AGV in order to design an optimal controller.

The AGV will be described with a dynamic model considering the structure of steering system and the friction forces acting on the wheels and on the vehicle.

Finally we will get a simple model of AGV on which it will be possible to design the controllers both for steering speed and vehicle longitudinal speed.

4.1 The AGV dynamic model

The vehicle can be modeled considering, first, the kinematic model of the differential drive as shown in Figure 4.1. As well known, in the considered framework it is possible to describe the vehicle kinematic using only two variables [1]

- the speed \bar{v} tangent to the vehicle trajectory
- the angular speed $\dot{\vartheta}$ due to the rotation around the Instantaneous Center of Rotation (ICR)

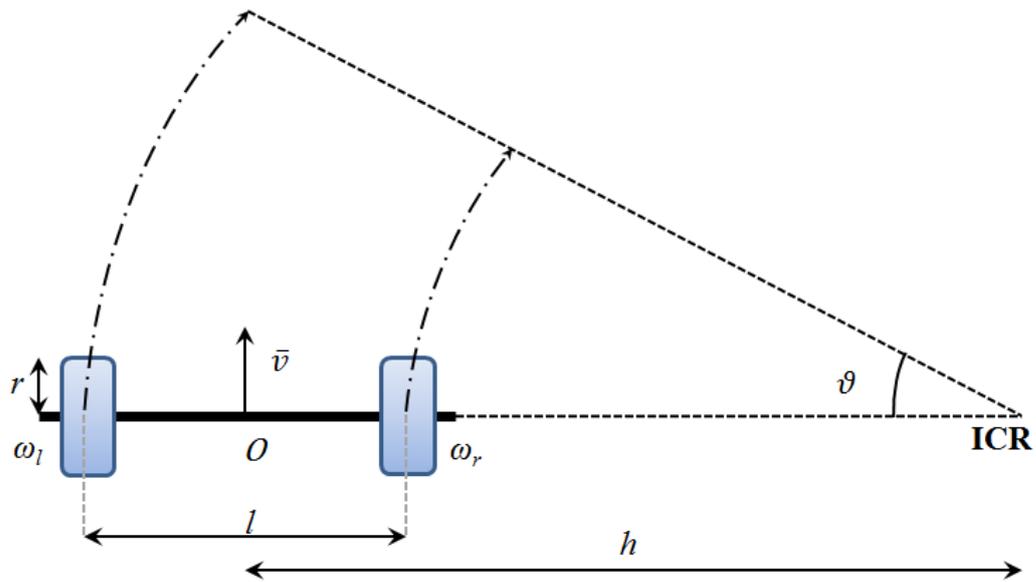


Figure 4.1 Kinematic model of a differential drive

Being (Figure 4.1)

- l is the distance between the driving wheels
- r is the driving wheel radius (assumed equal for both wheels)
- h is the distance between O (the midpoint of vehicle axis) and the ICR

Accordingly, in the hypothesis that no wheel slip occurs, the tangent speed \bar{v} and the angular speed $\dot{\vartheta}$ are:

$$\begin{cases} \bar{v} = \frac{\omega_l + \omega_r}{2} r \\ \dot{\vartheta} = \frac{\omega_l - \omega_r}{l} r \end{cases} \quad [4.1]$$

where ω_l and ω_r are the rotational speed of the left and the right wheels, respectively, which are measured variables for the considered vehicle.

Furthermore, h can be expressed as

$$h = \frac{l}{2} \frac{\omega_l + \omega_r}{\omega_l - \omega_r} \quad [4.2]$$

In order to control the vehicle it is necessary to derive a control oriented model, having in mind that the physical control inputs for the system are the two motor currents, which will be regulated by the inverters, while the controlled variables are

$$\begin{cases} \bar{\omega} = \frac{\omega_l + \omega_r}{2} = \frac{\bar{v}}{r} \\ \Delta\omega = \omega_l - \omega_r = \dot{\theta} \frac{l}{r} \end{cases} \quad [4.3]$$

The variables $\bar{\omega}$ and $\Delta\omega$ in equation [4.3] represent the *vehicle longitudinal speed* and the *steering speed*, respectively.

Coherently with such choice, the control variables are defined as:

$$\begin{aligned} \bar{i} &= \frac{i_{mot}^l + i_{mot}^r}{2} \\ \Delta i &= i_{mot}^l - i_{mot}^r \end{aligned} \quad [4.4]$$

where i_{mot}^l and i_{mot}^r represent the input current of the left and right motor, respectively.

Once both control and controlled variables have been defined, the Input/Output system dynamics can be described by the following MIMO model

$$\begin{bmatrix} \bar{\omega} \\ \Delta\omega \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} \bar{i} \\ \Delta i \end{bmatrix} \quad [4.5]$$

where $G_{11}, G_{12}, G_{21}, G_{22}$ are suitable transfer functions.

Now, consider the schematic view given in Figure 4.2, where

- J_{rot} is the rotational inertia of the vehicle;
- F_{in} is the inertial force;
- F_c is the centrifugal force;
- F_{ly} and F_{ry} are the lateral forces acting on the left and right driving wheels;

- $F_{frict}^{front-l}$ and $F_{frict}^{front-r}$ are the friction forces acting on the pivoting wheels;
- F_{frict}^{rear-l} and F_{frict}^{rear-r} are the friction forces acting on the left and right drag wheels;
- F_{frict}^l and F_{frict}^r are the friction forces acting on the left and right driving wheels
- F_{drag} is the aerodynamic drag force acting on the vehicle;
- F_{mot}^l and F_{mot}^r are the traction forces provided by the electric motors;
- M is the mass of the vehicle;

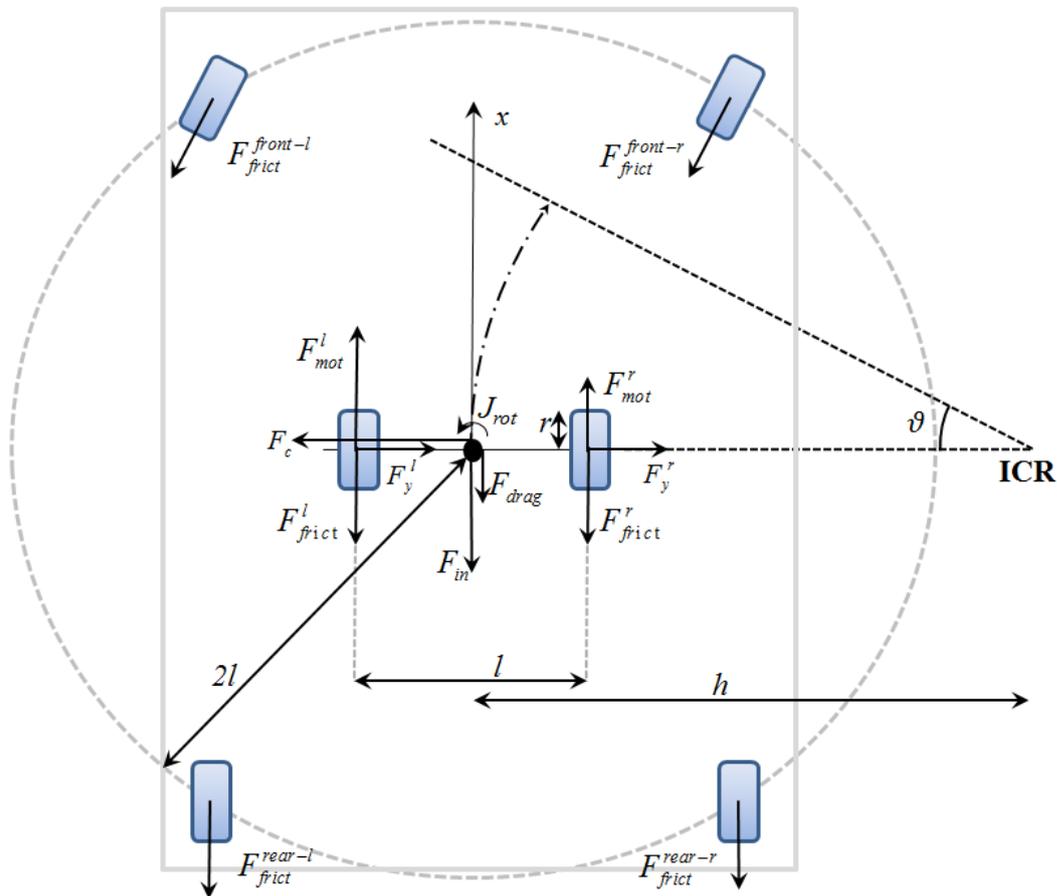


Figure 4.2 Schematic view of the forces on the considered vehicle

The wheel radius r is assumed equal for pivoting, drag and driving wheels. The center of mass is supposed to be coincident with the axis midpoint O . The vehicle

chassis can be modeled as a parallelepiped. The vehicle is considered as a rigid body moving on the plane with two degrees of freedom, since it can be supposed that, in normal operation, the centrifugal forces are always balanced by the lateral friction:

$$F_c = F_y^l + F_y^r \quad [4.6]$$

The friction forces acting on the vehicle are:

- Rolling resistance:

$$F_{roll} = C_{roll} Mg \quad [4.7]$$

- Dynamic resistance:

$$F_{dyn}(\bar{v}) = \beta_{dyn} \bar{v} \quad [4.8]$$

- The aerodynamic drag force:

$$F_{drag}(\bar{v}) = \frac{1}{2} \rho c_v A (\bar{v} + v_{wind})^2 \quad [4.9]$$

where:

- C_{roll} is the rolling coefficient;
- β_{dyn} is the dynamic viscous resistance coefficient;
- ρ is the density of the air;
- c_v is the drag coefficient;
- A is the reference front vehicle area;
- v_{wind} is the speed of the wind.

So the total friction force can be represented by the following equation:

$$F_{frict}(\bar{v}) = C_{const} + C_{prop} \bar{v} + C_{quad} \bar{v}^2 \quad [4.10]$$

Since in normal operating conditions the vehicle maximum speed is 3km/h and it is used indoor, it is possible to omit the quadratic term in equation [4.10].

Hence, the friction forces for the driving wheels can be expressed as follows:

$$F_{frict}^i(\omega_i) = C_{const}^i + C_{prop}^i \omega_i r, \quad i = l, r \quad [4.11]$$

In order to estimate the parameters of equation [4.11], coasting down experiments have been carried out by letting the system inertially decelerate from its maximum nominal speed (3km/h) to zero. The estimated parameters are listed in Table 4.1.

	C_{const} [N]	C_{prop} [Ns/m]
Left wheel	6.371	16.096
Right wheel	6.432	16.245

Table 4.1 Coefficients from coasting-down friction identification

Since $C_{const}^l \cong C_{const}^r$ it is possible to use a single parameter C_{const} in the following.

As a final remark on wheel parameters estimation, it is worth noting that, introducing aerodynamical friction in the model, the corresponding drag coefficient estimated by coasting down experiments is negligible for all tests.

Now, let's write the force balance along the vehicle longitudinal x axis:

$$\Sigma F^x = F_{in}^x + F_{frict}^x - F_{mot}^x = 0 \quad [4.12]$$

where

$$F_{mot}^x = \frac{\eta nk}{r} (i_{mot}^l + i_{mot}^r) = \frac{2\eta nk}{r} \bar{i} \quad [4.13]$$

$$F_{frict}^x(\bar{v}) = (C_{prop}^l + C_{prop}^r) \bar{v} + (C_{prop}^r - C_{prop}^l) \Delta v + 2C_{const} \quad [4.14]$$

and

$$F_{in}^x = \left(M + \frac{2J_p}{r^2} + \frac{J_{mot}\eta}{r^2 n^2} + 2M_w + \frac{2J_{pivot}}{r^2} \right) \dot{\bar{\omega}} r \quad [4.15]$$

where

- J_{pivot} is the inertia of each pivoting wheel;
- η is the gearbox efficiency;
- n is the gearbox reduction ratio;
- k is the torque constant;
- J_p is the inertia of vehicle;
- M_w is the mass of each drag wheel

Notice that in equation [4.15], M , the total mass vehicle, is a largely dominant addendum with respect to the other and so equation [4.15] can be strongly simplified as follows:

$$F_{in}^x = M \ddot{\omega} r \quad [4.16]$$

Thus, equation [4.12] takes the form

$$M \ddot{\omega} r + (C_{prop}^l + C_{prop}^r) \bar{v} + (C_{prop}^r - C_{prop}^l) \Delta v + 2C_{const} = \frac{2k\eta n \bar{i}}{r} \quad [4.17]$$

Now, consider the torque balance around the center of mass of the vehicle, which can be described as follows:

$$\Sigma T^{(o)} = \Sigma T_{in}^{(o)} + \Sigma T_{frict}^{(o)} - \Sigma T_{driving}^{(o)} = 0 \quad [4.18]$$

The inertia torque $\Sigma T_{in}^{(o)}$ is given by

$$\Sigma T_{in}^{(o)} = J_{rot} \ddot{\mathcal{G}} \quad [4.19]$$

The driving torque, according to the convention adopted in Figure 4.2 can be calculated as

$$\Sigma T_{driving}^{(o)} = \frac{l}{2} (F_{mot}^l - F_{mot}^r) = \frac{l}{2} \frac{\eta n k}{r} (i_{mot}^l - i_{mot}^r) \quad [4.20]$$

The friction torque $\Sigma T_{frict}^{(o)}$ has the form

$$\Sigma T_{frict}^{(o)} = \frac{l}{2} (F_{frict}^l - F_{frict}^r) + b F_{frict}^{pivot-drag} \quad [4.21]$$

Since $C_{rear}, C_{rear} \ll \frac{C_{prop}^l + C_{prop}^r}{2}$ and ϑ is a small angle in normal operations, the last term in equation [4.21] representing the friction forces due to pivoting and fixed wheels can be neglected.

Thus, equation [4.18] takes the form:

$$\begin{aligned} J_{rot} \frac{r}{l} \Delta \dot{\omega} + \frac{lr}{2} (C_{prop}^l + C_{prop}^r) \Delta \omega \\ + \frac{lr}{2} (C_{prop}^l - C_{prop}^r) \bar{\omega} + \frac{l}{2} (C_{const}^l - C_{const}^r) \\ = \frac{l}{2} \frac{\eta nk}{r} \Delta i \end{aligned} \quad [4.22]$$

Finally, the state equations of the model are

$$\left\{ \begin{aligned} \dot{\bar{\omega}} &= -\frac{(C_{prop}^l + C_{prop}^r)}{M} \bar{\omega} - \frac{(C_{prop}^r - C_{prop}^l)}{2M} \Delta \omega - \frac{2C_{const}}{Mr} + \frac{2k\eta n}{Mr^2} \bar{i} \\ \Delta \dot{\omega} &= \frac{l^2}{2J_{rot}} (C_{prop}^r - C_{prop}^l) \bar{\omega} - \frac{l^2}{4J_{rot}} (C_{prop}^l + C_{prop}^r) \Delta \omega \\ &\quad - \frac{l^2}{2rJ_{rot}} (C_{const}^l - C_{const}^r) + \frac{l}{2} \frac{\eta nk}{r} \Delta i \end{aligned} \right. \quad [4.23]$$

The output equation is

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{\omega} \\ \Delta \omega \end{bmatrix} \quad [4.24]$$

Since \mathbf{B} and \mathbf{C} matrices are diagonal, with reference to equation [4.5], the off-diagonal transfer functions G_{12} and G_{21} are zero and so the system is decoupled, i.e. the 2x2 MIMO system is simply made by two SISO systems.

Finally, remembering that $C_{const} = C_{const}^l \cong C_{const}^r$ and $C_{prop} = C_{prop}^l \cong C_{prop}^r$, all the friction coefficients for the left and right wheels can be assumed equal. Therefore, the state equations become

$$\begin{cases} \dot{\bar{\omega}} = -\frac{C_{app}}{M}\bar{\omega} - \frac{2C_{const}}{Mr} + \frac{2k\eta n}{Mr^2}\bar{i} \\ \Delta\dot{\omega} = -\frac{l^2 C_{app}}{2J_{rot}}\Delta\omega + \frac{l}{2}\frac{\eta nk}{r}\Delta i \end{cases} \quad [4.25]$$

where

$$C_{app} = \frac{C_{prop}^l + C_{prop}^r}{2} \quad [4.26]$$

4.2 Concluding remarks

In this chapter has been described how to get the AGV dynamic model. So, through the structure of the model, it's possible to manage two different control loops to command the *vehicle longitudinal speed* $\bar{\omega}$ and the *steering speed* $\Delta\omega$ thanks to the current control loop of each brushless motor.

Chapter 5

Design of path control

Large load weight values have a high impact on the guidance performances, even affecting the AGV mechanical stability, and strong speed limitations are usually applied. Moreover, when designed to move loads in a large range of weight values, from small light plastic objects to heavy metal components, the performance are conservatively chosen on the basis of the largest expected payload.

We focused our attention to the vehicle guidance control problem and aim to design controllers that guarantee desirable specifications for a wide range of possible load mass values. Hence, we assume that the payload mass is an uncertainty in the model and resort to robust control design methods. Moreover, we'd like not only to find one suitable controller but to analyze the stability domain in the controller parameter space, so that we can evaluate the effects on the guidance performances produced by changes in the controller parameters, preserving control system stability. In order to obtain such results we pay special attention to randomized techniques. The variety of randomized algorithms for control can be found in [2]; special techniques for fixed order controller design are described in [3], [4]; the problem of static output control feedback is discussed in [5]. We prefer this parameter space approach because it allows not only to compute controller parameters but also to compute a discrete set of stabilizing

controllers and to obtain a configuration of the admissible domain in the parameter space. Moreover, it is demonstrated in [4] how randomized techniques are oriented to deal with basic notions for any engineering characteristics — gain or phase margin, overshoot or other time-response characteristics, robustness margin — as well as mathematical objectives such as H_2 or H_∞ norm. Specifically, randomized method are applied to the tuning of PI controllers [6], taking into account control action limitations, due to the limited current supply values; system speed specifications, defined by desired values for settling time; H_∞ performances.

5.1 A parameter space approach for the description of H_∞ region

The aim of this paragraph is to give an instrument to describe the regions in the controller parameter space such that the H_∞ -norm of the closed loop transfer function of a given system is less than a prescribed level, see [7]. Will be considered a particular family of linear control systems in particular plant with PID controller.

The major obstacle to design a fixed-order controller is the difficulty in characterizing the entire set of stabilizing controllers in the parameter space and thus the subset of those that satisfy the H_∞ criterion. For PID controllers there exist many works which deal with the aforementioned issues. In this work will be used a technique to describe the total set of controllers that satisfy H_∞ criterion. The approach is based on representing L_∞ admissible set as an intersection of non-definite quadratic forms. We consider a general state space representation of the system with multiple inputs or outputs and a particular attention is paid to the input/output transformation and the related risk of looseness of system reachability and observability.

Consider a plant of order n with a control $u(t)$, in a state space description:

$$\begin{cases} \dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t) \\ z(t) = C_1x(t) + D_{11}w(t) + D_{12}u(t) \\ y(t) = C_2x(t) + D_{21}w(t) \end{cases} \quad [5.1]$$

where $x(t) \in \mathbb{R}^n$, $w(t) \in \mathbb{R}^p$, $y(t) \in \mathbb{R}^q$ and $z(t) \in \mathbb{R}^r$. The real matrices involved in the system description are of appropriate dimension, and the feedback control law is $u(t) = k^T y(t)$ (with the convention of a k as a column vector).

Taking into account the control law, the equation [5.1] results as follows:

$$\begin{cases} \dot{x}(t) = (A + B_2k^T C_2)x(t) + (B_1 + B_2k^T D_{21})w(t) \\ z(t) = (C_1 + D_{12}k^T C_2)x(t) + (D_{11} + D_{12}k^T D_{21})w(t) \end{cases} \quad [5.2]$$

The transfer function from input w to output z is described in equation [5.3]:

$$T(s) = (C_1 + D_{12}k^T C_2)(sI - A - B_2k^T C_2)^{-1} (B_1 + B_2k^T D_{21}) + (D_{11} + D_{12}k^T D_{21}) \quad [5.3]$$

Let $T(s)$ be an element of the space L_∞ . The norm of $T(s)$ is the scalar:

$$\|T(s)\|_\infty = \sup_{\omega} \bar{\sigma}[T(j\omega)]; \quad [5.4]$$

where $\bar{\sigma}$ denotes the greatest singular value.

In this note, we deal mainly with the set \mathcal{K} of all the vectors k such that the L_∞ norm of the related system is less than a prescribed level γ , equation :

$$\mathcal{K} = \{k : \|T(s, k)\|_\infty \leq \gamma, T(s, k) \in L_\infty\} \quad [5.5]$$

Among them, the final aim is to find the sub-region \mathcal{K}_S of \mathcal{K} so that the vector k is stabilizing for system [5.2], that means we are interested finally in the H_∞ region

$$\mathcal{K}_S = \{k : k \in \mathcal{K} \text{ and } k \text{ stabilizing}\} \quad [5.6]$$

This technique provide a description of the admissible region \mathcal{K} , in terms of simple conditions quadratically depending on controller parameters. Moreover the stable subregions are well separated from the unstable ones, so they can be easily marked.

Given a system [5.1], \mathcal{K} has the following equivalent description:

$$\frac{k_a^T N(\omega) k_a}{k_a^T P(\omega) k_a} \leq \gamma^2, \forall \omega \quad [5.7]$$

where N and P are $(q + 1) \times (q + 1)$ positive semidefinite matrices in the real variable ω and k_a is the augment vector $k_a = \{k_1, k_2 \dots, 1\}$.

This theorem allows to describe the region \mathcal{K} as a family of subset in \mathbb{R}^p defined by non –definite quadratic forms in k .

In order to tackle the problem [5.5] and [5.6] in terms of non-definite quadratic forms, for the $\Gamma(\omega) = N(\omega) - \gamma^2 P(\omega)$ the following sets are defined:

$$\mathcal{C}_\omega = \{k : k_a^T \Gamma(\omega) k_a \leq 0\} \quad [5.8]$$

$$\mathcal{C} = \bigcap_{\omega \in [0, \infty)} \mathcal{C}_\omega \quad [5.9]$$

the intersection domain is limited to $[0; \infty)$ since Γ is symmetric with respect to ω . It is worth noticing that definition of \mathcal{C}_ω follows from [5.7] if its denominator is not null. Notice that there is no problem with dealing with the inequality sign, since $P(\omega)$ is positive semidefinite. The aim is to show the equivalence of set \mathcal{K}

and set \mathcal{C} , and finally tackle the stability issue in order to describe the subset of \mathcal{K} constituted by the stabilizing controllers.

The boundary $\partial\mathcal{K}$ is the set of controllers that lead to "regular" systems whose related norm is equal to γ , and that may lead to "pathological" systems whose related norm is less than γ . More precisely \mathcal{K} and \mathcal{C} are defined as follows:

$$\partial\mathcal{C} = \left\{ k : \exists \tilde{\omega} \ k_a^T \Gamma(\tilde{\omega}) k_a = 0, \ k_a^T \Gamma(\omega) k_a = 0 \ \omega \neq \tilde{\omega} \right\} \quad [5.10]$$

$$\partial\mathcal{K} = \left\{ k : \|T(s, k)\|_\infty = \gamma, \operatorname{Re}(\lambda_i) \neq 0 \right\} \cup \left\{ k : \|T(s, k)\|_\infty \leq \gamma, \exists i \operatorname{Re}(\lambda_i) = 0 \right\} \quad [5.11]$$

If k is a two dimensional vector (e.g. system with PI controller, PID with a fixed parameter, etc.), the problem posed by [5.5] is reduced to the analysis of conical curves given by [5.8], in terms of non-homogeneous quadratic form. In general, the equality in [5.8], $k_a^T \Gamma(\omega) k_a = 0$, defines a family of surface in \mathbb{R}^p . This approach has much in common with D-decomposition. Since \mathcal{K} can be described in terms of quadratic forms, it is possible to express inequality [5.8] as follows:

$$k^T \Gamma_0 k + k^T \Gamma_1 + \gamma_0 \leq 0; \quad \Gamma = \begin{bmatrix} \Gamma_0 & \frac{1}{2} \Gamma_1 \\ \frac{1}{2} \Gamma_1^T & \gamma_0 \end{bmatrix} \quad [5.12]$$

Therefore, representing \mathcal{K} by [5.12] the following conditions hold:

- $\Gamma > 0$ then $\mathcal{K} = \emptyset$. Indeed, if $\Gamma > 0$ then for definition of positive definite matrix, inequality in [5.8] is never satisfied for every element in \mathbb{R}^{q+1} , thus neither for k_a ;
- $\Gamma \leq 0$ then $\mathcal{K} = \mathbb{R}^q$; indeed if $\Gamma \leq 0$ then for definition of negative semidefinite matrix, inequality in [5.8] is satisfied for every element in \mathbb{R}^{q+1} , thus even for k_a ;
- $\Gamma_0 \leq 0$ then \mathcal{K} is not bounded. Actually for large values of k , [5.12] tends to the term $k \Gamma_0 k^T \leq 0$ which is negative semidefinite, thus admissible for \mathcal{K} ;

- $\Gamma_0 > 0$ then \mathcal{H} is bounded; notice that it is the vice versa of the previous property.
- if there exists $k \neq 0$ such that $k\Gamma_0k^T \leq 0$ for all ω , then \mathcal{H} is not bounded. Indeed, Suppose such a k exist, and take αk . With $\alpha \rightarrow \infty$, [5.12] tends to $\alpha^2 k\Gamma_0k^T \leq 0$, thus αk is admissible

So a practical description of $\partial \mathcal{H}$ is possible in this way:

$$\partial \mathcal{H} = \left(k : \begin{cases} k_a^T \Gamma(\omega) k_a = 0 \\ k_a^T \Gamma_{\omega}(\omega) k_a = 0 \\ k_a^T \Gamma_{\omega\omega}(\omega) < 0 \end{cases} \right) \cup \left(k : \begin{cases} k_a^T N(\omega) k_a = 0 \\ k_a^T P(\omega) k_a = 0 \end{cases} \right) \quad [5.13]$$

Where $\Gamma_{\omega}(\omega)$ and $\Gamma_{\omega\omega}(\omega)$ are the first and second derivative of $\Gamma(\omega)$ with respect to ω .

5.2 Vehicle motion control architecture

As shown in Figure 5.1 the control system is composed by two nested control loops: a servo-controller for motor currents and an external loop that regulates vehicle speed and steering speed as defined in equation [4.3]. The current control loops are embedded in the inverters that feed the DC brushless motors and they are designed to ensure the frequency decoupling with the mechanical outer loop.

The mechanical control loops are driven by two set-points: the desired longitudinal speed $\bar{\omega}^*$ provided by user and desired steering speed $\Delta\omega^*$ provided by the Guidance Electronic Board that computes it on the basis of the strip reading.

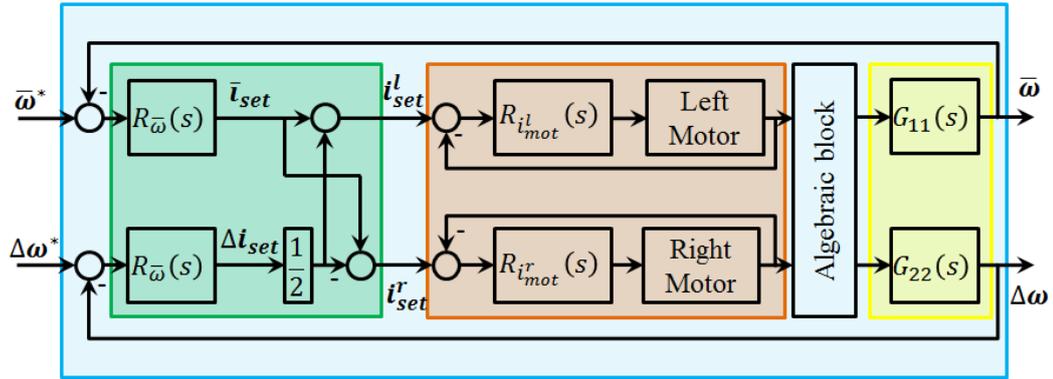


Figure 5.1 Decoupled control layout for the considered AGV

5.3 Parameter analysis for controller design

In this paragraph, randomized techniques are used to evaluate the stability domain in the parameter controller space. Randomized techniques are of particular interest in applications. In fact, they are oriented to deal in an immediate and intuitive way with basic control engineering design parameters, such as gain or phase margins, time-response characteristics. But they can accommodate as well also mathematical objectives such as H_2 or H_∞ norm.

In the present application, a PI controller [6] is considered, which is a very simple controller with only two parameters. In this way, the controller parameter analysis will be represented by two-dimensional plots. This will provide a very simple interpretation of the results obtained by application of randomized techniques. The first step is to generate samples in the stability (or robust stability) domain, i.e. to find controller parameter values (robustly) stabilizing the plant. In general, the geometry of this domain can be quite complicated [8], but sampling allows to overcome this difficulty.

In the following, first the sampling method for the stability domain is described and the robust stability domain with respect to mass variation is depicted. Then, the control action constraints (current limitations) and the performance

requirements (transient time duration) are shown in the stability domain. Finally, also H_∞ norm constraints are shown in the same picture, merging robustness, performance and constraint requirements in a single analysis, simple and easy to understand.

In these paragraphs will be analyzed the second equation of the system [4.25].

5.3.1 Sampling for stability domain for polynomials

Consider a LTI SISO plant $G(s) = \frac{h(s)}{e(s)}$. We wish to stabilize it with the controller $C(s) = \frac{f(s)}{g(s)}$ where polynomials $f(s), g(s)$ have fixed orders (for instance, as in the case of the present application, it can be a PI controller). The closed-loop characteristic polynomial of the system is linear in the controller parameters, given by the vector k , whose dimension is equal to the number of controller parameters so that: $p(s) = h(s)f(s) + e(s)g(s) = p_0 + \sum_{i=1}^n k_i p_i$

Suppose that at least one stabilizing controller is known, with parameter vector k^0 such that $p(s, k^0)$ is stable. Our aim is to generate samples k^i that are asymptotically uniform in the stability domain. To achieve this goal we apply the Hit-and-Run algorithm [9]. At every step, the algorithm generates a random direction uniformly sampled among all directions and then it chooses the next point by uniform sampling over the segment of the line along the chosen direction and inside the stability domain.

Let d be a random direction uniformly chosen in the unit sphere in \mathbb{R}^n , n being the controller parameter space dimension, in order to specify the segment of the line d in the stability domain, namely, the values $L = \{t \in \mathbb{R}: p(s, k^0) + t \sum d_i p_i(s) \text{ is stable}\}$.

It is necessary to solve the stability analysis problem for a SISO system with a real scalar gain. This problem can be analytically solved [11]. Then, a new sample $k^1 = k^0 + \tau d$ is chosen, where τ is uniform random in L . Finally, k^0 is replaced

with k^1 and the procedure is repeated to obtain an arbitrary large number of samples k^0, k^1, \dots, k^N .

A similar procedure can be applied to sample for robust stability domain for interval plant $G(s) = \frac{h(s,q)}{e(s,q)}$, $\underline{q} < q < \bar{q}$ (see [10] for details).

5.3.2 Robust stability domain

As described in paragraph 5.1 to use the “parameter space approach for the description of H_∞ region” it’s necessary to have a plant description in state space. In this case is very simple to obtain a state space representation, see equation [5.14].

$$A_p = -l^2 \cdot \frac{C_{app}}{2J_{rot}} \quad B_p = \frac{l^2 \cdot k \cdot \eta \cdot n}{2J_{rot} \cdot r^2} \quad C_p = 1 \quad D_p = 0 \quad [5.14]$$

The plant will be controlled with a PID with the following structure, equation [5.15]:

$$C(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{(1 + Ns)} \quad [5.15]$$

that can be re-written in state space, equation [5.16], see Figure 5.2:

$$A_c = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{N} \end{bmatrix} \quad B_c = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad C_c = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & -\frac{1}{N^2} \end{bmatrix} \quad D_c = \begin{bmatrix} 1 \\ 0 \\ \frac{1}{N} \end{bmatrix} \quad [5.16]$$

where the outputs are weighted by controller parameters k_p , k_i and k_d and finally added up.

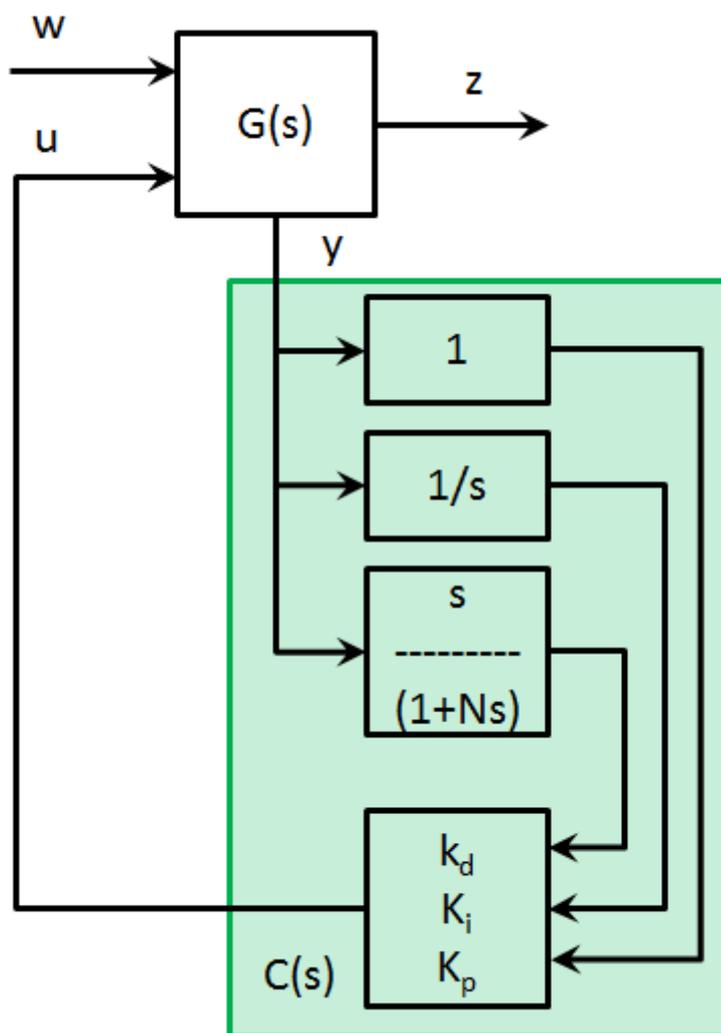


Figure 5.2 Close system diagram

Using the control law $u = k^T y$ is possible to obtain the following equation [5.17]:

$$A = \begin{bmatrix} A_p & 0 \\ -B_c C_p & A_c \end{bmatrix} \quad B_1 = \begin{bmatrix} 0 \\ B_c \end{bmatrix} \quad B_2 = \begin{bmatrix} B_p \\ 0 \end{bmatrix} \quad C_1 = [0] \quad [5.17]$$

$$C_2 = [-D_c C_p \quad C_c] \quad D_{21} = [D_c] \quad D_{11} = D_{12} = [0]$$

According to equation [5.3] and forcing $k_d = 0$ the closed loop frequency response can be written as, equation [5.18]:

$$T(j\omega) = \frac{-k_p\omega + k_I j}{k_p\omega + 0.06602\omega - k_I j + 3.84198 j\omega^2} \quad [5.18]$$

It's possible to use the equivalent description, through equation [5.7], of equation [5.18]:

$$\begin{aligned} N_1(\omega) &= [\omega \ 0 \ 0] \\ N_2(\omega) &= [0 \ -1 \ 0] \\ P_1(\omega) &= [\omega \ 0 \ 0.06602] \\ P_2(\omega) &= [0 \ -1 \ 3.84198\omega^2] \end{aligned} \quad [5.19]$$

Consistently with what is shown in equation

$$N(\omega) = \begin{bmatrix} \omega^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [5.20]$$

$$P(\omega) = \begin{bmatrix} \omega^2 & 0 & 0.06602\omega^2 \\ 0 & 1 & -3.84198\omega^2 \\ 0.06602\omega^2 & -3.84198\omega^2 & 0.00436\omega^2 + 14.7608\omega^4 \end{bmatrix}$$

So now it's possible to write the system according to equation [5.12] with $\gamma = 1.9$

$$\begin{aligned} \Gamma_0 &= \begin{bmatrix} -2.61\omega^2 & 0 \\ 0 & -2.61 \end{bmatrix} & \Gamma_1 &= \begin{bmatrix} -0.47666\omega^2 \\ 27.739\omega^2 \end{bmatrix} \\ \gamma_0 &= [-53.2866\omega^4 - 0.015735\omega^2] \end{aligned} \quad [5.21]$$

The transfer function from current Δi to steering speed $\Delta\omega$ depends on the mass value M and has the following form

$$G(s) = \frac{260.26}{Ms + 17.18} \quad [5.22]$$

We are looking for PI controllers $C(s) = k_p + \frac{k_i}{s}$ stabilizing the system for all mass values $50 \leq M \leq 1000 \text{ kg}$. In this case the precise robust stability domain description, obtained by D -decomposition technique, is:

$$\mathcal{K}_0 = \{k_i > 0, k_p > -0.066\} \quad [5.23]$$

Note that mathematically robust stability domain is not bounded but practically we are unable to realize controller with arbitrary large gains k_p and k_i .

Moreover, performance specifications on the controlled system dynamical behavior must be fulfilled and expressed as constraints in the controller parameter space.

Solving equation [5.13] for different values of γ it is possible to obtain regions where the H_∞ norm of the closed-loop system is more or less than γ chosen.

In this way the controller parameters k_p and k_i can be selected, but it's possible to insert other types of requests that the controller has to satisfy like limitation of control action and speed performance.

5.3.3 Limitations on control action

The control input $\Delta i < \Delta i^*$ value is restricted by 10A. This constraint naturally makes the stability parameter domain bounded. To estimate the feasible set we put 10000 samples in the domain $0 < k_i < 160, -0.066 < k_p < 50$ and calculate $\max \Delta i$ needed to realize such a controller. In Figure 5.3 the domains for different maximum admissible Δi are plotted. As a consequence of this analysis, we restrict the controller parameter values to $0 < k_i < 158, -0.066 < k_p < 20$.

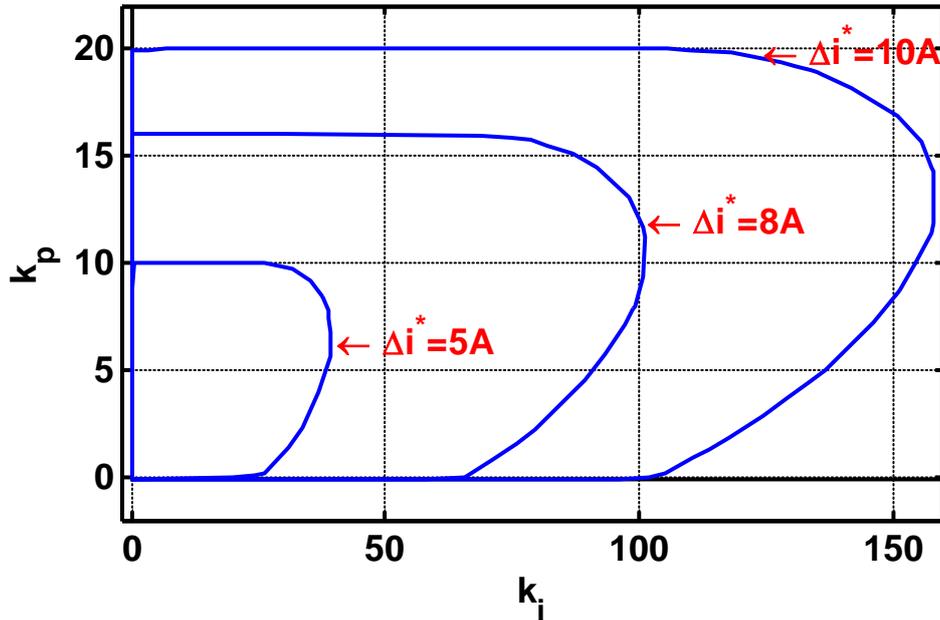


Figure 5.3 Robust stability domain \mathcal{H}_0 and domains of maximal admissible $\Delta i^* = [5, 8, 10]A$.

5.3.4 Controlled system speed performance requirements

In practice, specification on the system speed are often given as limitations on the step response transient time value. Again, using sampling, 10000 uniform random samples are taken in the controller parameter domain and the regions are plotted such that the transient time τ is so that $\tau \leq T$ $T = [0.2, 0.25, 0.3, 0.45, 0.7]s$.

The transient time is calculated for the largest possible mass value, i.e. $M = 1000kg$, hence, smaller transient time for smaller masses is guaranteed. Figure 5.4 depicts level sets for various transient time. Notice that when we restrict ourselves with $\Delta i < 8A$ it is possible to choose a controller such that transient time remains below 0.25s. When we increase upper bound for Δi to 10A it becomes possible to achieve transient time 0.2s.

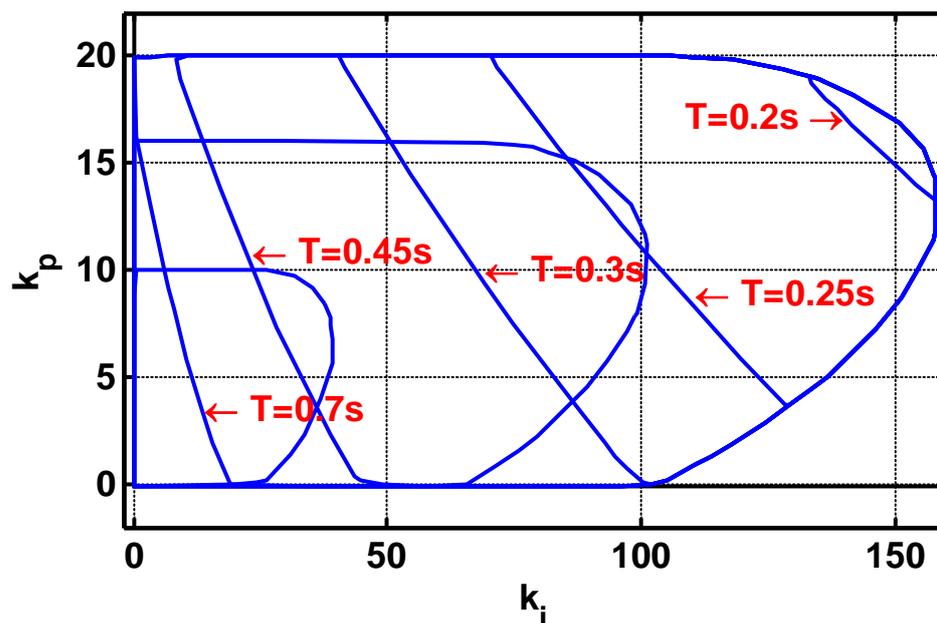


Figure 5.4 Domains of maximal admissible $\Delta i^* = [5, 8, 10]$ A and level sets of transient time for $T = [0.2, 0.25, 0.3, 0.45, 0.7]$ s

5.3.5 H_∞ performance evaluation

Now we want to take into account H_∞ performance. The paragraph 5.1 shows how the domains in controller parameter space such that $\|T(s, k_i, k_p)\|_\infty \leq \gamma$ can be efficiently plotted for given values of γ . Here we apply that result to the AGV plant with the largest mass value $M = 1000$ kg. In Figure 5.5 the controller parameter space is shown with the current limitation values, transient time values and the γ values.

	k_p	k_i
C ₁	13	95
C ₂	16	150

Table 5.1 Selected controller parameters

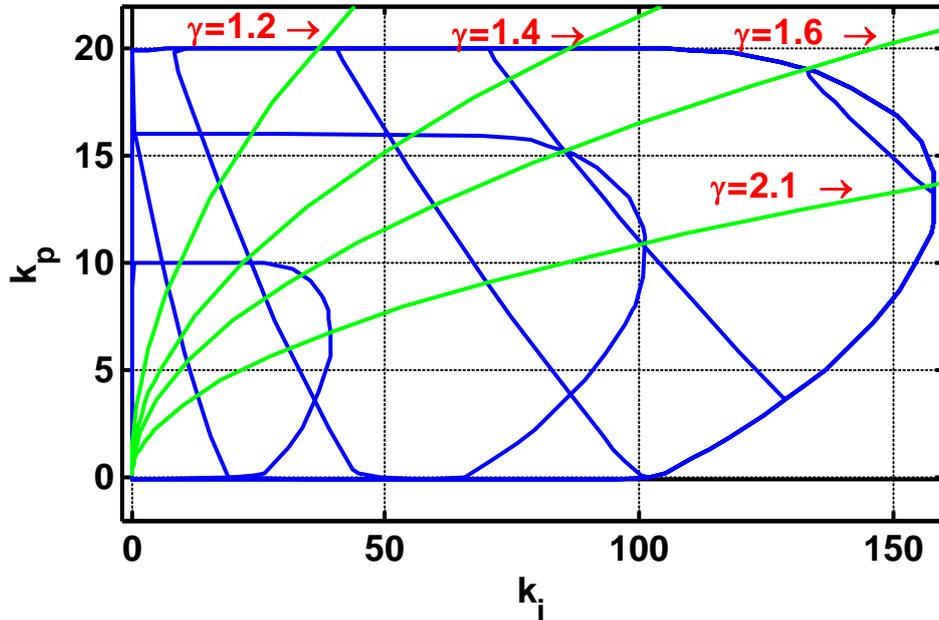


Figure 5.5 Domains of maximal admissible $\Delta i^* = [5, 8, 10]$ A, level sets of transient time and $\|T\|_{\infty} \leq \gamma$ with $\gamma = [1.2, 1.4, 1.6, 2.1]$

5.3.6 Dependence on the mass value of the control current limitation and the transient time speed

All previous analysis is performed for the largest mass value $M = 1000\text{kg}$ in order to obtain a robust solution feasible for the total mass range. In this subsection we are interested in evaluating how conservative this solution is. Namely, we take two controllers, whose values are reported in Table 5.1.

C_1 is chosen among the controllers that guarantees transient time below 0.25s with $\max \Delta i < 8\text{A}$; C_2 is chosen so that transient time is below 0.2s with $\max \Delta i < 10\text{A}$. In Figure 5.6 it is shown the dependence of the current $\max \Delta i$ on the mass M . Notice that for masses smaller than 500kg the current needed is approximately constant, i.e. the mass increase with respect to the empty vehicle's one has no effect on the controller torque requests. In Figure 5.7 it is shown the dependence

of transient time on the mass M . The transient time is monotonic in mass and the difference in transient time is larger for larger mass values.

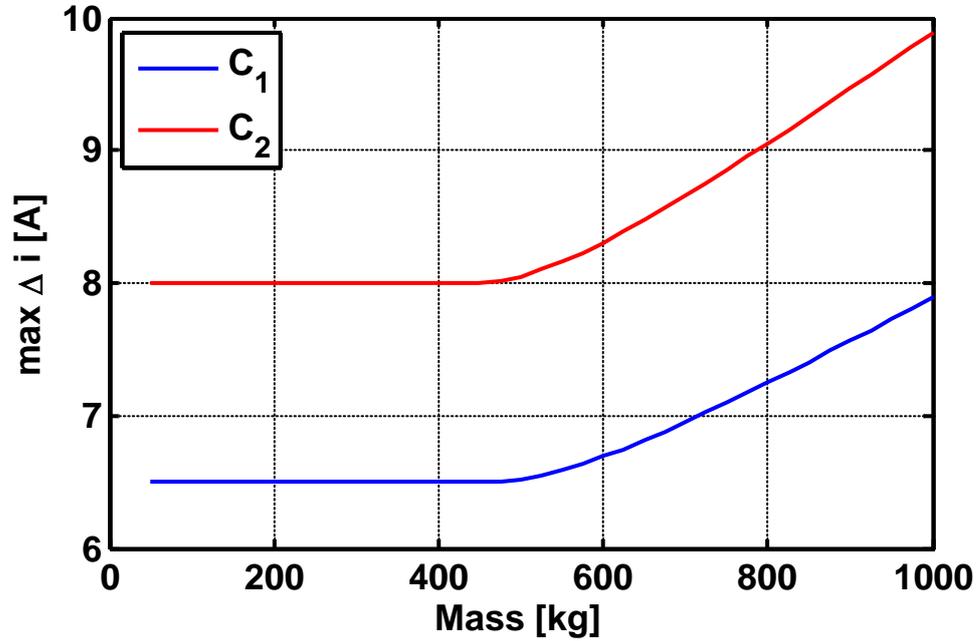


Figure 5.6 Maximum current as a function of M for two controllers with different current limitations (red: 10A, blue: 8A)

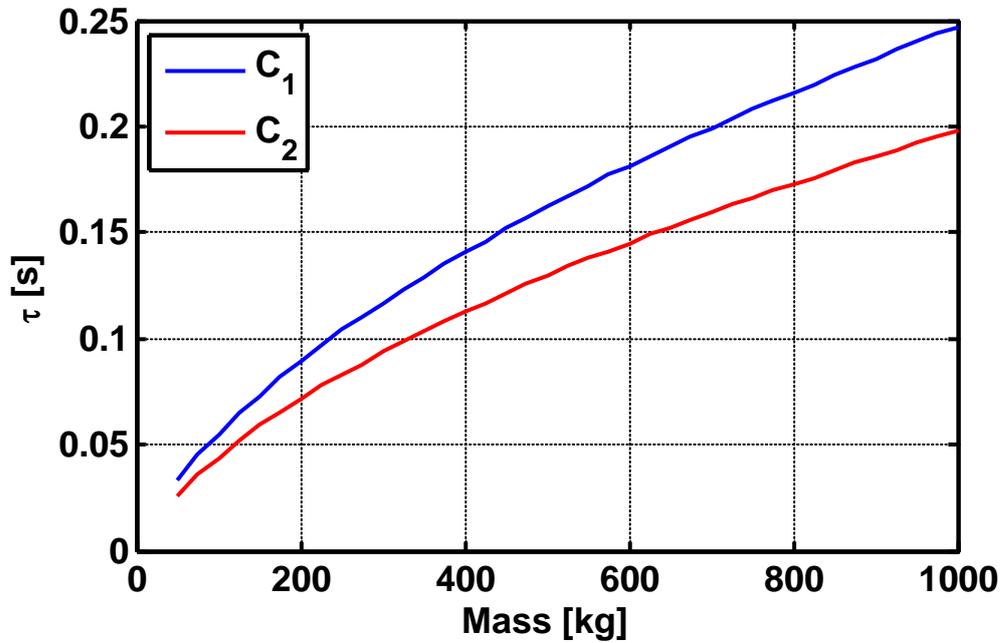


Figure 5.7 Transient time τ as a function of mass for two controllers with different current limitations (red: 10A, blue: 8A)

5.4 PI discretization

All the previous analysis were made in continuous-time, but to utilize these techniques the selected controller has to be implemented in a microcontroller with a defined sampling time. Thus the PI will be discretized with a bilinear transformation adding anti wind-up and reset methods.

The bilinear transformation, named Tustin's transformation, is represented in equation [5.24]

$$s = \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}} \quad [5.24]$$

where T_s is the sampling time.

Thanks to bilinear transformation the stability region of the s-plane, corresponding to the left half-plane $\text{Re}(s) \leq 0$, is mapped on the z-plane as shown in Figure 5.8.

The main advantage of Tustin's approximation is that the left half s-plane is transformed into the unit disc in the z-plane, so a stable continuous-time system will always give a stable discrete-time system.

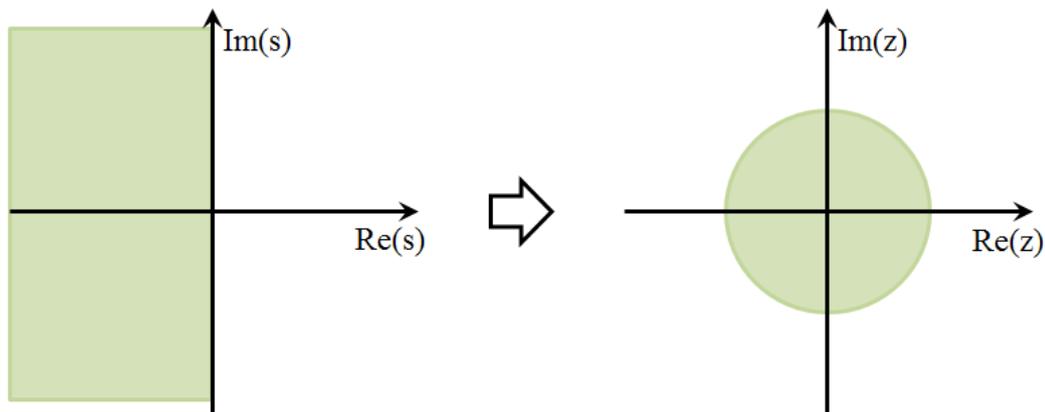


Figure 5.8 Mapping of stability region with bilinear transformation

Applying the bilinear transformation, described in equation [5.24], to equation [5.15] and forcing $k_d = 0$ it's possible to obtain the PI controller discretization as follow

$$C(z) = k_p + k_i \frac{1+z^{-1}}{1-z^{-1}} \frac{T_s}{2} \quad [5.25]$$

That written in recursively is

$$u(t) = k_p (e(t) - e(t-1)) + k_i \frac{T_s}{2} (e(t) + e(t-1)) + u(t-1) \quad [5.26]$$

As can be seen in Figure 5.9 the implementation of a PI controller with reset and anti-windup in Matlab /Simulink is very simple.

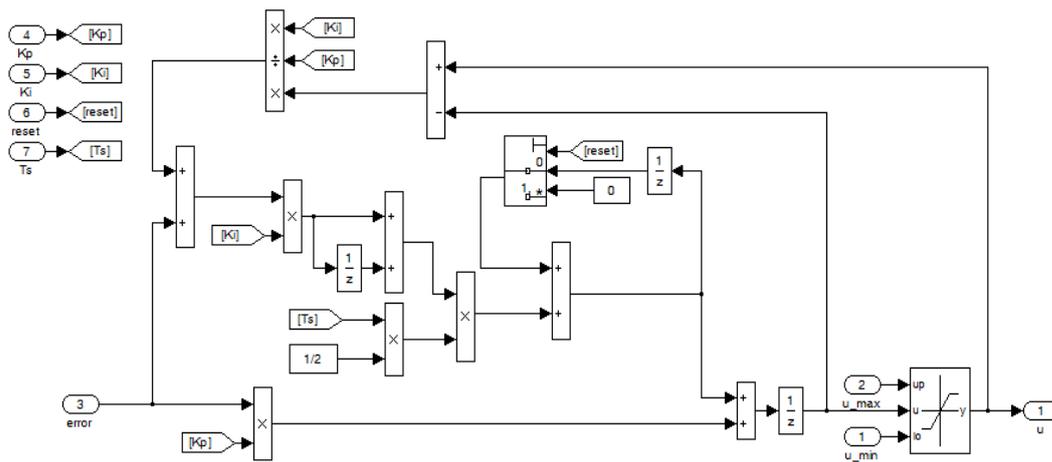


Figure 5.9 PI implementation in Matlab/Simulink

5.5 Concluding remarks

In this chapter is designed the control loop for steering speed $\Delta\omega$ in a robust way in order to make the AGV unaffected by mass variation. This has been done through a particular procedure like the Hit & Run and the parameter space analysis in H_∞ .

Thanks to these technique a PI has been tuned taking into account control action limitations, due to the limited current supply values, system speed specifications, defined by desired values for settling time, and H_∞ performances.

Chapter 6

Current control of brushless motor with open-end windings

As shown in Figure 5.1, there are two nested loops, the inner loop is meant to control the current of a brushless motor and the outer is meant to control the steering speed and the longitudinal speed. To complete the control of the AGV it's necessary to close these loops. The brushless motors have a particular feature: open-end windings and to control this particular kind of motors it's necessary to implement a new space vector modulation described in the following paragraphs.

6.1 Open-end windings brushless motor drive

Multilevel inverters are preferred in high power medium-voltage energy control applications, due to their high voltage capability, reduced Common Mode Voltages, nearly sinusoidal output voltage and power quality associated with reduced harmonic distortion and lower electromagnetic interference (EMI). They can increase output voltage magnitude and reduce output voltage and current harmonic content, switching frequency and voltage supported by each power

semiconductor. The two popular techniques for the generation of PWM in multilevel inverters are sine-triangle PWM (SPWM) and Space Vector PWM (SVPWM). SPWM involves the comparison of a reference signal with a number of level shifted carriers to generate the PWM signal. SVPWM involves synthesizing the reference voltage space vector by switching among the three nearest voltage space vectors. Several SVPWM techniques have been discussed extensively in literature [11], [12] and [13]. The PWM techniques complexity and computational cost increase with the number of levels of the inverter.

The schematic diagram of a conventional two-level voltage source inverter driven brushless motor drive is shown in Figure 6.2. Figure 6.1 shows the open-end winding brushless motor drive configuration.

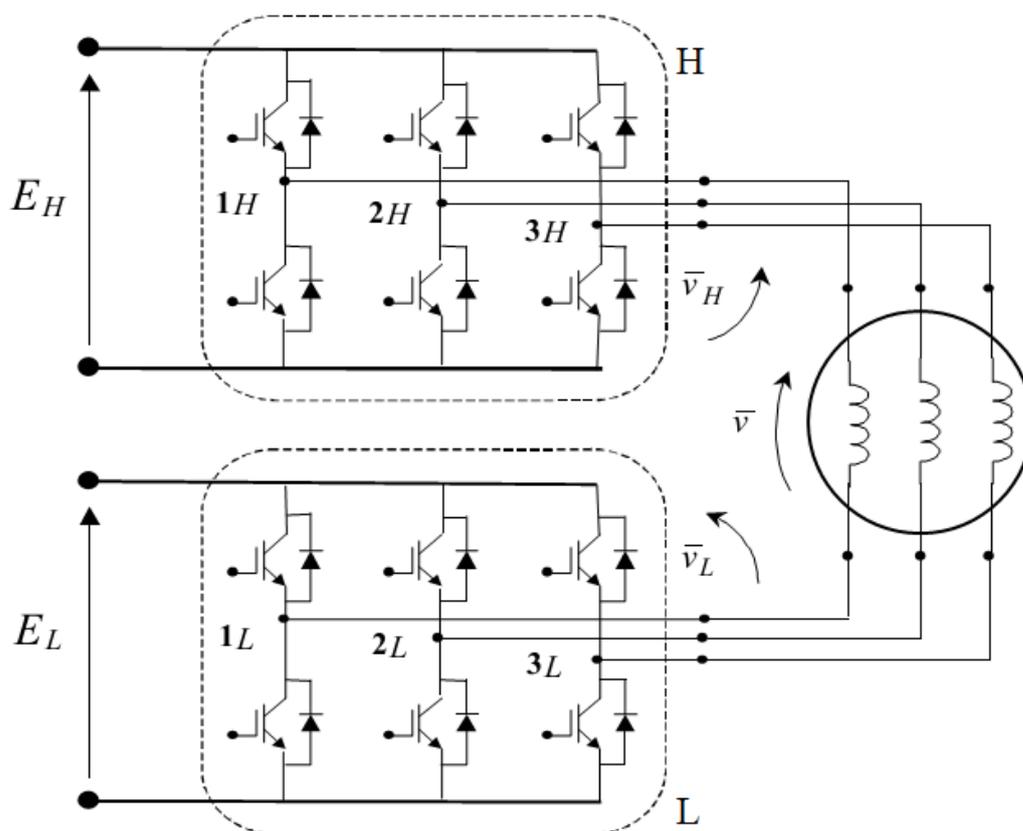


Figure 6.1 Electric scheme of a dual level dual inverter for a brushless motor

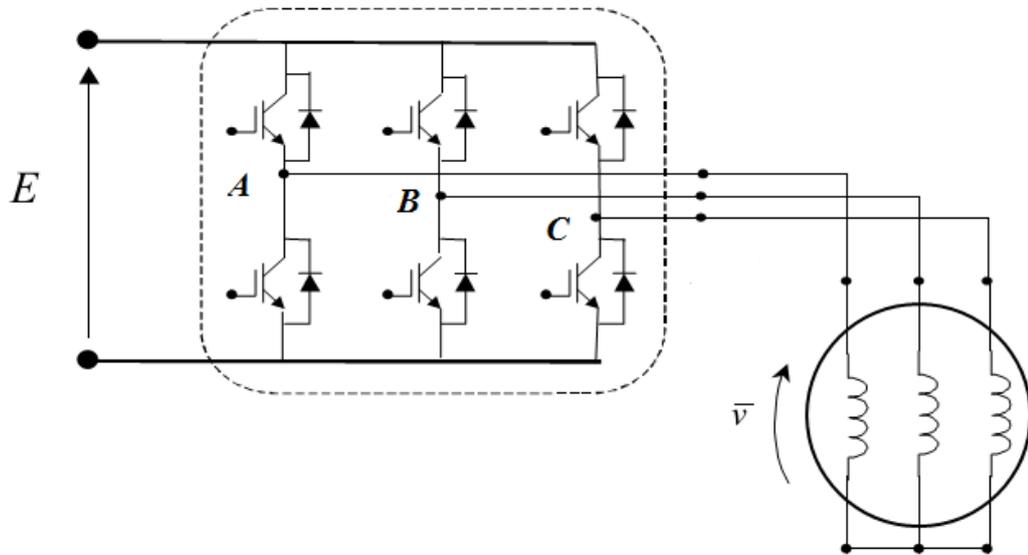


Figure 6.2 Electric scheme of a conventional inverter for a brushless motor

The open-end winding brushless motor drive is obtained by opening the neutral point of the star connected stator windings of a conventional three-phase brushless motor and feeding the motor from both end using inverters. Similar to the conventional brushless motor drives, this drive is suitable for high-power applications.

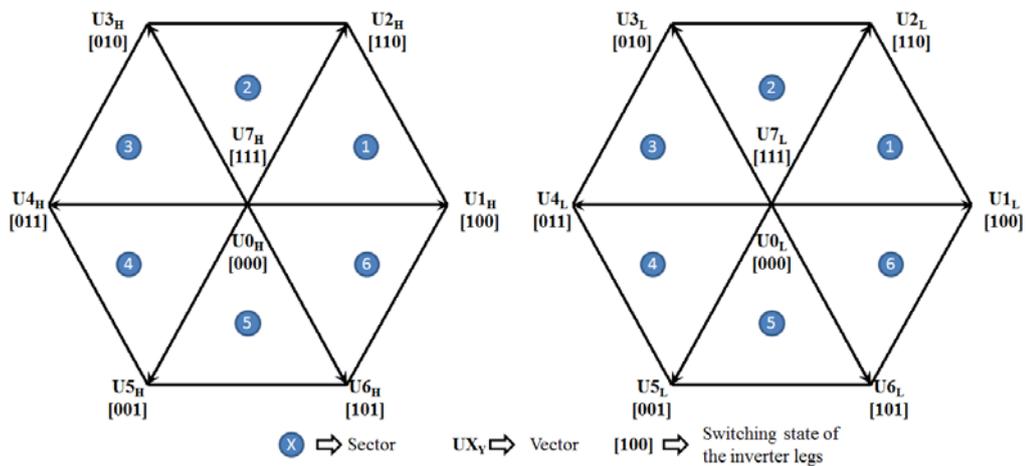


Figure 6.3 Basic switching state (vectors) of inverter 1 (High) and inverter 2 (Low)

Top and bottom switches, see Figure 6.2 , are working in a complementary mode; for example if the top switch I_T is on, then the corresponding bottom switch I_B is

off and vice versa. Considering that the value 1 is assigned to the on state of the top switch, and value 0 is assigned to the on state of the bottom switch, the switching vector $[A, B, C]^T$ can be defined. Creating such a vector allows numerical definition of all possible switching states. The basic switching states for individual inverters are shown in Figure 6.3.

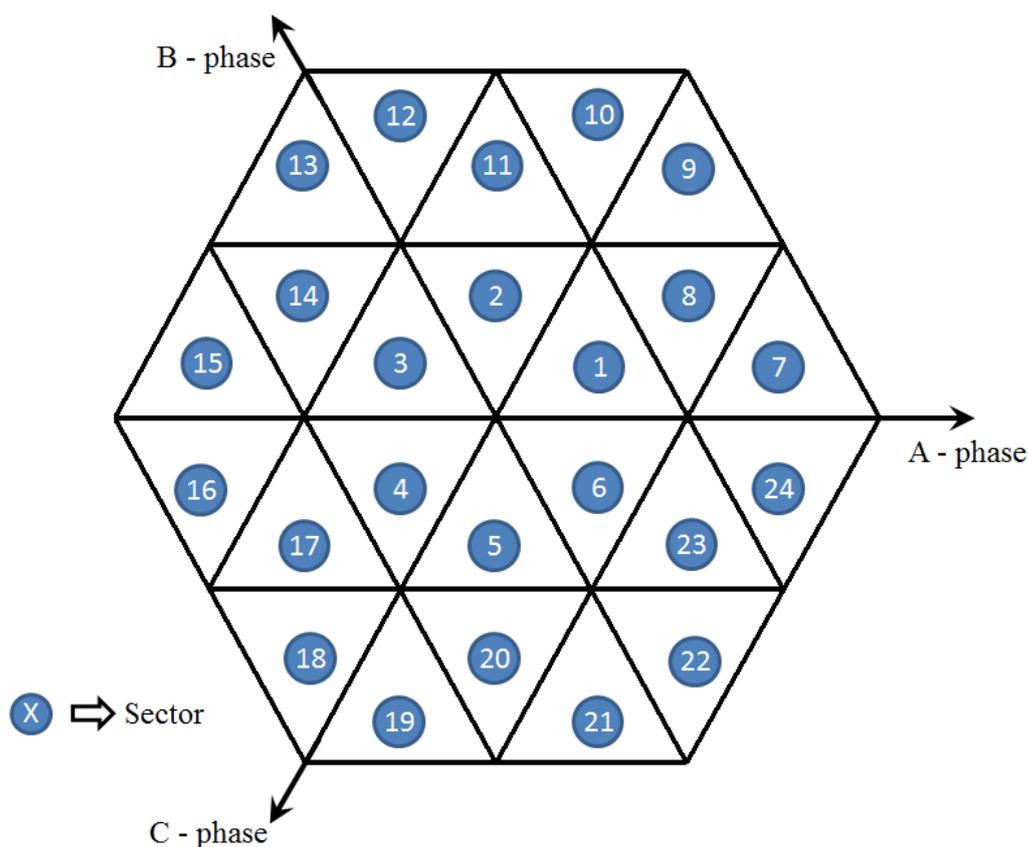


Figure 6.4 Resultant sectors in the two level two inverter scheme

The combination of the eight switch configurations for each two level inverter yields 64 possible switches states for the whole multilevel converter, corresponding to 18 different output voltage vectors and a null vector and 24 sectors as represented in Figure 6.4 and Figure 6.5. By using the SVM technique, these voltage vectors can be combined to obtain any output voltage vector lying inside the outer hexagon. To understand the SVM technique for two level two inverter it's necessary to have a basic understanding of SVM standard for motor

with star connection, but first of all it's very important to understand how the current control will be implemented.

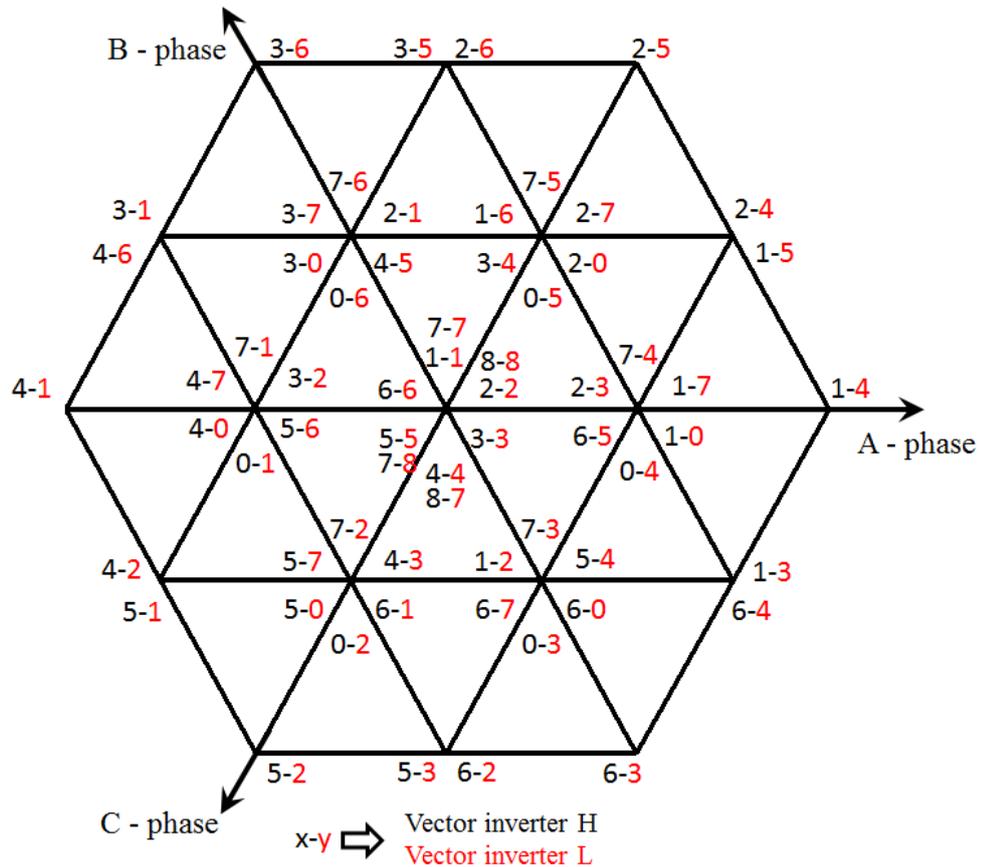


Figure 6.5 Resultant space vector combination in the two level two inverter scheme

6.2 Basis of current control

The model used for current control design can be understood by using space vector theory. The 3-phase motor quantities (such as voltages, currents, magnetic flux, etc.) are expressed in terms of complex space vectors. Such a model is valid for any instantaneous variation of voltage and current and adequately describes the performance of the machine under both steady-state and transient operation.

The complex space vectors can be described using only two orthogonal axes, α and β , so the motor can be considered a two-phase machine. Using a 2-phase motor model reduces the number of equations and simplifies the control design. The transformation from 3-phase motor quantities to 2-phase motor quantities is very simple, its name is Clarke transform as represented in equation [6.1]

$$X_{\alpha\beta\gamma} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad [6.1]$$

The inverse Clarke transform is represented in equation [6.2]

$$X_{abc} = \frac{2}{3} \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \begin{bmatrix} X_\alpha \\ X_\beta \\ X_\gamma \end{bmatrix} \quad [6.2]$$

If the system is balanced, as for a motor with star connection, $X_a + X_b + X_c = 0$ and thus $X_\gamma = 0$ the transform simplifies to

$$X_{\alpha\beta} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \end{bmatrix} \quad X_{abc} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} \quad [6.3]$$

Besides the stationary reference frame attached to the stator, motor model equations can be formulated in a general reference frame, which rotates at a general speed, ω_g . This transformation, named Park transform, is used to

transform from the α - β stationary orthogonal coordinate system to the d-q rotating orthogonal coordinate system as represented in equation [6.4]

$$X_{dq} = \begin{bmatrix} \cos(\vartheta) & \sin(\vartheta) \\ \cos(\vartheta) & -\sin(\vartheta) \end{bmatrix} \begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} \quad X_{\alpha\beta} = \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) \\ \sin(\vartheta) & \cos(\vartheta) \end{bmatrix} \begin{bmatrix} X_d \\ X_q \end{bmatrix} \quad [6.4]$$

It's possible to pass from 3-phase motor quantities to 2-phase rotating orthogonal coordinate system using the combination of Clarke and Park transformation as represented in equation [6.5], but for simplicity are used the single transformations.

$$X_{dq0} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\vartheta) & \cos\left(\vartheta - \frac{2}{3}\pi\right) & \cos\left(\vartheta + \frac{2}{3}\pi\right) \\ -\sin(\vartheta) & -\sin\left(\vartheta - \frac{2}{3}\pi\right) & -\sin\left(\vartheta + \frac{2}{3}\pi\right) \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad [6.5]$$

$$X_{abc} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & \frac{\sqrt{2}}{2} \\ \cos\left(\vartheta - \frac{2}{3}\pi\right) & -\sin\left(\vartheta - \frac{2}{3}\pi\right) & \frac{\sqrt{2}}{2} \\ \cos\left(\vartheta + \frac{2}{3}\pi\right) & -\sin\left(\vartheta + \frac{2}{3}\pi\right) & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} X_d \\ X_q \\ X_0 \end{bmatrix}$$

Thanks to these transforms the current control is very simple because it's enough to feed back the direct and quadrature currents, I_d and I_q respectively, to reach good performance. The references for these currents are very simple to find, in fact the direct current I_d has always to remain around 0 such that the torque generated by the quadrature current I_q is maximum thus obtaining the maximum efficiency. Whereas the reference for I_q is the profile that the user wants to follow. Afterward the calculation of the controllers it's necessary to return to a 2-phase quantities through the inverse Park transform and then calculate the correct duty-cycle ratios, which are needed for generating the given stator reference voltage vector. This task is performed by Space Vector Modulation, SVM, explained in the next paragraph.

To conclude the scheme represented in Figure 6.6 shows the currents control loops.

Usually the controllers are a simple PI and are calibrated on a single phase model of the motor, generally described by equation [6.6].

$$G(s) = \frac{1}{Ls + R} \quad [6.6]$$

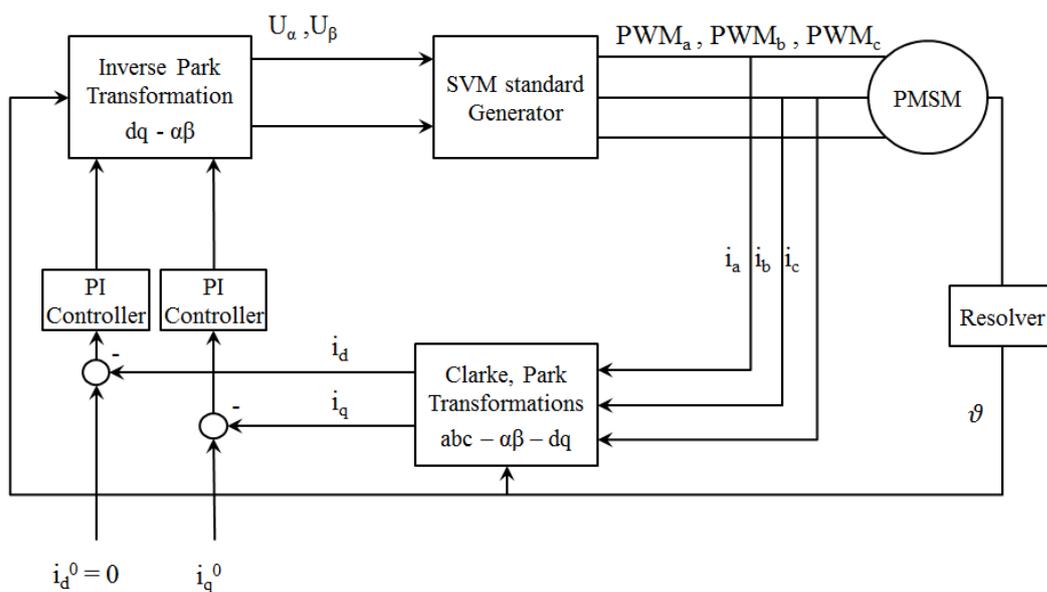


Figure 6.6 Current control scheme for a star connected motor

6.3 SVM standard

The principle of the standard Space Vector Modulation (SVM), see [14] and [15], resides in applying the appropriate switching states for a certain time and thus generating a voltage vector identical to the reference one, see Figure 6.7.

Referring to that principle, an objective of the standard space vector modulation is an approximation of the reference stator voltage vector U_S with an appropriate combination of the switching patterns, composed of basic space vectors. The graphical explanation of this objective is shown in Figure 6.8.

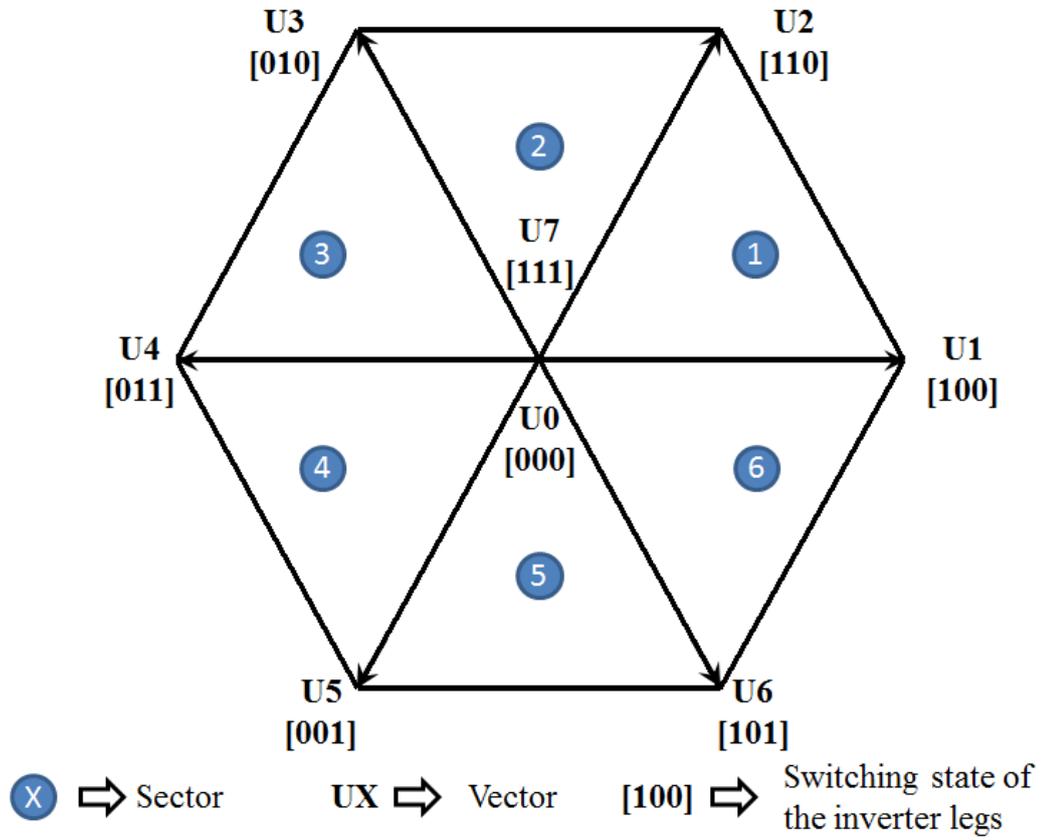


Figure 6.7 Basic space vectors

The stator-reference voltage vector U_s is phase-advanced by 30° from the direct- α , and thus might be generated with an appropriate combination of the adjacent basic switching states $U1$ and $U2$. This figure also indicates resultant direct- α and quadrature- β components for space vectors $U1$ and $U2$.

In this case, the reference-stator voltage vector U_s is located in sector I and, as previously mentioned, can be generated with the appropriate duty-cycle ratios of the basic switching states $U2$ and $U1$. The principal equations concerning this vector location are:

$$T = T_2 + T_1 + T_{null} \quad [6.7]$$

$$U_s = \frac{T_2}{T} U_2 + \frac{T_1}{T} U_1 \quad [6.8]$$

where T_2 and T_1 are the respective duty-cycle ratios, for which the basic space

vectors U_2 and U_1 should be applied within the time period T . T_{null} is the course of time, for which the null vectors U_7 and U_0 are applied. Those duty-cycle ratios can be calculated using the following equations:

$$\begin{aligned}
 U_\beta &= \frac{T_2}{T} |U_2| \sin 60^\circ \\
 U_\alpha &= \frac{T_1}{T} |U_1| \frac{U_\beta}{\tan 60^\circ}
 \end{aligned}
 \tag{6.9}$$

Considering that normalized magnitudes of basic space vectors are $|U_2| = |U_1| = 2/\sqrt{3}$, and by substitution of the trigonometric expressions $\sin 60^\circ$ and $\tan 60^\circ$ by their quantities $\sqrt{3}/2$ and $\sqrt{3}$, respectively, equation [6.9] can be rearranged for the unknown duty-cycle ratios T_2/T and T_1/T as follows:

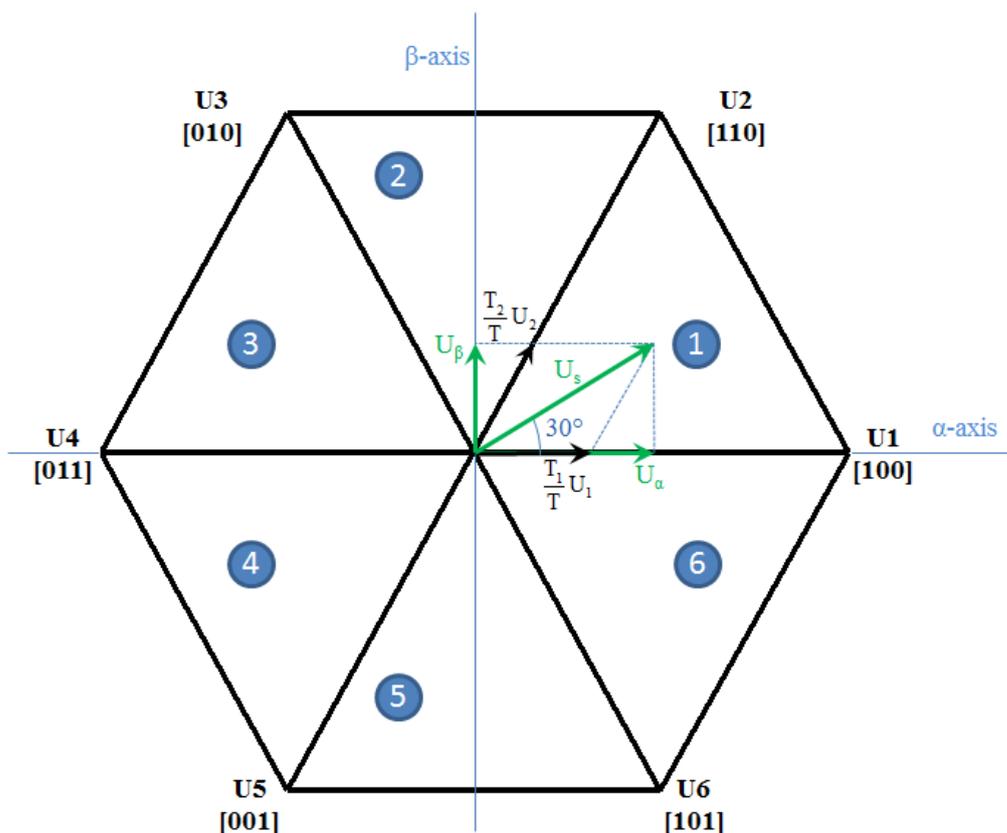


Figure 6.8 Projection of reference voltage vector in sector I

$$U_{\beta} = \frac{T_2}{T}$$

$$U_{\alpha} = \frac{T_1}{T} \frac{2}{\sqrt{3}} + \frac{T_2}{T} \frac{1}{\sqrt{3}}$$
[6.10]

Sector II is depicted in Figure 6.9. In this particular case, the reference-stator voltage vector U_s is generated by the appropriate duty-cycle ratios of the basic switching states U_2 and U_3 . The basic equations describing this sector are:

$$T = T_3 + T_2 + T_{null}$$
[6.11]

$$U_s = \frac{T_3}{T} U_3 + \frac{T_2}{T} U_2$$
[6.12]

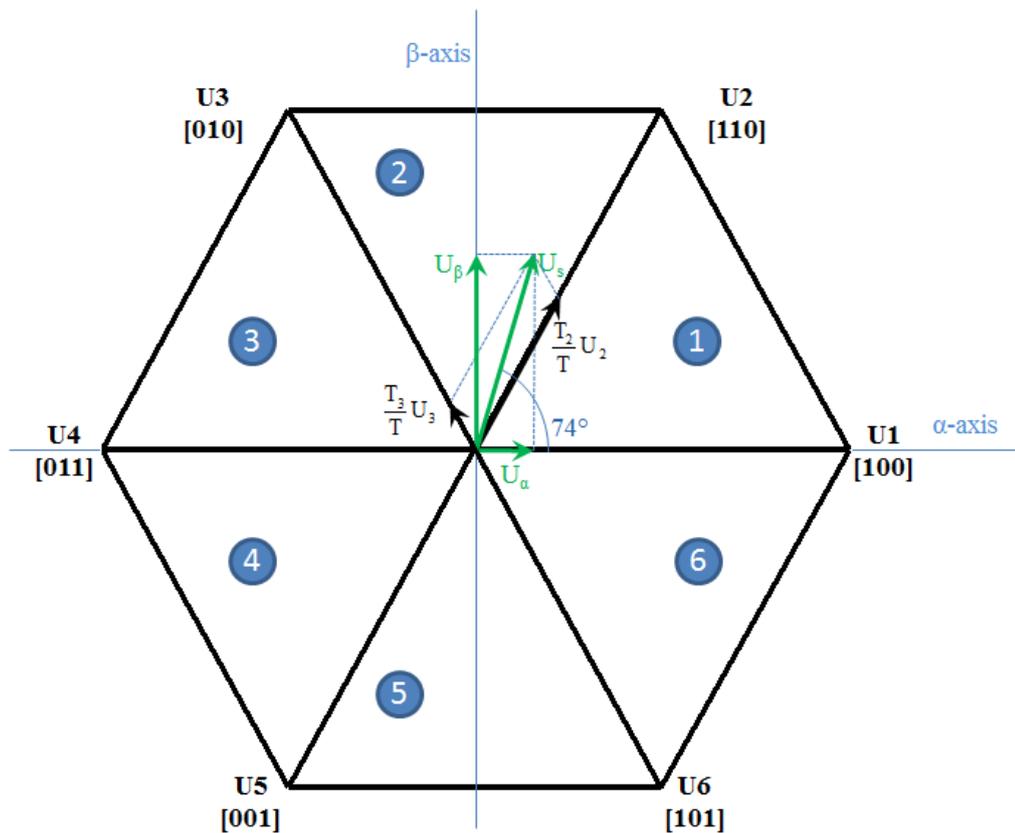


Figure 6.9 Projection of the reference voltage vector in sector II

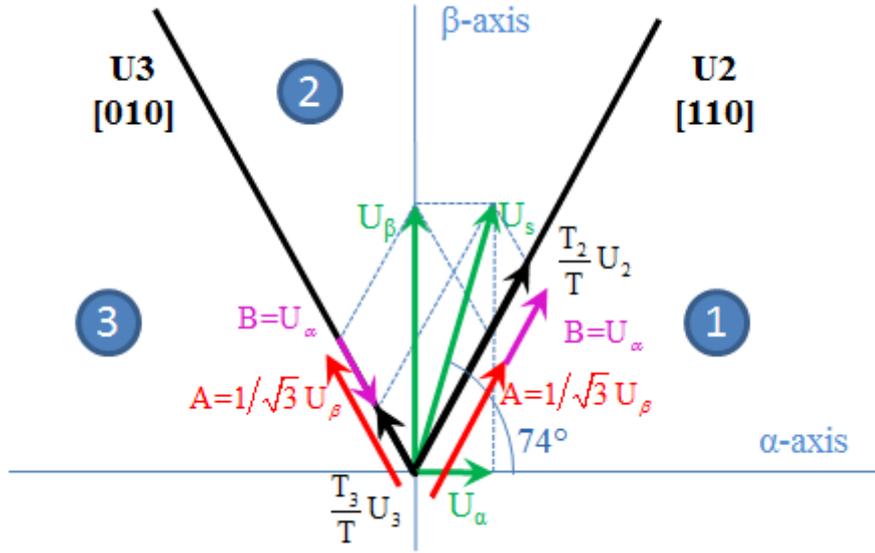


Figure 6.10 Detail of the voltage vector projection in sector II

where T_3 and T_2 are the respective duty-cycle ratios, for which the basic space vectors U_3 and U_2 should be applied within the time period T . T_{null} is the course of time, for which the null vectors U_0 and U_7 are applied. These resultant duty-cycle ratios are formed from the auxiliary components termed A and B . The graphical representation of the auxiliary components is shown in Figure 6.10

The equations describing those auxiliary time-duration components are

$$\frac{\sin 30^\circ}{\sin 120^\circ} = \frac{A}{U_\beta} \quad [6.13]$$

$$\frac{\sin 60^\circ}{\sin 60^\circ} = \frac{B}{U_\alpha}$$

Equation [6.13] has been formed using the sine rule and can be rearranged for the calculation of the auxiliary time-duration components A and B . This is done simply by substituting the trigonometric terms $\sin 30^\circ$, $\sin 120^\circ$, and $\sin 60^\circ$, by their numerical representations $1/2$, $\sqrt{3}/2$, and $\sqrt{3}/2$, respectively.

$$A = \frac{1}{\sqrt{3}} U_\beta \quad [6.14]$$

$$B = U_\alpha$$

The resultant duty-cycle ratios, T_3/T and T_2/T , are then expressed in terms of the auxiliary time-duration components defined by equation.

$$\begin{aligned}\frac{T_3}{T}|U_3| &= (A - B) \\ \frac{T_2}{T}|U_2| &= (A + B)\end{aligned}\tag{6.15}$$

With the help of these equations, and also considering the normalized magnitudes of the basic space vectors to be $|U_3| = |U_2| = 2/\sqrt{3}$, the equations expressed for the unknown duty-cycle ratios of basic space vectors T_3/T and T_2/T can be written as follows:

$$\begin{aligned}\frac{T_3}{T} &= \frac{1}{2}(U_\beta - \sqrt{3}U_\alpha) \\ \frac{T_2}{T} &= \frac{1}{2}(U_\beta + \sqrt{3}U_\alpha)\end{aligned}\tag{6.16}$$

The duty-cycle ratios in the remaining sectors can be derived using the same approach. The resulting equations will be similar to those derived for sector I and sector II.

To depict the duty-cycle ratios of the basic space vectors for all sectors, we define:

- Three auxiliary variables:

$$X = U_\beta$$

$$Y = 1/2 (U_\beta + \sqrt{3}U_\alpha)$$

$$Z = 1/2 (U_\beta - \sqrt{3}U_\alpha)$$

- Two expressions:

$$t_1$$

$$t_2$$

which generally represent the duty-cycle ratios of the basic space vectors in the respective sector; for example, for the first sector, t_1 and t_2 represent duty-cycle ratios of the basic space vectors U_2 and U_1 ; for the second sector, t_1 and t_2 represent duty-cycle ratios of the basic space vectors U_3 and U_2 , and so on, see Table 6.1.

Sectors	U1,U2	U2,U3	U3,U4	U4,U5	U5,U6	U6,U1
t_1	X	Y	-Y	Z	-Z	-X
t_2	-Z	Z	X	-X	-Y	Y

Table 6.1 Determination of t_1 and t_2

For the determination of auxiliary variables X , Y and Z , the sector number is required. This information can be obtained through several approaches. An approach discussed here requires the use of modified inverse Clarke transformation to transform the direct- α and quadrature- β components into a balanced three-phase quantity u_{ref1} , u_{ref2} and u_{ref3} , see equation [6.17], used for straightforward calculation of the sector number, to be shown later.

$$\begin{aligned}
 u_{ref1} &= U_{\beta} \\
 u_{ref2} &= \frac{-U_{\beta} + \sqrt{3}U_{\alpha}}{2} \\
 u_{ref3} &= \frac{-U_{\beta} - \sqrt{3}U_{\alpha}}{2}
 \end{aligned} \tag{6.17}$$

The modified inverse Clarke transformation projects the quadrature- U_{β} component into u_{ref1} , as shown in Figure 6.11 and Figure 6.12, whereas voltages generated by the conventional inverse Clarke transformation project the direct- U_{α} component into u_{ref1} .

Figure 6.11 depicts the direct- U_{α} and quadrature- U_{β} components of the stator reference voltage vector U_S that were calculated by the equations $U_{\alpha} = \cos \vartheta$ and $U_{\beta} = \sin \vartheta$, respectively.

The sector identification tree, shown in Figure 6.13, can be a numerical solution of the approach shown in Figure 6.12.

It should be pointed out that in the worst case three simple comparisons are required to precisely identify the sector of the stator-reference voltage vector. For example, if the stator reference voltage vector resides according to the one shown in Figure 6.8, the stator-reference voltage vector is phase-advanced by 30° from the direct α -axis, which results in the positive quantities of u_{ref1} and u_{ref2} and the

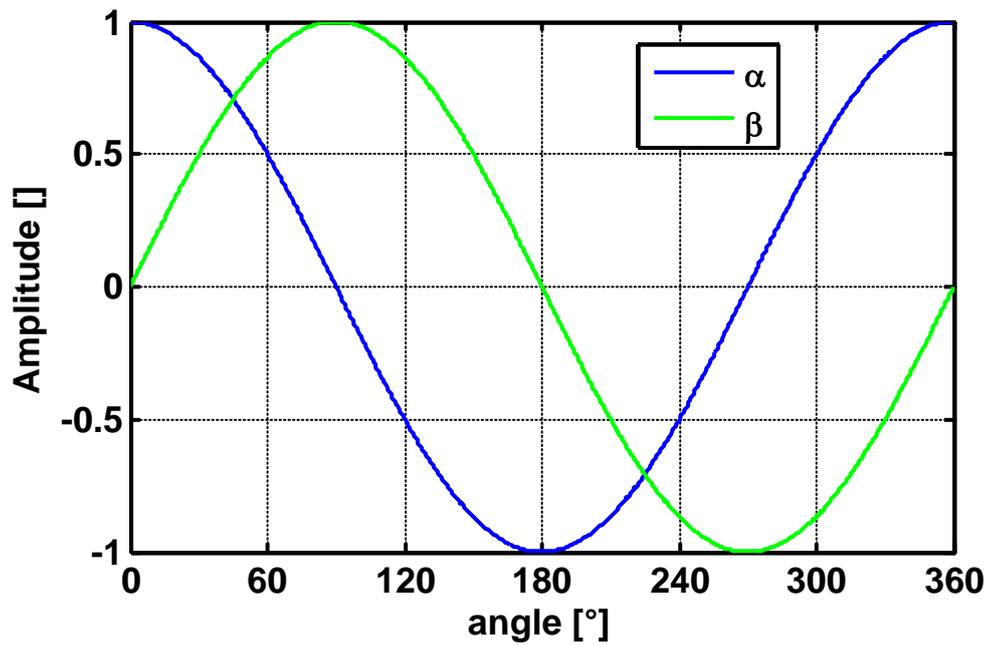


Figure 6.11 Direct- U_α and quadrature- U_β components of stator reference voltage

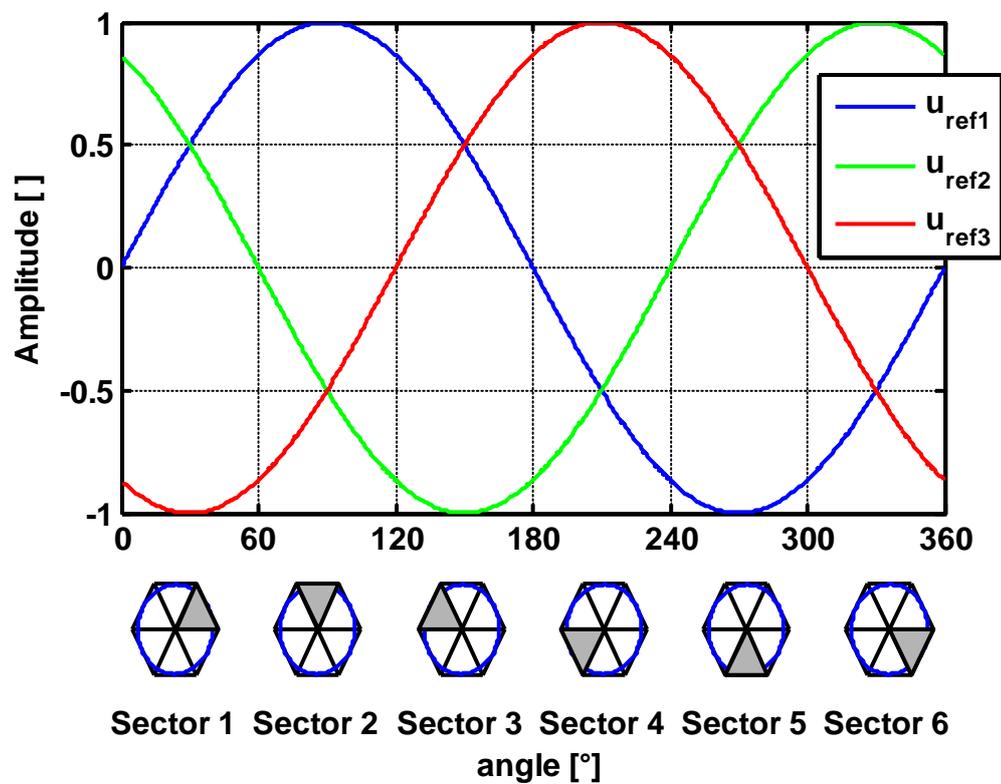


Figure 6.12 Reference voltages u_{ref1} , u_{ref2} and u_{ref3}

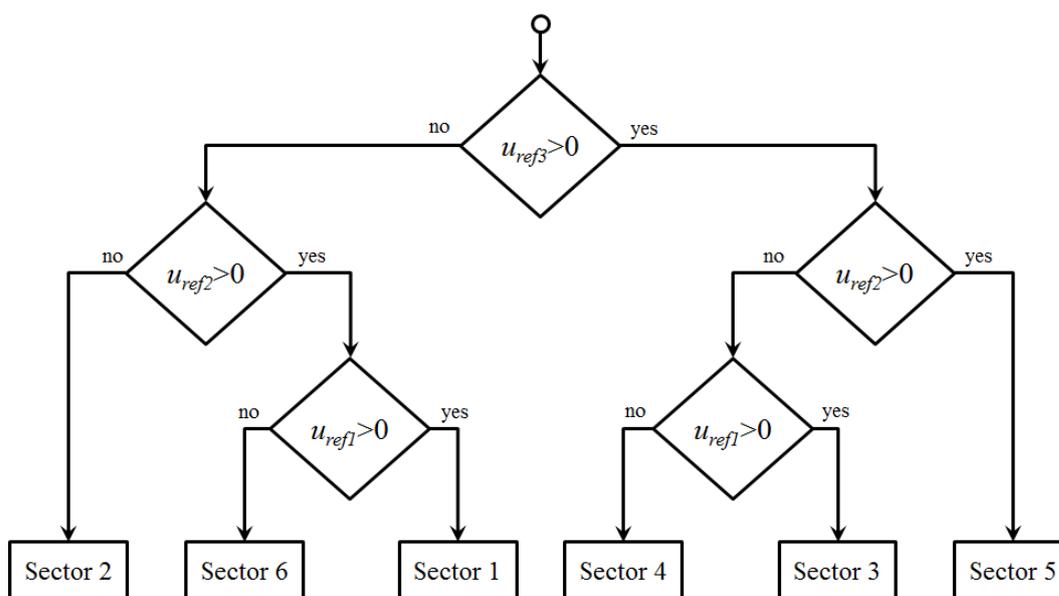


Figure 6.13 Identification of sector number

negative quantity of u_{ref3} ; refer to Figure 6.12 and Figure 6.13. If these quantities are used as the inputs to the sector identification tree, the product of those comparisons will be sector I. Using the same approach identifies the sector II, if the stator-reference voltage vector is located according to the one shown in Figure 6.10. The variables t_1 , t_2 , and t_3 , representing switching duty-cycle ratios of the respective three-phase system, are given by the following equations:

$$\begin{aligned}
 t_1 &= \frac{T - t_{-1} - t_{-2}}{2} \\
 t_2 &= t_1 + t_{-1} \\
 t_3 &= t_2 + t_{-2}
 \end{aligned}
 \tag{6.18}$$

where T is the switching period, t_1 , t_2 and t_3 are the duty-cycle ratios of the basic space vectors, given for the respective sector; see Table 6.1. Equation [6.18], is specific solely to the standard space vector modulation technique; consequently, other space vector modulation techniques will require deriving different equations.

The next step is to assign the correct duty-cycle ratios, t_1 , t_2 , and t_3 , to the respective motor phases. This is a simple task, accomplished in a view of the

position of the stator-reference voltage vector; see Table 6.2.

	$U1,U2$	$U2,U3$	$U3,U4$	$U4,U5$	$U5,U6$	$U6,U1$
pwm_a	t_3	t_2	t_1	t_1	t_2	t_3
pwm_b	t_2	t_3	t_3	t_2	t_1	t_1
pwm_c	t_1	t_1	t_2	t_3	t_3	t_2

Table 6.2 Assignment of the duty-cycle ratios to motor phases

The principle of the space vector modulation technique consists of applying the basic voltage vectors U_X for the certain time in such a way that the mean vector, generated by the pulse width modulation approach for the period T , is equal to the original stator-reference voltage vector U_S . This provides a great variability of the arrangement of the basic vectors during the PWM period T . Those vectors might be arranged either to lower the switching losses or to achieve different results, such as center-aligned PWM, edge-aligned PWM, or a minimal number of switching states. A brief discussion of the widely-used center-aligned PWM follows.

The generation of the center-aligned PWM pattern is accomplished practically by comparing the threshold levels, pwm_a , pwm_b , and pwm_c , with a free-running up-down counter. The timer counts to 1 and then down to 0. It is supposed that when a threshold level is larger than the timer value, the respective PWM output is active. Otherwise, it is inactive; see Figure 6.14

Figure 6.15 graphically shows the calculated waveforms of duty-cycle ratios using standard space vector modulation.

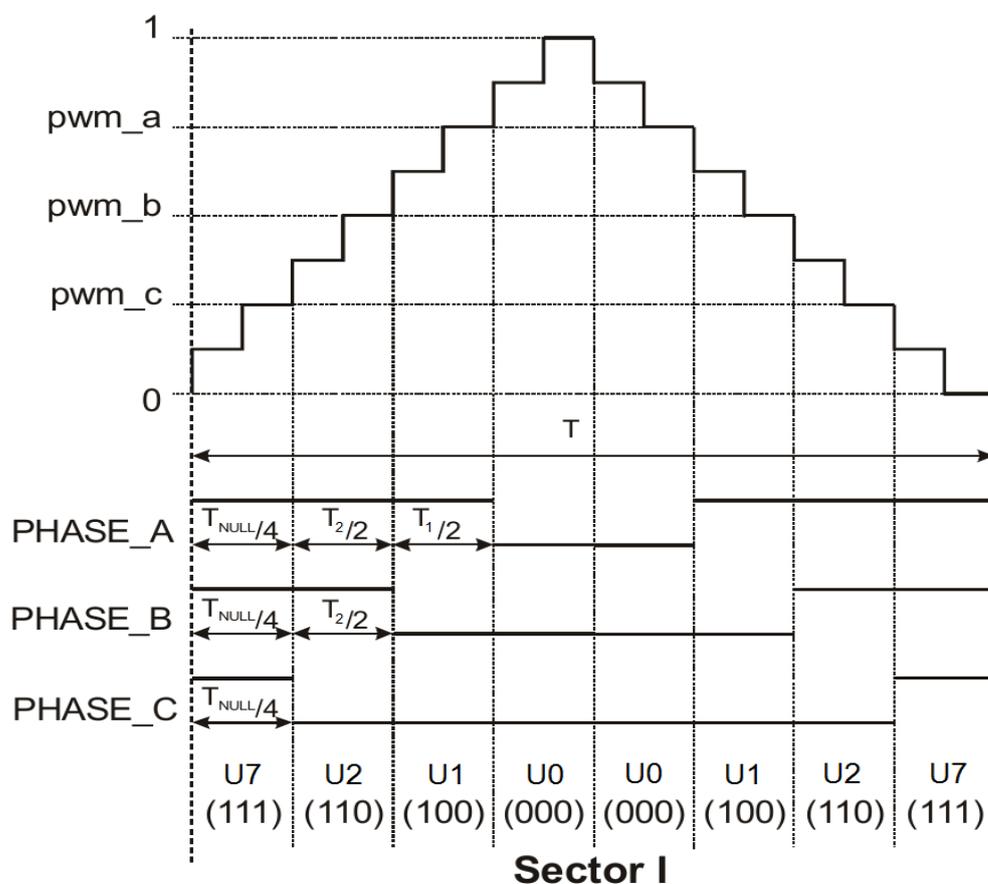


Figure 6.14 Standard space vector modulation technique, center-aligned PWM

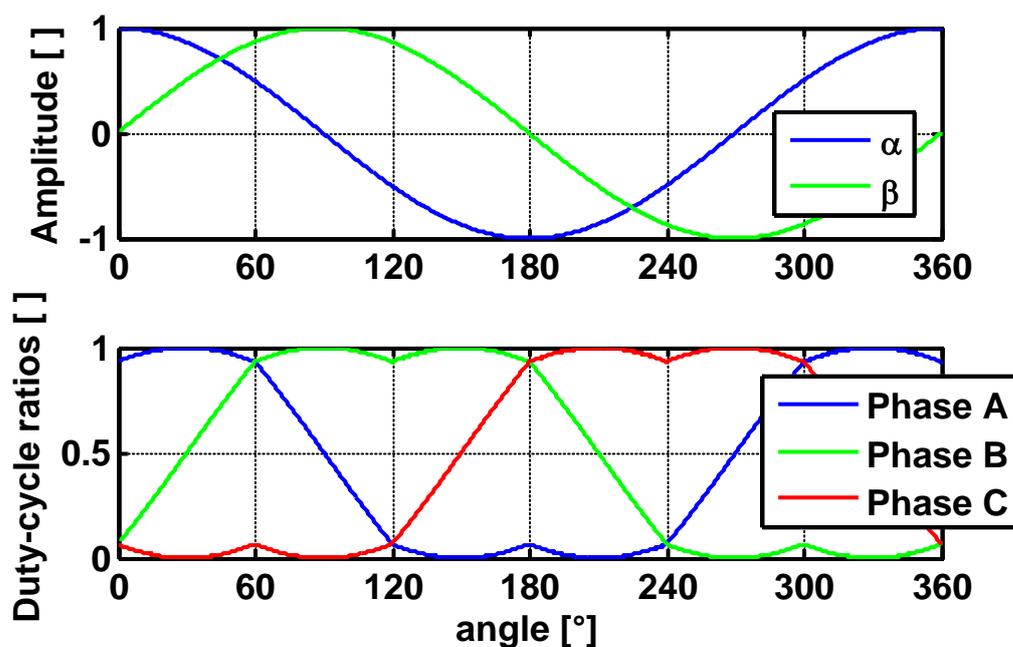


Figure 6.15 Standard space vector modulation technique

6.4 SVM and current loop for brushless motors with open-end windings

As can be seen in Figure 6.1, there are two inverters that have to be commanded simultaneously. To achieve this purpose it's necessary to study in depth Figure 6.5 and Figure 6.7. It's possible to notice that the hexagon of the resultant space vectors combination can be created using only some combinations and not all of them, see Figure 6.5, hence this work will focus on simplifying the number of combinations and use them efficiently.

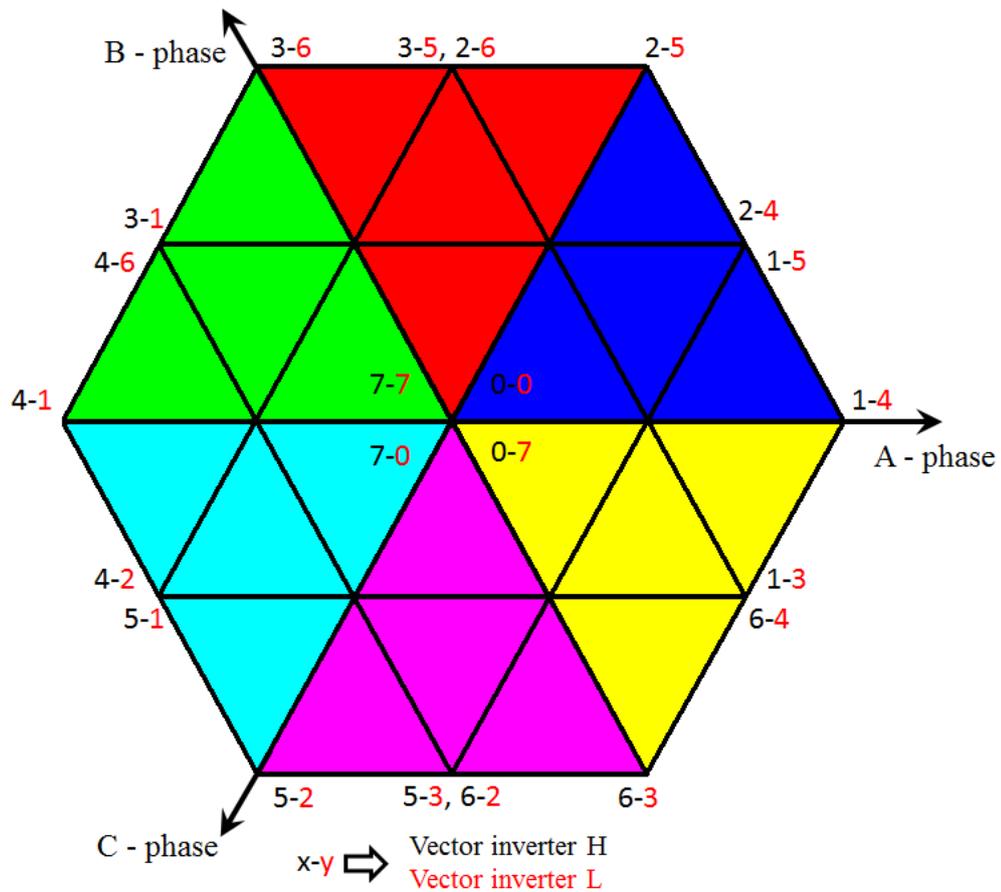


Figure 6.16 Simplification of resultant space vector combination in the two level two inverter scheme and macro sectors with different colors

Using this approach it's possible to reuse the SVM standard studied for motors with star connection for each inverters.

The sectors 1, 7, 8 and 9 are contained by four combinations $\{1,4\}$, $\{2,4\}$, $\{1,5\}$ and $\{2,5\}$, and as can be shown in paragraph 6.3 these sector can be seen as a macro sector 1 where for the inverter H the active space vectors are $U1$ and $U2$ while for inverter L the active space vectors are $U4$ and $U5$ that are the opposite space vectors with respect to the ones of inverter 1, see Figure 6.7. The combinations for the other macro sectors are very simple to deduce, see Figure 6.16.

Therefore to apply these couple of space vectors it's necessary to make sure that inverter L computes always the opposite space vectors of inverter H . To do this is sufficient to take SVM standard but feed it with negated signals, so when the SVM standard of inverter H calculates the PWMs for sector I the inverter L calculates the PWMs for sector IV.

In this way the generation of PWMs is very simple and it's possible to reuse the SVM standard. To control efficiently the current it's necessary to use the completed Clarke and Park transform because the system is not balanced. In Figure 6.17 can be seen the current loop scheme for the motors with open-end windings.

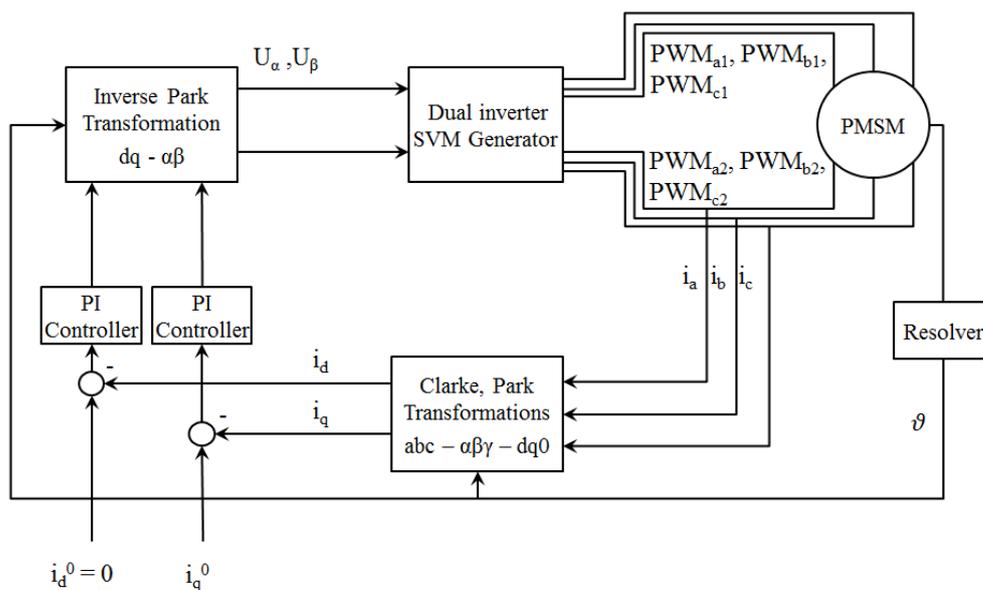


Figure 6.17 Current loop for motors with open-end windings

6.5 Experimental results

After the closing of current loop for motors with open-end windings, see Figure 6.17, some tests have been done to verify the quality of the performance. To do this a sinusoidal reference feeds the current control loop. The sinusoidal reference has a high frequency like 25Hz in order to stimulate the control loop that is closed with a cut-off frequency at 300Hz. So the quadrature current has to be in phase with the sinusoidal reference.

As can be seen in Figure 6.18, the quadrature current is perfectly in phase and the controller ensures good tracking of the reference.

The direct current, see Figure 6.19, has a good behavior in fact it stays always around the reference, so the generated torque has very little oscillations around the working point.

In Figure 6.20 it's possible to notice the mechanical behavior of the rotor that has a sinusoidal trend like the quadrature current.

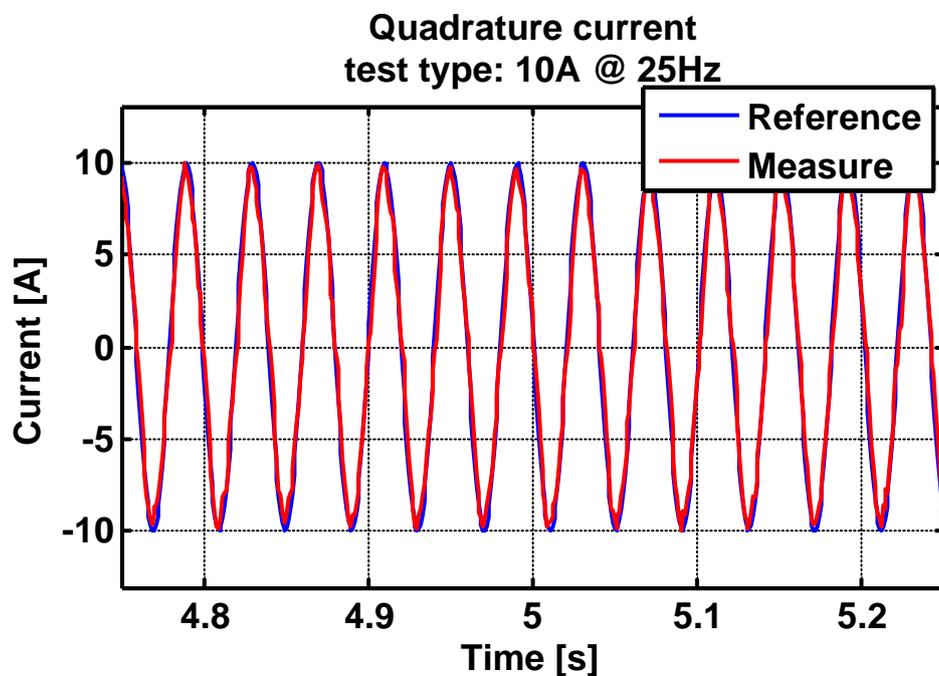


Figure 6.18 Quadrature current for a sinusoidal reference at 25Hz and 10A of amplitude

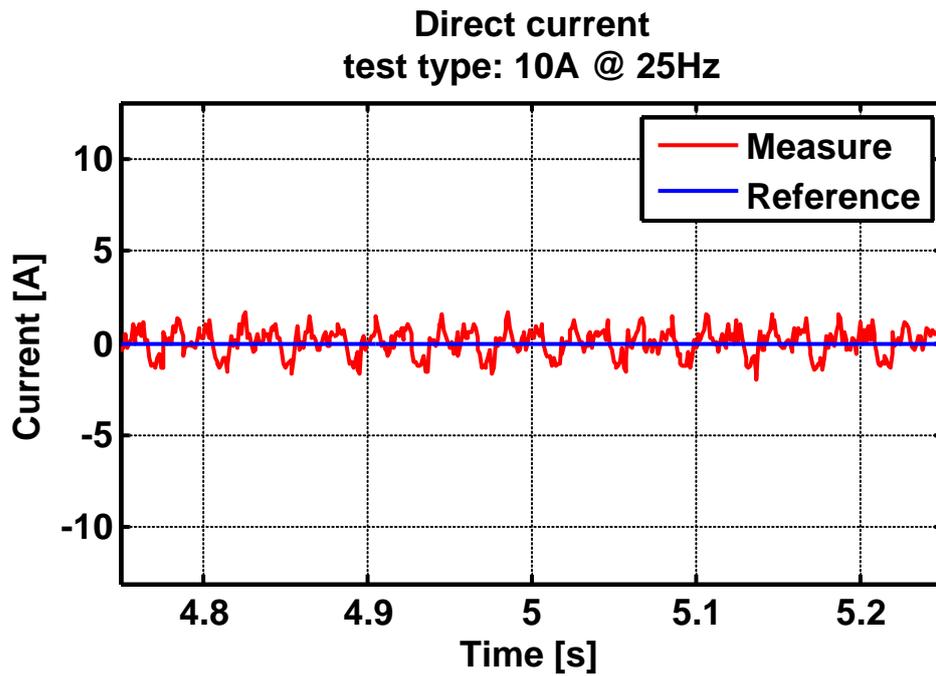


Figure 6.19 Direct current for a sinusoidal reference at 25Hz and 10A of amplitude

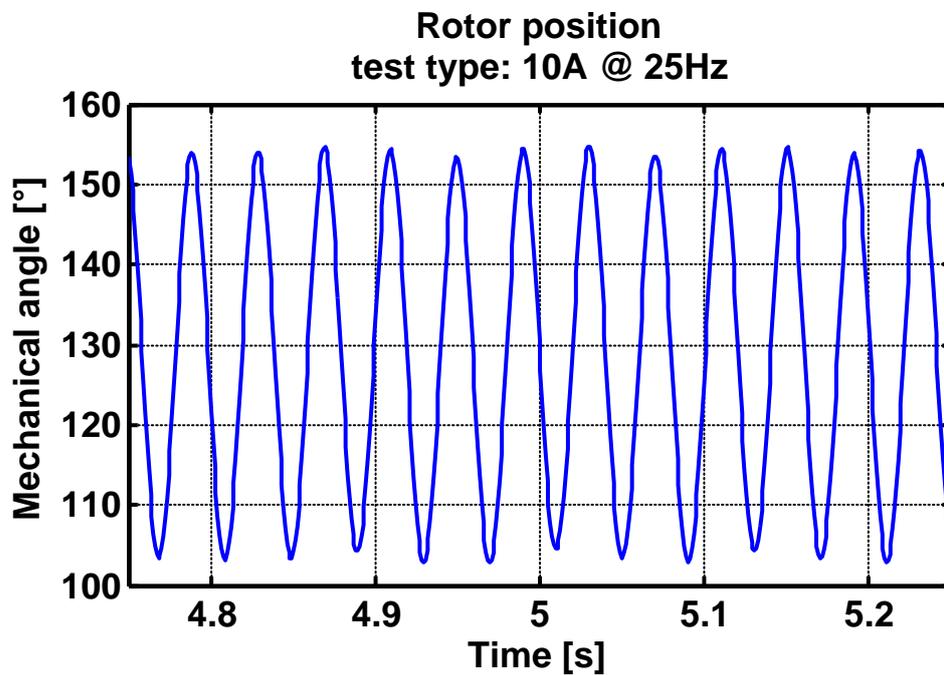


Figure 6.20 Rotor position for a sinusoidal reference at 25Hz and 10A of amplitude

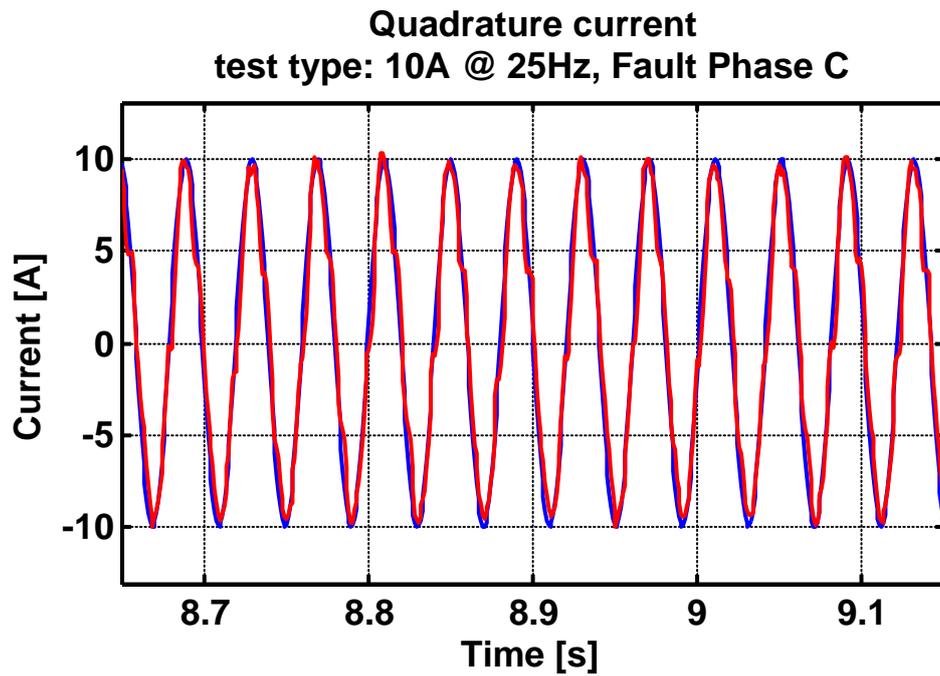


Figure 6.21 Quadrature current for a sinusoidal reference at 25Hz and 10A of amplitude, with phase C interrupted

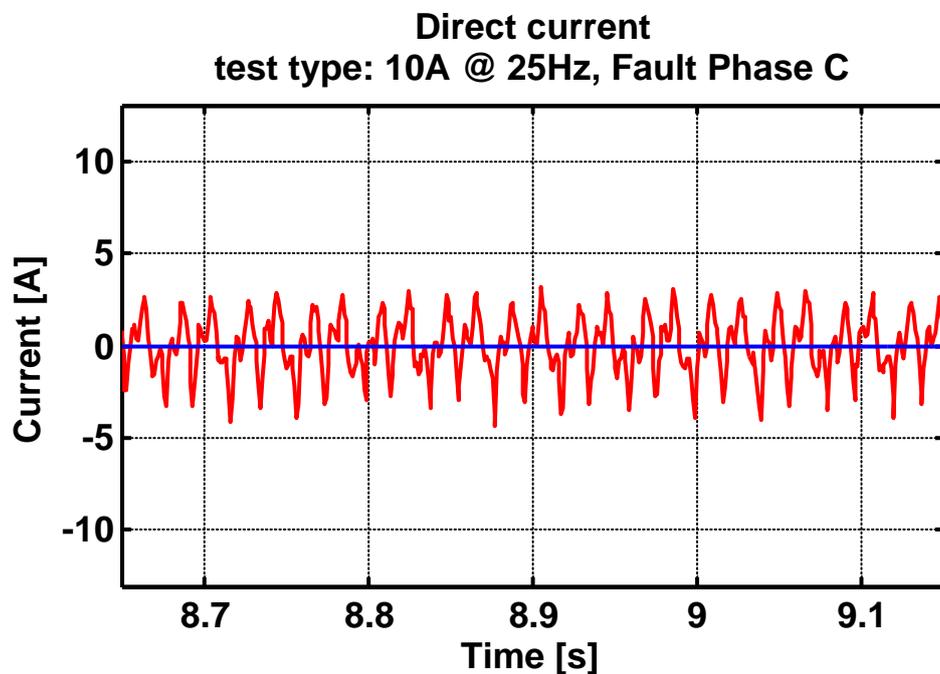


Figure 6.22 Direct current for a sinusoidal reference at 25Hz and 10A of amplitude, with phase C interrupted

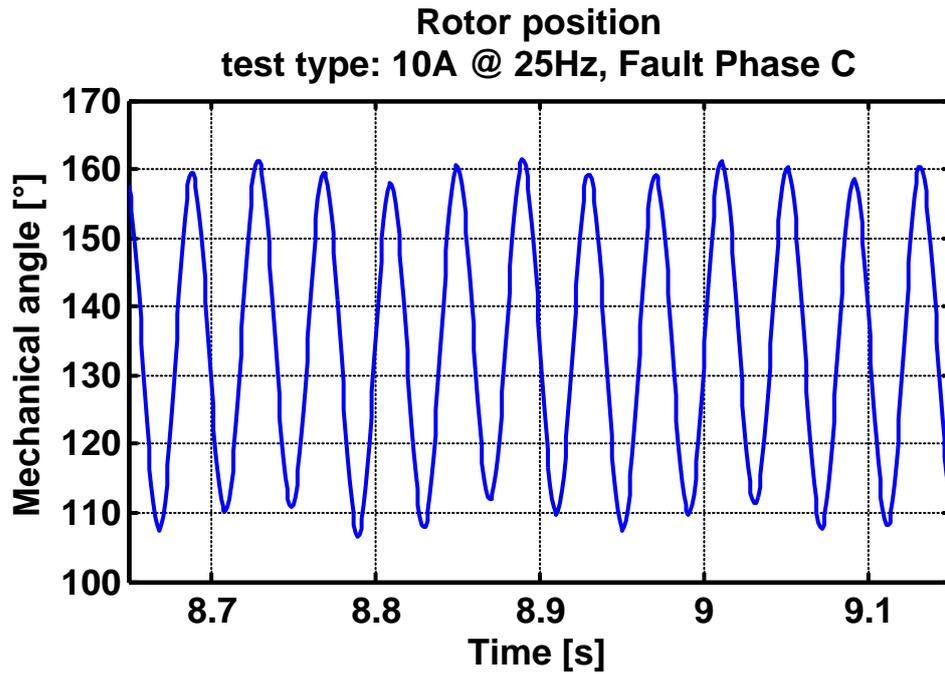


Figure 6.23 Rotor position for a sinusoidal reference at 25Hz and 10A of amplitude, with phase C interrupted

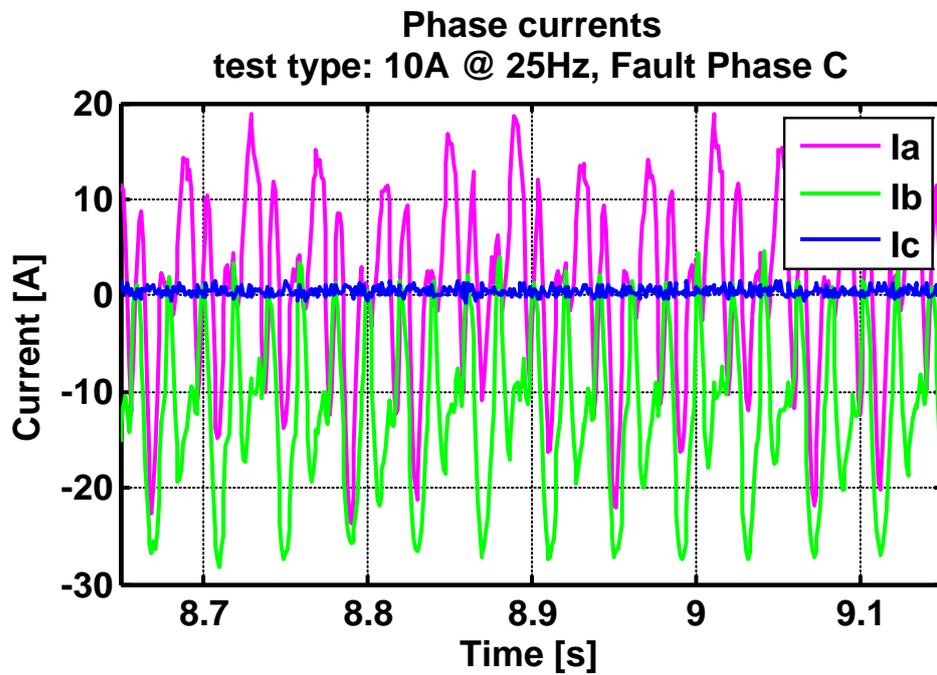


Figure 6.24 Phase currents for a sinusoidal reference at 25Hz and 10A of amplitude, with phase C interrupted

The current control for a motor with open-end windings is very safe because if a phase is interrupted due to a fault the behavior remains the same with a very little degradation, see Figure 6.21, Figure 6.22, Figure 6.23 and Figure 6.24.

In Table 6.3 it's possible to compare the results without and with the fault examining the Root Mean Square (RMS) of the currents signals and the peak-to-peak for the rotor position.

	<i>No Fault</i>	<i>Fault C</i>
I_{q-rms} [A]	6.5832	6.5595
I_{d-rms} [A]	0.73838	1.5509
Rotor position [°]	51.5	48

Table 6.3 Comparison of performances with and without the fault

6.6 Concluding remarks

In this chapter have been described the implementation of a current control loop for a brushless motor with open-end windings with a new strategy of a space vector modulation for a dual level dual inverter.

It was shown that this method is safe and fault tolerant, in fact the current control has an optimal behavior against a fault occurred on phase C.

Chapter 7

Experimental results

The controller performances have been tested on an AGV used on a commercial vehicle engines assembly line in order to verify the real behavior of the AGV.

The vehicle will be prepared with three different operating configurations and using a repeatable experiment the results will be compared. Afterwards will be examined the tests to complete the analysis.

7.1 Description of the test

Accordingly to the parameter analysis described in paragraph 5.3, the selected controller parameters shown in Table 3.1 have been tested on vehicle. The vehicle has been prepared with three different operating configurations:

- No load, $M = 50\text{kg}$;
- Small load, $M = 200\text{kg}$;
- Full load, $M = 1000\text{kg}$;

To validate the mathematical analysis described in Chapter 5, a repeatable experiment has been implemented. The AGV has to follow a path, see Figure 7.1,

composed by three pieces: a straight stretch followed by a chicane with the first left bend and then another straight stretch. In this way the steering control is well stimulated and setting the longitudinal speed $\bar{\omega}^*$ for all the tests to 15m/min it's possible to compare the results obtained with different loads.

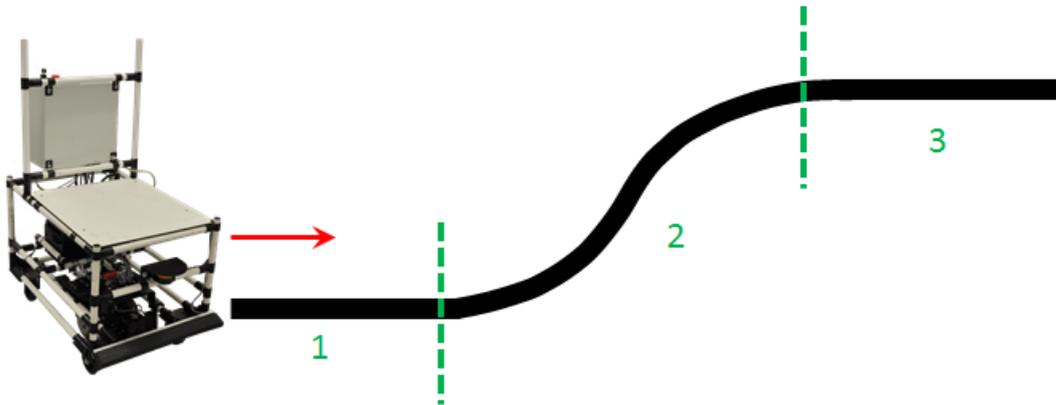


Figure 7.1 Path followed by AGV with three parts: 1) straight stretch, 2) chicane with the first left bend and 3) straight stretch

AGV has to go along the path following the red arrow, see Figure 7.1, the Guidance board, using its three magnetic sensor, reconstructs the relative position between the magnetic tape and the driving group. Afterwards it sends $\Delta\omega^*$ through the CAN bus to the Input Output board in order to perform the path tracking control as described in Chapter 5.

7.2 Performances analysis

The AGV was prepared to perform the tests. The Guidance board provides the reference $\Delta\omega^*$ after a processing of the three measurement signals of the magnetic field. All the signals were logged with CANcase and then analyzed through CANalyzer.



Figure 7.2 CANcase: CAN interface



Figure 7.3 CANalyzer screenshot

The former is a Network Interface with USB Interface for CAN and the latter is the universal software analysis tool for ECU networks and distributed systems. With this software is very easy to observe, analyze, and supplement data traffic via CAN protocol, see Figure 7.2 and Figure 7.3.

Performing the tests described in paragraph 7.1 with the controller parameters C_1 described in Table 5.1 it's possible to obtain the results shown in Figure 7.4 and Figure 7.5.

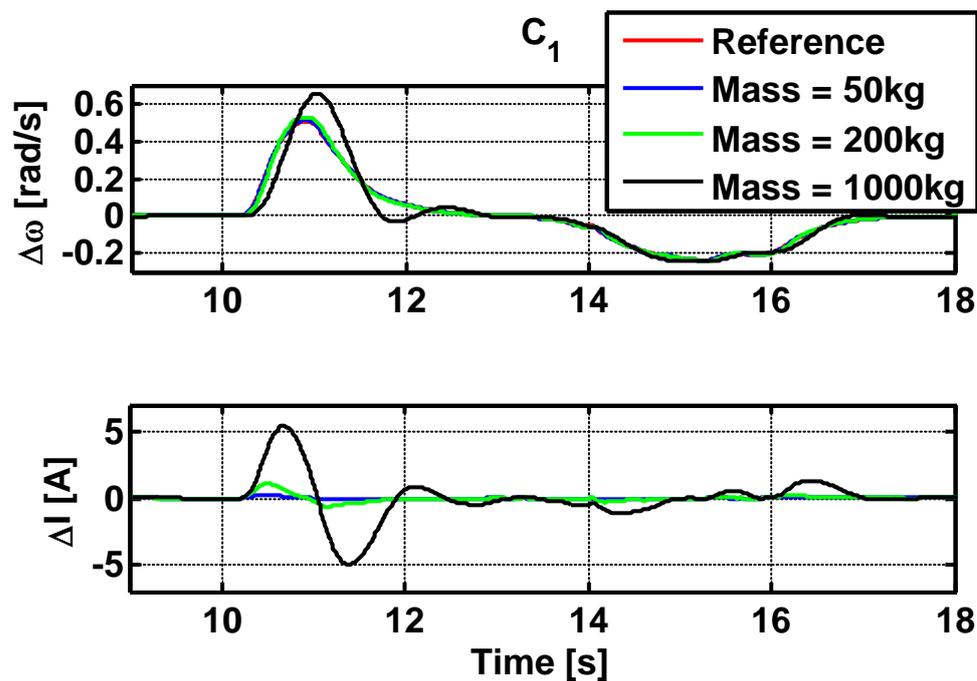


Figure 7.4 Closed-loop steering control $\Delta\omega$ in the experimental test with different loads, C_1 controller

The system ensures good tracking of $\Delta\omega^*$ and robustness on changes of mass. Specifically, when the mass is equal to 50kg the tracking performance is very good, in fact the reference speed, red line, is completely hidden by the controlled output, blue line, see Figure 7.5. For higher mass values acceptable performance decrease is observed green and black lines.

As expected, increasing the mass, the response time of the system becomes longer and produces an overshoot but without a performance deterioration. In fact, the

vehicle can always follow the magnetic tape without activating the derailment diagnostic set to 30cm.

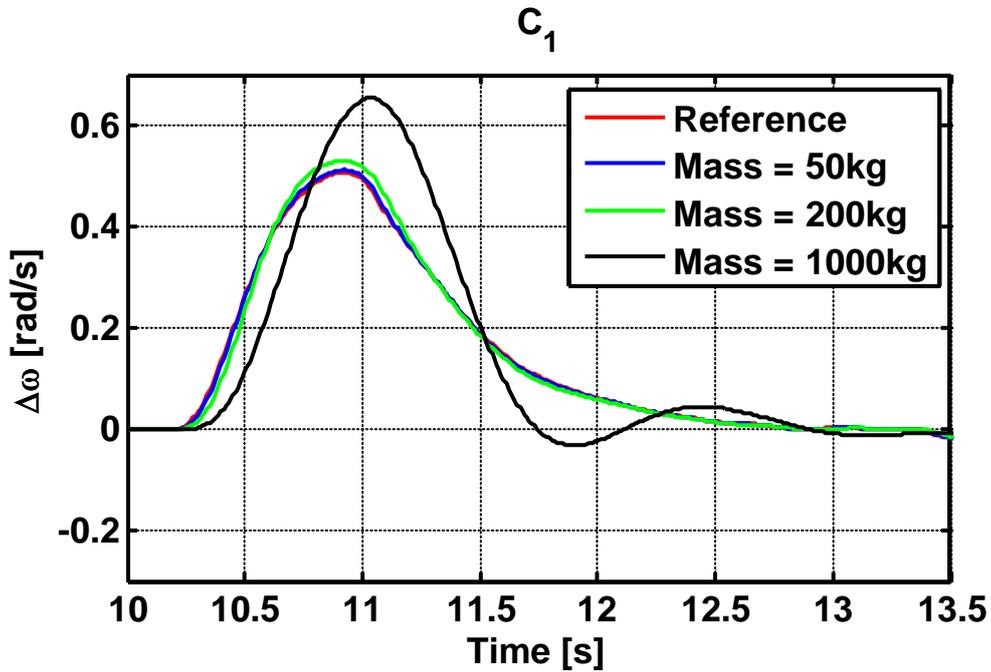


Figure 7.5 Zoom of closed-loop steering speed $\Delta\omega$ in the experimental test with different loads, C_1 controller

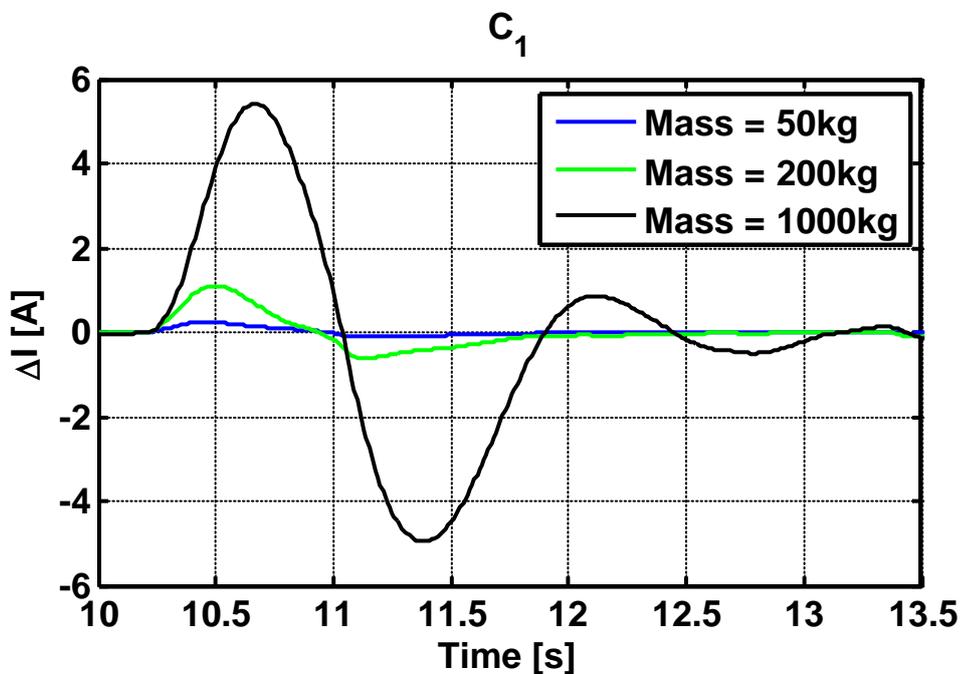


Figure 7.6 Zoom of closed-loop control variable Δi in the experimental test with different loads, C_1 controller

As can be seen in Figure 7.4 and Figure 7.6, the control variable Δi increases with increasing the mass but the peak of the current, however, remains lower than $\Delta i^* = 8A$ with which the controller was designed. In fact the controller was designed to remain under the threshold of 8A for a step variation of the reference.

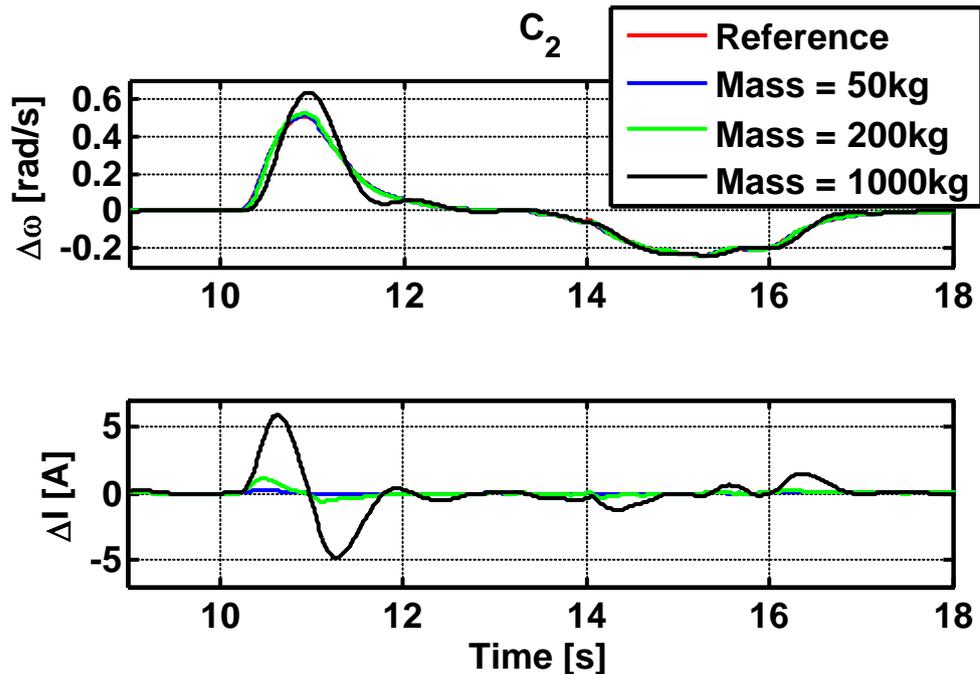


Figure 7.7 Closed-loop steering control $\Delta\omega$ in the experimental test with different loads, C_2 controller

The performances of the closed loop system with the controller parameters C_2 are shown in Figure 7.7, Figure 7.8 and Figure 7.9.

As can be seen in Figure 7.4 and Figure 7.7, the performances of closed loop system in the second configuration C_2 are better, as expected from analysis described in paragraph 5.3, than the ones obtained through the first configuration. In fact, there are smaller oscillations when the AGV is fully loaded. Conversely there is a higher utilization of the control variable Δi and consequently a higher energy consumption. This is in accordance with the controller specifications that have larger current limitations in this second case.

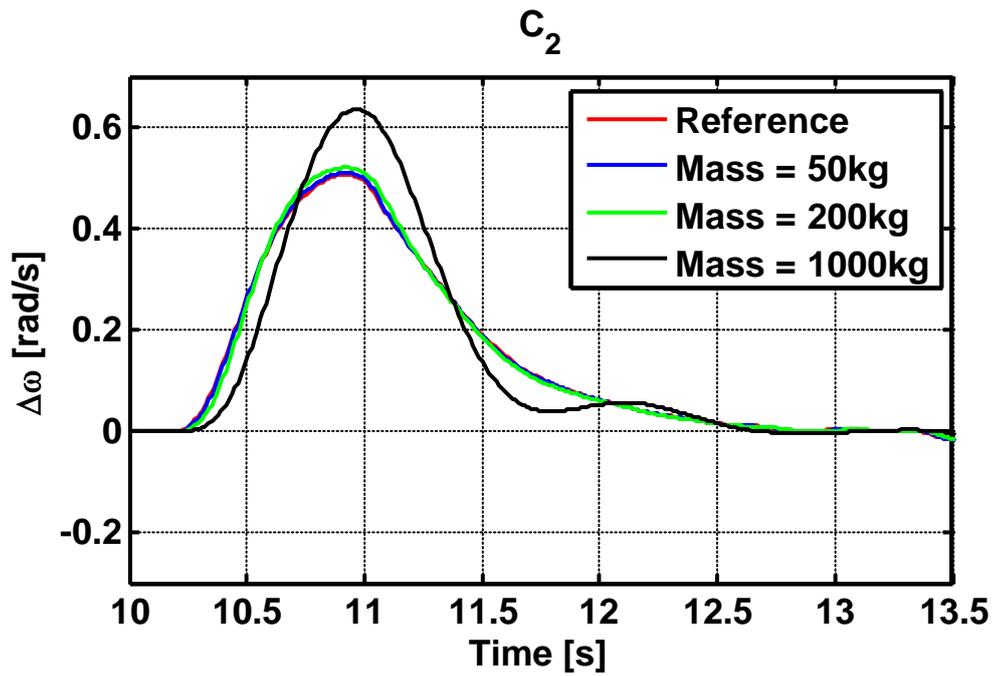


Figure 7.8 Zoom of closed-loop steering speed $\Delta\omega$ in the experimental test with different loads, C_2 controller

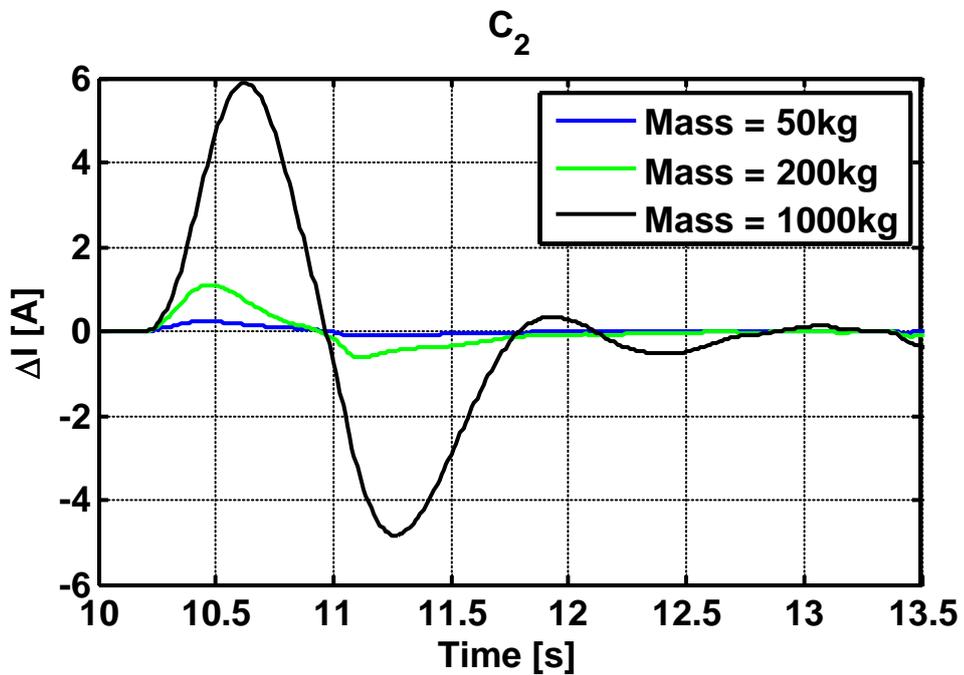


Figure 7.9 Zoom of closed-loop control variable Δi in the experimental test with different loads, C_2 controller

As a final remark, as highlighted by the performances achievable with both the controllers experimentally tested, the proposed method allows easy control design robust to any load condition, with clear representation of performance requirements, control limitations and H_∞ limits.

7.3 Concluding remarks

In this chapter were shown the achievable performances with two types of controllers selected by the technique described in Chapter 5, to satisfy response time, peak control action and the H_∞ norm of the closed-loop lower than γ .

It's easy to verify that both controllers ensures good tracking for all the types of loads and the time responses and current consumption are consistent with what was designed.

Chapter 8

Conclusions

The purpose of this work have been the analysis , design and implementation of a control loop of the path tracking for a light AGV. The controller has to guarantee desirable specifications for a wide range of possible load mass values (from 50kg to 1000kg). So the payload mass is an uncertainty in the model and resort to robust control design methods. The analysis involved the stability domain in the controller parameters, in this way it's possible to evaluate the effects on the guidance performances changing the controller parameters, preserving control system stability. To obtain such results we paid special attention to randomized techniques and fixed order controller design techniques.

This approach is useful because it allows not only to compute controller parameters but also to compute a discrete set of stabilizing controllers and to obtain a configuration of the admissible domain in the parameter space.

Based on the analysis performed following the robust control theory, randomized techniques and parameter space techniques, two controllers have been chosen to be tested on the vehicle. Therefore these controllers have been implemented in Matlab/Simulink and the automatically translated in C code, afterwards this C code, together with middleware and firmware, is compiled and downloaded on the microcontroller.

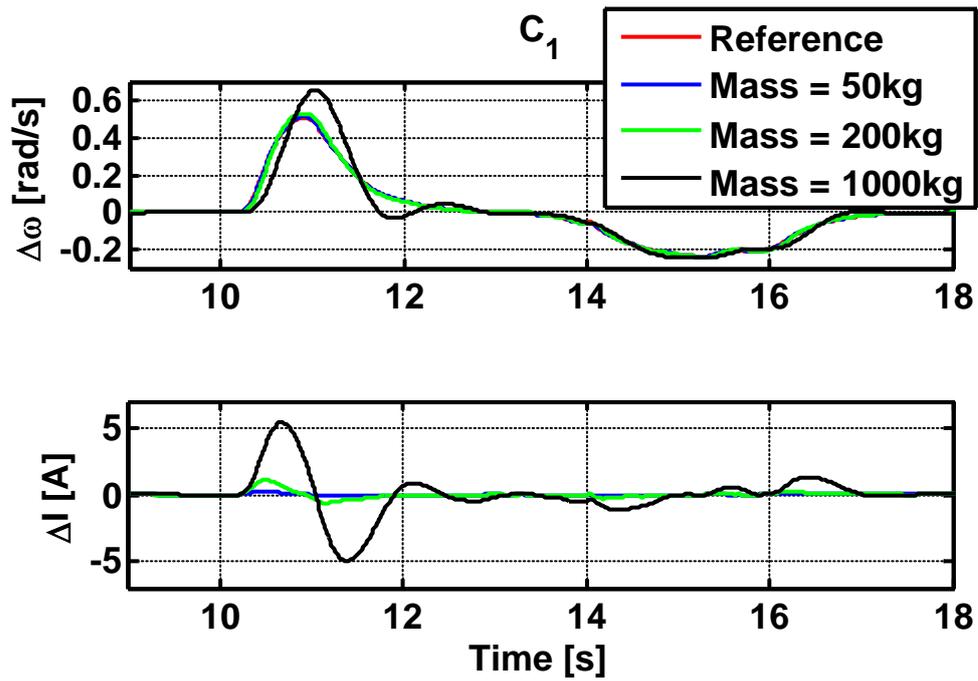


Figure 8.1 Closed-loop steering control $\Delta\omega$ in the experimental test with different loads, C_1 controller

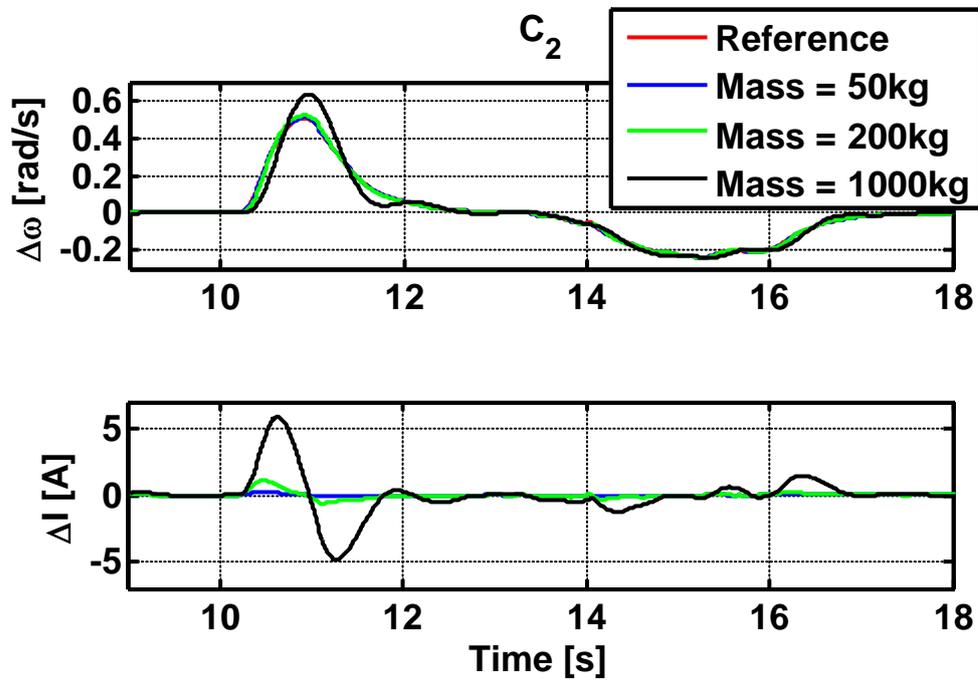


Figure 8.2 Closed-loop steering control $\Delta\omega$ in the experimental test with different loads, C_2 controller

The two chosen controllers have been tested on a track composed by three pieces: a straight stretch followed by a chicane with the first left bend and then another straight stretch. This track shape allow to well stimulate the steering control and evaluate correctly the performances.

The vehicle has been prepared with three different operating configurations:

- No load, $M = 50\text{kg}$;
- Small load, $M = 200\text{kg}$;
- Full load, $M = 1000\text{kg}$;

It has been demonstrated that the performances are consistent with the analysis and the AGV follows correctly and without problems the track in all load conditions.

8.1 Future works

It's possible to find inside this work new future works that can be developed, such as:

- Remove the touch panel and substitute it with a tablet or smartphone connected to the vehicle via Wi-Fi.
- Create an interface in order to command a fleet of AGVs inside a plant by means of a tablet or smartphone via Wi-Fi.
- Test the vehicle behavior when a trailer with a full payload is connected to the AGV
- Close the longitudinal speed loop with an approach similar to the one used for the steering speed

- Write the software safety functions compliant to the IEC 61508 in order to certify the software and get the correct Safety Integrity Level (e.g. SIL2)

Bibliography

- [1] R. Tempo, G. Calafiore e F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*, London: Springer-Verlag, 2004.
- [2] Y. Fujisaki, Y. Oishi e R. Tempo, «Mixed deterministic/randomized methods for fixed order controller design,» in *Tac Vol. AC-53, pages 2033-2047*, 2008.
- [3] B. Polyak e E. Gryazina, «Hit-and-Run: New Design Technique for Stabilization, Robustness and Optimization of Linear Systems,» in *Proceedings of the 17th IFAC World Congress, Seoul, pages 376-380*, 2008.
- [4] D. Arzelier, E. Gryazina, D. Peaucelle e B. Polyak, «D. Arzelier and E. Gryazina and D. Peaucelle and B. Polyak,» in *Proceedings of the IEEE American Control Conference*, Baltimore, 2010.
- [5] K. J. Åström e T. Hägglund, *PID Controllers*, Research Triangle Park, NC, USA: International Society for Measurement and Control, 1995.
- [6] The MathWorks, *Matlab & Simulink - Getting Started Guide - R2011b*, Natick: The MathWorks, 2011.
- [7] J. C. Alexander e J. H. Maddocks, «On the Kinematics of Wheeled Mobile Robots,» *Int. J. of Robotic Research*, vol. 8, n. 5, pp. 15-27, 1989.
- [8] C. Spelta e E. Gryazina, «Describing the H_∞ set in the controller parameter space,» in *European Control Conference, pp 816-823*, Kos, 2007.
- [9] E. Gryazina e B. Polyak, «Stability regions in the parameter space: D-decomposition revisited,» *Automatica*, vol. 42, n. 1, pp. 13-26, 2006.

- [10] R. Smith, «Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions,» *Operations Research*, vol. 32, n. 6, pp. 1296-1308, 1984.
- [11] B. Polyak e E. Gryazina, «Robust stabilization via Hit-and-Run techniques,» in *IEEE Multi-Conference on Systems and Control*, pp. 537-541, Saint Petersburg, 2009.
- [12] N. Celanovic, D. Boroyevich, H. C. Skudelny and G. V. Stanke, "A fast Space Vector Modulation algorithm for multilevel three-phase converters," in *IEEE Transaction on Industry Application Vol.37, No. 2*, March/April 2011.
- [13] A. K. Gupta e A. Khambadkone, «A Space Vector PWM Scheme for Multilevel Inverters Based on Two-Level Space Vector PWM,» in *IEEE Transaction on Industrial Electronics Vol.53, No. 5*, October 2006.
- [14] H. W. Van Der Broeck, H. C. Skudelny e G. V. Stake, «Analysis and Realization of a Pulsewidth Modulator Based on Voltage Space Vectors,» in *IEEE Transaction on Industry Application Vol. 24, No. 1*, January/February 1988.
- [15] Freescale Semiconductor, «56800E MCLIB,» Freescale Semiconductor, 2011.
- [16] Freescale Semiconductor, «Design of a PMSM Servo System using the 56F8357 device,» Freescale Semiconductor, 2006.
- [17] M. G. Madaschi, E. Gryazina, A. L. Cologni, C. Spelta, F. Previdi, S. M. Savaresi e I. Pesenti, «Robust control of magnetic guidance lightweight AGVs path tracking using randomization methods,» in *Proc. of European Control Conference*, Zurich, 2013.
- [18] E. G. Shivakumar, K. Gopakumar, S. K. Sinha, A. Pittet e V. T. Ranganathan, «Space vector PWM control of a dual inverter fed open-end winding induction motor drive,» in *APEC*, Anaheim, CA, 2001.
- [19] J. Rodriguez, J. S. Lai e F. Z. Peng, «Multilevel inverters: a survey of topologies, controls and applications,» *IEEE Transactions on Industrial*

- Electronics*, vol. 49, n. 4, pp. 724 - 738, 2002.
- [20] M. A. S. Anness, A. Gopinah e M. R. Baiju, «A simple space vector PWM generation scheme for any general n-level inverter,» *IEEE Transactions on Industrial Electronics*, vol. 56, n. 5, pp. 1649 - 1656, 2009.
- [21] M. R. Baiju, K. K. Mohapatra, R. S. Kanchan e K. Gopakumar, «A dual two-level inverter scheme with common mode voltage elimination for an induction motor drive,» *IEEE Transactions on Power Electronics*, vol. 19, n. 3, pp. 794 - 805, 2004.
- [22] S. Ahmed, . G. M. A. Sowilam e A. Salim, «Practical implementation of a dual inverter operates open ends induction motor,» in *Design and Test Workshop (IDT)*, Riyadh, 2009.
- [23] K. Ramachandrasekhar, S. Mohan e S. Srinivas, «An improved PWM for a dual two-level inverter fed open-end winding induction motor drive,» in *2010 XIX International Conference on Electrical Machines (ICEM)*, Rome, 2010.
- [24] S. Bittanti, *Identificazione dei modelli e sistemi adattativi*, Bologna: Pitagora Editrice, 2005.
- [25] P. Bolzern, R. Scattolini e N. Schiavoni, *Fondamenti di automatica*, Milano: McGraw Hill, 1998.

