Università degli Studi di Bergamo

Dipartimento di Ingegneria Gestionale dell'Informazione e della Produzione

Dottorato di Ricerca in Meccatronica, Informazione, Tecnologie Innovative e Metodi Matematici

XXVIII Ciclo

Statistical Functional Data Approach to Early Detection and Diagnostics. With Application to Steam Sterilizers.

Relatore	**Tesi di**
Chiar.mo Prof. Alessandro Fassó	Maurizio Toccu
	matr. 1013256

Anno Accademico

2014/2015

RINGRAZIAMENTI

In primis, ringrazio i Professori Alessandro Fassó e Giuseppe Psaila per la disponibilitá dimostratami nei tre anni di Dottorato.

Inoltre, ringrazio il Personale della W&H Sterilization per il supporto fornitomi durante la fase delle prove di laboratorio.

Infine, ringrazio i miei Genitori e mia Sorella per la vicinanza durante questo lungo ma stimolante percorso di crescita personale e professionale.

Contents

Chapter 1

Introduction

Statistical surveillance is the continuing collection, analysis and interpretation of data concerning the health of a system and its main instruments are control charts. These instruments have been traditionally used to monitor industrial processes. However, they are increasingly being used in many applications fields such as engineering, economics, finance, epidemiology and environmental statistics.

In literature a number of different control charts have been introduced beginning from the simplest Shewhart control chart, which dates back to 1920's and was introduced to monitor the mean quality level of a production process, see Bayart[1]. A popular alternative is the Exponentially Weighted Moving Average (EWMA), which is effective in discovering small level shifts. There are several variations and application fields of the EWMA control chart. For example, recently Lazariv and Schmid[2] and Zhang and Pintar[3] study some EWMA control charts for the variance of a stationary time series, whith interesting application in finance.

However, despite the use of univariate control charts is a useful solution in several situations, in other cases it is inefficient and can lead to wrong conclusions. In fact, many situations are characterized by several related variables, and multivariate methods are required. In this context, Lowry et al.[4] extended the EWMA algorithm to the multivariate case and developed the multivariate exponentially weighted moving average (MEWMA) control charts, that is one of the most popular instruments related to statistical surveillance.

Recently, MEWMA control charts are meeting new application fields. For example Fassó and Locatelli[5] consider a generalized one-sided MEWMA control chart, which is capable to monitor dimension and deformation in production of vehicle brake discs. Sparks[6] considers social networks monitoring, by means of detecting changes in communication volume between social network members. Moreover, considering finance data, Golosnoy and Schmid[7] present some control charts based on exponential smoothing in order to monitor the weights of the "global minimum variance portfolio". Epidemiology and health surveillance are another emerging area. For example, Tse et al.[8] present a multivariate control chart based on likelihood ratio tests that allows to discover the variance-covariance shift due to a disease outbreak and Han and Zhong[9] compare MEWMA based spatio-temporal control charts.

A further development of statistical surveillance is motivated by advanced technological applications, where flexible models are often necessary for properly monitoring process outputs which are given by nonlinear curves or profiles. The simplest approach is given by Fan et al.[10], which use a piecewise linear approximation. Chang and Yadama[11] propose a method to monitor non-linear profiles, which is capable to detect mean shifts or shape changes. Cano et al.[12] introduce a novel hybrid nonparametric-parametric procedure for monitoring nonlinear profiles, which considers the shape property of the profiles. Zou and Qiu[13] develop an innovative multivariate statistical process control method based on the LASSO, which is capable to detect the shift of a small number of components in a large multivariate vector. Finally, Qiu et al.[14] propose nonparametric profile monitoring where within-profile data are correlated.

In this work, we propose the innovative f-MEWMA control chart that integrates splines into the multivariate exponentially weighted moving average scheme. In particular, our approach transforms the multiple data profiles in spline form, thus generating a multivariate functional object which is repeatedly observed over time. Then its trend is monitored by the f-MEWMA control chart, which is able to detect the early onset of anomalies in the given system. In particular f-MEWMA extends the traditional MEWMA control chart exploiting functional data and is capable of monitoring the health of a system which, at each run, generates complex output data given by nonlinear profiles In order to show the effectiveness of f-MEWMA algorithm, we consider a steam sterilizer for medical equipments that, for each sterilization run, measures several variables in high frequency giving a set of sterilization profiles. During each sterilization run, several variables (temperature, pressure etc.) are measured in high frequency. Hence variables are functions of sterilization time for each sterilization run. Because the profiles of each sterilization run are arbitrarily nonlinear smooth functions of sterilization time, we use splines to model them.

The rest of the thesis is structured as follows.

In the Chapter 2, first of all, we present the W&H Sterilization that is the company that financed our research project. The W&H managers made available to us both laboratories and three new sterilizers *Lisa522 Fully Automatic* in order to perform the experiments. Moreover, we present the sterilizer *Lisa522 Fully Automatic*, dividing this Chapter in two parts:

- In the first part, we describe the engineering characteristics of our steam sterilizers (technical projects and operational functioning), the three types of cycle (in our experiments we use Universal 134), the phases that compose each sterilization cycle. Finally we describe the log-files that the sterilizer produces. In particular, these log-files contain the operating parameters (variables) that the sterilizers sensors measure second by second.

- In the second part, we provide an overview of some sterilizer components that we tampered with in order to perform altered sterilization cycles and, then, we study the behavior of our sterilizers in anomalous conditions. In particular, we made the choice of the components taking into account both the frequency of the fault of a give component, based on market data, and the technical feasibility of the simulation of this faults/anomalies.

Chapter 3 presents the statistical methods that we used in Chapter 4 to build the our innovative control chart. We divide this Chapter in two parts:

- In the first part, we introduce the Functional Data Analysis (FDA) and recall spline concepts. In particular, we use cubic spline to model each sterilization cycle. Moreover, we introduce the connection between splines and linear mixed models in which we show fixed and random effects. Finally, exploiting the linear mixed model representation, we introduce the penalized least square function.

- In the second part, we introduce the statistical surveillance and some control charts. Moreover, we introduce our innovative approach called *functional MEWMA* or *f-MEWMA* that is developed by combining together functional data analysis and Multivariate Exponentially Weighted Moving Average control chart. Our method is able to monitor the health of a system which, at each sterilization cycle, generates complex output data given by nonlinear profiles. Finally, we introduce a probabilistic classifier that allows to classify each sterilization cycle with respect to the selected components/faults.

Chapter 4 presents the data and the results of our research. In particular, we divide this Chapter in two parts:

- In the first part, we describe the design of experiments that we used to performed the *sterilization cycles*. In particular, we followed a quasi balanced DOE involving six factors which are related to components that we described in Chapter 2. For each factor we considered two levels: fault free (good runs) vs altered. Finally, we perform an exploratory analyses of the collected data. In particular, by nonparametric ANOVA (Kruskal-Wallis test), we compare the behavior of the sterilizer when it works in normal conditions with respect it works in altered conditions.

- In the second part, we show how to applicate the algorithm *f-MEWMA* to the data generated from the above sterilizers, in order to demonstrate the effectiveness of the our method. Moreover, we present a probabilistic anomalies classifier that allows to classify the sterilization cycles with respect to faults.

Finally, in Chapter 5 we present our conclusions and future developments.

Chapter 2

Lisa 522, Variables and Faults

Sterilization is defined as the end result of a process that tends to ensure the condition in which the survival of bacteria is highly unlikely. Therefore, to perform a sterilization it is necessary to use a medical device called *sterilizer* or *autoclave*. In particular, a sterilizer is a pressure chamber used to sterilize equipment and supplies by subjecting them to high pressure saturated steam at several temperatures. Autoclaves are found in many medical settings, laboratories, and other places that need to ensure the sterility of an object. In our case, dentists use the sterilizers *Lisa 522 Fully Automatic* to sanitize dental instruments that they use for curing their patients. Lisa 522 is a sterilizer produced by W&H Sterilization. In particular, W&H is a company that operates in more than 110 countries worldwide and manufactures instruments and transmission equipment for dentistry. The W&H products are among the most high-quality tools available on the market, used in dental offices, dental clinics, dental laboratories and in microsurgery. This company makes significant investments in research and development in order to improve their products, in particular sterilizers.

First of all, in this Chapter we present the Lisa 522 from the engineering point of view. Moreover, we describe variables that the sterilizer measures second by second, the types of cycle (in particular we used the Universal134) and phases in which each sterilization run is divided. Finally, we describe the laboratory activities and in particular the components that we tampered with in order to simulate some anomalies/faults related to above components.

2.1 Lisa522 Fully Automatic

Lisa522 Fully Automatic is a sterilizer manufactured by W&H Sterilization. With the programmable delayed cycle start it is possible to save time and energy. Made-to-measure automatic cycles reduces the cycle time according to the number and type of items to be sterilized. An integrated memory card with USB reader archives the sterilization and test cycle reports.

Figure 2.1: Lisa 522 Fully Automatic

The sterilizer Lisa522 has an integrated and automatic traceability system with user identification and load release option. The double micro-processor technology manages the integrated traceability software which guarantees the speed and efficiency of the sterilization cycles.

The patented water separation and filtering system of the Lisa sterilizer prevents the penetration of oil residues and other impurities in the vacuum pump. Through the automatic Air Detection System the automatic cycle performance is ensured and the result is an additionally extra safety infection prevention. Lisa 522 Fully Automatic sterilizers feature type B sterilization cycles. They are developed, manufactured and tested according to the European Norm EN 13060 for small water steam sterilizers.

At the end of each sterilization cycle (*run*), Lisa 522 Fully Automatic produces a log-file that contains the measures of the operating parameters (*variables*). We will describe these variables in subsection 2.3.

2.1.1 2CS System

The patented 2CS system is designed to collect condensate that drops to the bottom of the sterilization chamber during all steam phases of sterilization cycles, and return it to the steam generator to produce more steam. As a result the sterilizer uses considerably less water, and as the returned condensate from the chamber is boiling hot, it also helps to save energy and time. The 2CS process is active:

- during the fractionated pre-vacuum after every steam phase;
- during the sterilization (plateau).

Before describing the system 2CS, we explain the operation of the solenoid valve EVD that regulates the injections of steam into the chamber by 2CS. In particular:

- when EVD switches ON (horizontal ; bit=1 ; step 56), then the steam enters into the chamber;
- when EVD switches OFF (vertical ; bit=0 ; step 55), allows to recover condensate (loop 2CS) and then to use it in the cycle (saving water).

The system 2CS consists of below three steps.

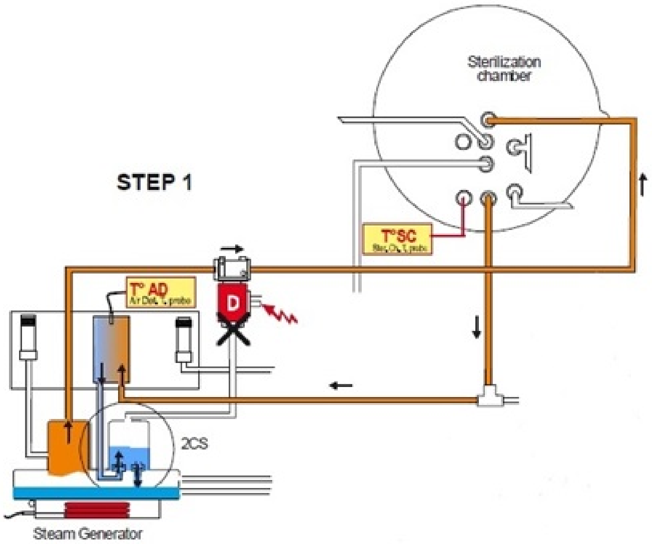


Figure 2.2: System 2CS - Step1

During steam injections into the sterilizer chamber (solenoid valve EVD is switched horizontal), condensate collects at the bottom/rear of the chamber and flows to the 2CS tower of the steam generator.

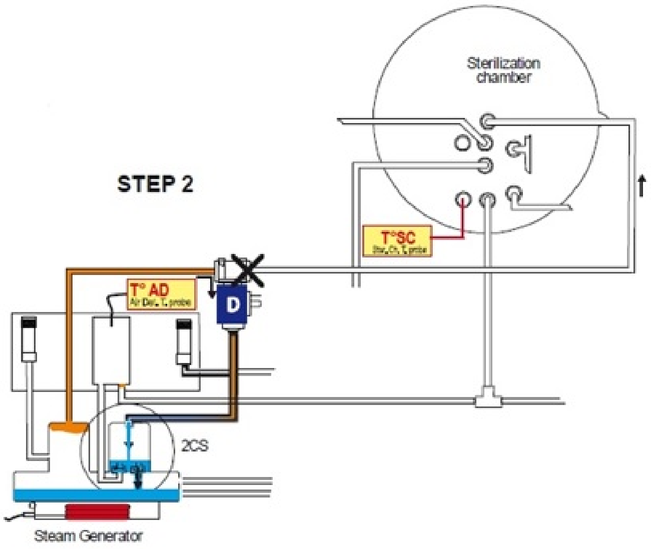


Figure 2.3: System 2CS - Step2

Whenever solenoid valve EVD switches vertical, there is a closed circuit between the 2CS tower and the steam generator main chamber (pressure balance; equal pressure). As a consequence, the water collected in the 2CS collector drops (gravity) through the internal one-way valve into the steam generator.

The steam generator turns the water into steam, which is injected into the chamber through valve EVD (switched horizontal; bit=1; step=56).

Finally, we explain the operation of the Air Detector system, called *T-AD*, that is monitored by the variable `TAir`). Every sterilization cycle on Lisa 500 sterilizers features a pre-vacuum air removal phase. During this phase air is removed from the chamber/load and is replaced with steam. At the end of the pre-vacuum phase prior to heating the load to sterilization conditions, the air detector checks for the presence of non-condensible gases (air), thus ensuring that the pre-vacuum air removal phase was successful.

A temperature sensor mounted in the air detector tower reads the steam temperature while a pressure transducer mounted in the sterilizer chamber reads the steam pressure. The sterilizer software checks if the temperature and pressure readouts are

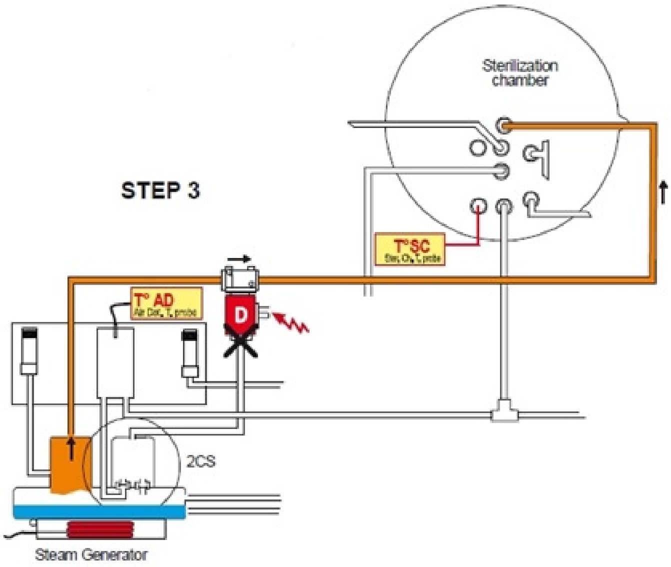


Figure 2.4: System 2CS - Step3

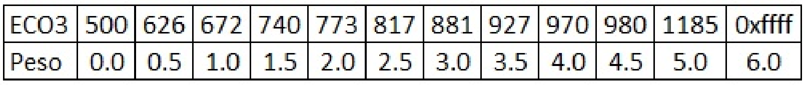
in accordance with the saturated steam pressure-temperature correlation. In case the correlation is off (there is air present in the chamber) the alarm code A190 is generated and the cycle is aborted.

2.1.2 ECO System

The procedure ECO-DRY allows the adaptation of the drying time according to the amount of load placed inside the chamber. The procedure involved ECO-DRY the variables: `ECO3`, `H20`, `DV2`. With loads of more than 4 kg drying time is automatically set to 16 minutes.

2.1.3 ECO3 Variable

`ECO3` is an accumulation variable of dissipated energy by the generator during a specific period of the phase PPH identified as $PPH1$. The variable `ECO3` is related to amount of load placed into the chamber of sterilizer. Physically, this is due to the fact that the greater the load to be heated, more electrical energy must be introduced into the system. Figure 2.5 shows the threshold values of the variable `ECO3` and corresponding estimates of load weight (kg).

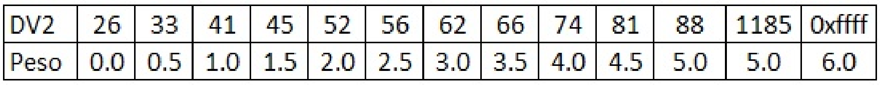


ECO3	500	626	672	740	773	817	881	927	970	980	1185	0xffff
Peso	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	6.0

Figure 2.5: Variable ECO3 - Threshold values and estimates of load weight (kg)

2.1.4 DV2 Variable

DV2 is the duration of the *step*60. During this step the sterilizer passes by a known and fixed pressure (1.32 absolute bar) to another known and fixed pressure (0.50 absolute bar). The time that the sterilizer employs to perform this step is strongly correlated to the amount of load fed into the chamber of sterilizer. Physically, the sterilization chamber is brought to depression by the vacuum pump and the necessary time depends on residual moisture in the chamber and the load volume. Figure 2.6 shows the threshold values of the variable DV2 and corresponding estimates of load weight (kg).

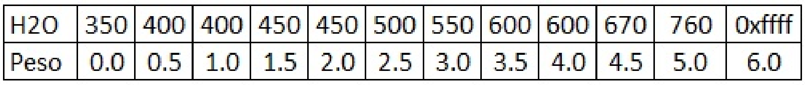


DV2	26	33	41	45	52	56	62	66	74	81	88	1185	0xffff
Peso	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.0	6.0

Figure 2.6: Variable DV2 - Threshold values and estimates of load weight (kg)

2.1.5 H2O Variable

H2O is the amount of water used during the cycle. The greater the load that must be heated, the higher the steam that must be generated and then more water must be introduced into the system. Figure 2.7 shows the threshold values of the variable H2O and corresponding estimates of load weight (kg).



H2O	350	400	400	450	450	500	550	600	600	670	760	0xffff
Peso	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	6.0

Figure 2.7: Variable H2O - Threshold values and estimates of load weight (kg)

2.1.6 Cycles

The sterilizer has three programs of sterilization (types of cycle): Universal134, Prion134, Universal121. Below, we describe these types of cycle:

- **Universal134**:This is the default sterilization cycle of the sterilizer. The cycle is a type-B sterilization cycle (suitable for all types of loads; solid, porous, hollow A and B; unwrapped, bagged, single or double wrapped) that features a pre-vacuum phase, a plateau phase of 4 minutes at a temperature of 134 ℃and a post vacuum drying phase.
- **Prion134**:This is a special sterilization cycle in accordance with the WHO's recommendations on CJD (Creutzfeld Jakob Disease) for a longer sterilization plateau. The cycle is a type-B sterilization cycle (suitable for all types of loads; solid, porous, hollow A and B; unwrapped, bagged, single or double wrapped) that features a pre-vacuum phase, a plateau period of 18 minutes at a temperature of 134 ℃and a post vacuum drying phase.
- **Universal121**: This is a low-temperature sterilization cycle (121 ℃) primarily designed to sterilize items that cannot withstand the higher temperatures of the 134 ℃default cycle (plastics, textiles). The cycle is a type-B sterilization cycle (suitable for all types of loads; solid, porous, hollow A and B; unwrapped, bagged, single or double wrapped) that features a plateau period of 15 minutes at a temperature of 121 ℃and a post vacuum drying phase.

Figure 2.8 describes the profile of pressure with respect to different types of cycle. All available sterilization cycles feature the same pressure profile as shown in the Fig. 2.8. Only the duration of the plateau period, drying time and the sterilization temperature varies.

However, in this project we have considered the cycle Universal 134 because it is mostly used by dentists.

2.1.7 Phases

Table 2.1 describes the 11 phases that compose a single *run*. Moreover, Figure 2.9

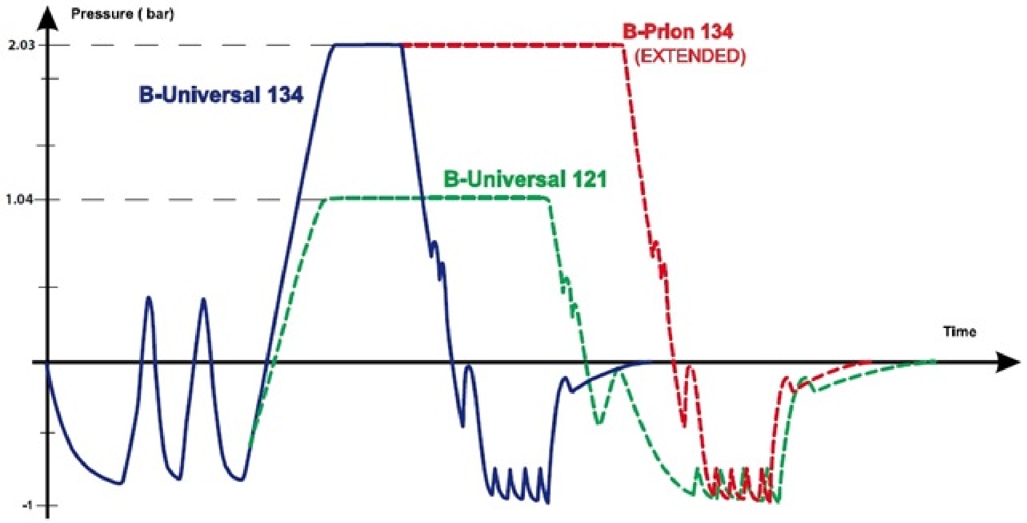


Figure 2.8: Pressure profile of the 3 available type-B sterilization cycles

shows the trend of the Pressure into the chamber (variable PChamb) and highlights the phases of a sterilizer cycle.

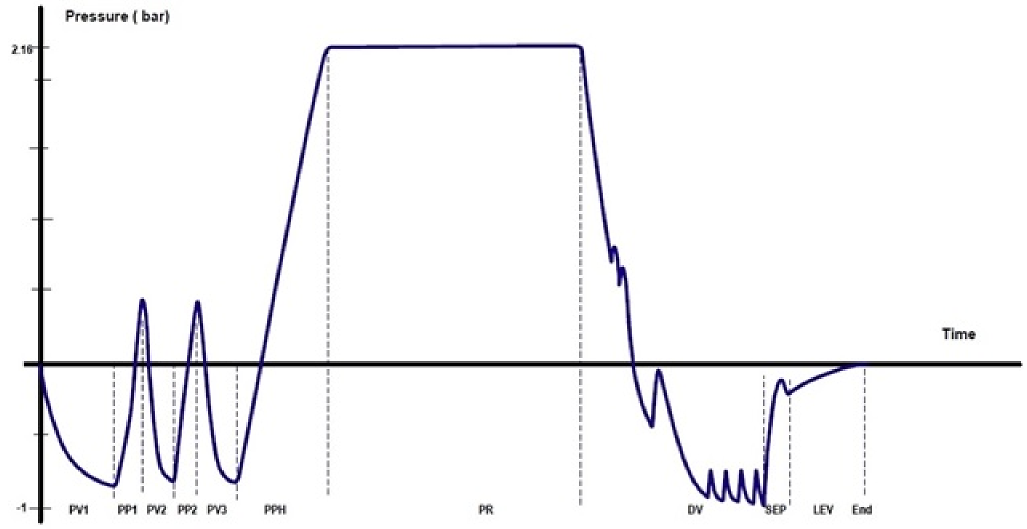


Figure 2.9: Profile of run with its phases

The main phases are the first eight. After the DRY phase there are three phases: SEP (duration 45”); LEV (duration17”-18”); END (it is not always present).
The duration of DRY phase is 360” when the chamber is empty and 960” when the weight load is at least 4 kg.

Phase	**Description**
PV1	The sterilizer creates the vacuum
PP1	Injection of steam into the chamber
PV2	The sterilizer creates the vacuum
PP2	Injection of steam into the chamber
PV3	The sterilizer creates the vacuum
PPH	Injection of steam into the chamber
PR	Sterilization phase
DRY	Dry phase
SEP	Water drain
LEV	Leveling
END	End of cycle

Table 2.1: Run phases description

Finally, we talk about PR phase (*plateau*). This phase represents the actual sterilization of the dentist instruments and its duration is constant (240 seconds in the cycle type Universal 134).

The plateau, shown in Fig. 2.10, is composed of 4 steps (53-54-55-56) and its trend is set by sterilizer software, in particular:

- the *ascents* (micro-phase 1), represent recoveries of H2O that are not entered in the steam generator but in the collecting tray H2O system 2CS (Figure 2.2). In this micro-phase switches to the EVD horizontal (EVD=1) and the steam generator switches on.
- the *descents* (micro-phase 2-3), represent recoveries of H2O that are inserted in the steam generator (Figure 2.3). In this micro-phase, the EVD switches to vertical (EVD=0) and the steam generator is switched off.

Moreover, in the step 56 the steam generator is always switched on.

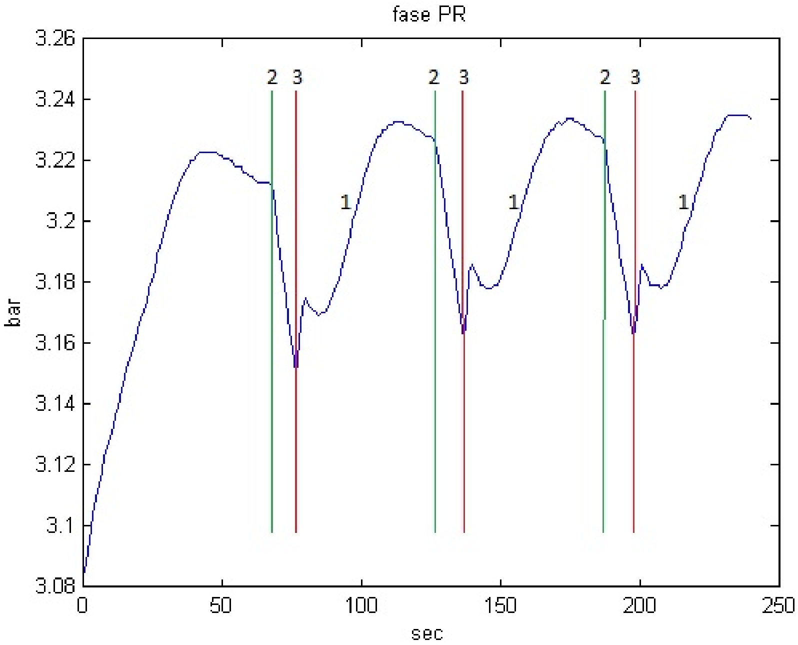


Figure 2.10: Plateau phase and its micro-steps

2.2 Load Types

Table 2.2 reports the types of load that it is possible to insert into the chamber of Lisa522.

Load	**Code**	**Description**
Empty	1	The chamber of sterilizer is empty (0 kg).
Solid	2	The chamber of sterilizer contains only metal.
Porous	3	The chamber of sterilizer contains only porous patches.
Mixed	4	The chamber of sterilizer contains both metal and porous.

Table 2.2: Types of load

According to information of the market, the load types most frequently used by dentists are Solid (only metal) and Mixed (metal and porous patches), moreover 2/3 kg are the weight more frequent. However, to characterize the sterilizer behavior, we performed several sterilization cycles with the following weights: 0 kg (empty sterilizer), 4 kg, 6 kg and 8 kg. It is possible to see more details of the combinations between load types and weights in Fig. 4.1.

2.3 Variables Description

For each sterilization cycle the sensors of our steam sterilizers measure, second by second, some operating parameters or variables, that we describe in Table 2.3. These variables are stored in the dataset *Lisa_ TOT*.

Variabile	**UdM**	**Description**
`TChambInt`	°C	Internal temperature of the chamber.
`TChambExt`	°C	Outside temperature detected near the radial band.
`TAir`	°C	Temperature of the fluid in a point of the circuit.
`TSteamGen`	°C	Temperature of the steam generator.
`TCPU`	°C	Temperature detected near the CPU.
`TCondenser`	°C	Temperature measured on condenser tubes.
`TPowerBoard`	°C	Temperature measured near the card implementation loads.
`PChamb`	bar	Internal pressure of the chamber.
`PowerSteamGen`	kWh	Power applied instantly to the generator.
`PowerChambHeat`	kWh	Power applied instantly to the radial band.
`I24`	mA	Current intensity measured on 24 Volt channel.
`H2O`	cc	Cumulative water that is fed into the generator.

Table 2.3: Variables recorded for each run

Moreover, *Lisa_ TOT* contains another information. In particular we use the following factors to implement the exploratory analysis in Chapter 4:

- `CCycle`: represents the programs of sterilization listed in Section 2.1.6
- `ID_Phase`: represents the phases of each sterilizer cycle showed in Table 2.1.

These factors allow to study and understand the Lisa522 behavior in different phases of a sterilizer cycle.

2.4 Faults

Each product has different modes of failure, therefore an analysis of potential failures helps analysts to understand the impact of potential risks of failure. In particular, *failure analysis* is the process of data collecting and analysis that is useful to determine the cause of a potential failure, in order to improve existing products.

Therefore, we performed an analysis of W&H service reports and selected the sterilizer components for which the defects occur more frequently. In particular, we made the choice of above components taking into account both the frequency of the fault and the technical feasibility of the simulation of these faults in our experiments.

After choosing the above components, we have tampered with them thanks to the help of the W&H technicians. Below, we describe both each components/fault and the manumissions the we have introduced. To verify the proper functioning of the sterilizer we thought to test some of its components in order to determine the sterilizer behavior in normal and altered conditions. Therefore, we have performed the altered runs tampering with, one at a time, the components of Lisa522. In particular, these components are: Door Seal, Pressure Transducer, 2CS, Vacuum Pump, Bacteriological Filter. In next subsections we describe the above components and their manumissions.

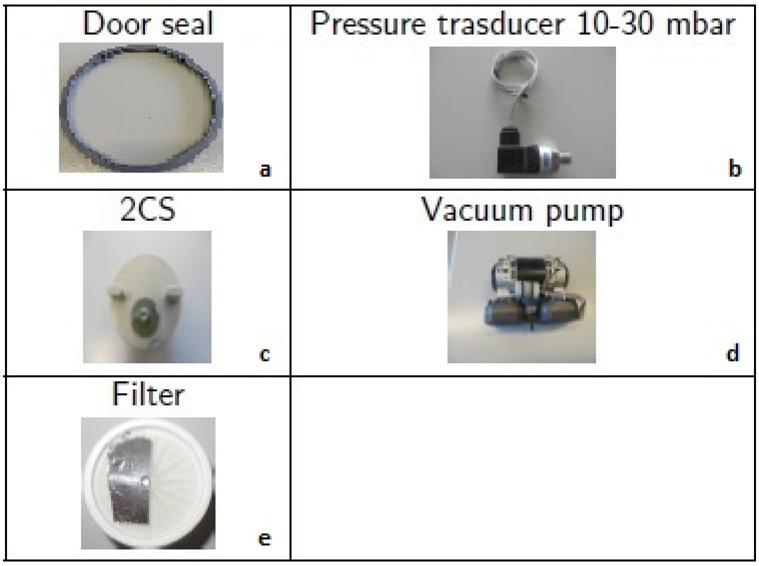


Figure 2.11: Damaged components of Lisa 522

2.4.1 Door Seal

A door seal is a mechanical component which fills the space between two mating surfaces, generally to prevent dispersion from or into the joined objects while under compression. In our case, this component allows to hold pressure and temperature inside the chamber. It is guaranteed for 800 sterilization cycles or one year. An early exchange of the seal can be due to the use of chemical components used by the dentist (e.g., disinfectants under the rules of use of the sterilizer). The seal that we used in our experiments (see Fig. 2.11 a), was used in the laboratory for thousands of sterilization cycles. In particular, the double lip of the seal is worn unevenly. This seal has lost some of its elastic properties and, therefore, the capacity to hold temperature and pressure is not optimal.

2.4.2 Pressure Transducer

A pressure sensor measures pressure, typically of gases or liquids. In our case, this component, shown in Fig. 2.11 b, measures the pressure into the chamber of the sterilizer. We tampered this component by the sterilizer software. In particular we consider two cases (A and B):

- Reference Pressure A: 0.000 absolute bar.
- Pressure Measure A: 0.010 absolute bar (0.030, for code 23).
- Reference Pressure B: 3.500 absolute bar.
- Pressure Measure B: 3.490 absolute bar (3.470, for code 23).

Theoretically, above alterations of this component should influence some algorithms that compare pressure and temperature at a given time.

2.4.3 2CS

2CS is a specific and patented component of Lisa 522. is designed to collect condensate that drops to the bottom of the sterilization chamber during all steam phases of sterilization cycles, and return it to the steam generator to produce more steam. As a result the sterilizer uses considerably less water, and as the returned condensate from the chamber is boiling hot, it also helps to save energy and time. The 2CS process is active during the fractionated pre-vacuum after every steam phase, and during the sterilization (plateau). We tampered the lower diaphragm (green washer) of the 2CS system, puncturing the membrane as shown in Fig. 2.11 c. The effect is that there is no recovery of condensate and water consumption increases.

2.4.4 Vacuum Pump

This component, shown in Fig.2.11 d, removes the air into the sterilizer chamber in order to create saturated steam. The vacuum pump that we have use in the experiments causes poor performance of our sterilizers. In fact, the above fault causes long vacuum phases and then a sterilization cycle more long, moreover, this entails a greater energy consumption.

2.4.5 Bacteriological Filter

A bacteriological filter, shown in Fig.2.11 e, is a filtration component that prevents organisms above a certain size from passing through, limiting the movement of most bacteria. The efficacy of a bacterial filter is determined by the size of the particles that can travel through it. On medical devices bacterial filters limit the spread of microorganisms between patients. Moreover, in our sterilizers this component regulates the input of air into the sterilizer chamber. We occluded to 90% of bacteriological filter. This tampering simulates the effect of a filter clogged with dust. The effect is that the load will not be completely dry at the end of the sterilization cycle.

Chapter 3

Statistical Methods

This Chapter is divided in two parts. In the first part we introduce the Functional Data Analysis (FDA) and recall spline concepts. In particular, we talk about the models that we used to treat the functional data coming from our steam sterilizers. In the second part, we introduce the SPC and some control charts. Moreover, we present our innovative approach called *functional MEWMA* or *f-MEWMA* that integrates splines into the multivariate exponentially weighted moving average scheme. In particular, our approach transforms the multiple data profiles in spline form, thus generating a multivariate functional object which is repeatedly observed over time. This method is capable of monitoring the health of a system which, at each run, generates complex output data given by nonlinear profiles. Finally, we introduce a probabilistic classifier that allows to classify each sterilization cycle with respect to the faults that we described in Section 2.4.

3.1 Functional Data Analysis

In some circumstances is virtually impossible to model the data using traditional parametric techniques, therefore it is necessary to use more flexible regression techniques, in particular nonparametric regression.

Ramsay et al.[15] defined Functional Data Analysis (FDA) as a branch of statistics that analyses data providing information about curves, surfaces or anything else varying over a continuum. The continuum is often the time. The assumption is that these curves are intrinsically smooth and this characteristic often defines functional data analysis. In particular, functional data analyses uses the information about slopes and curvatures of curves, as reflected in their derivatives. Plots of first and second derivatives as functions of t, or plots of second derivative values as functions of first derivative values, may reveal important aspects of the processes generating the data.

The basic philosophy of Functional Data Analysis (FDA) is to consider the observed functional data as single objects, rather than as a sequence of individual observations, see Ramsay et al.[15] and Ramsay and Silverman[16].

Statistical methods for FDA are natural but not obvious extensions of the classical methods applied to multivariate data that do not have simple underlying form. In fact, FDA methods can be used to address complex high dimensional objects that can be characterized as continuous functions. Models for functional data and methods for their analysis may resemble those for conventional multivariate data, including linear and nonlinear regression models, principal components analysis, kernel smoothing, splines, etc. However, the possibility of using derivative information greatly extends the power of these methods. In particular, the use of kernel smoothing and splines ensures the smoothness assumptions of the functional data. Applications and developments of FDA are flourishing, for example, spatial sampling of functional data are considered in Bohorquez et al.[17]. Moreover, functional heteroskedastic regression is used in Fassó et al.[18] for atmospheric profiles based on radiosonde data.

In order to define a suitable functional representation for the sterilization runs, *splines* are reviewed in the rest of this section. Azzalini and Scarpa[19] provide an evocative introduction of splines from an historical viewpoint: *"The term spline originally meant the flexible strips of wood used to shape ships hulls. Some points on the cross-section of the hull were chosen, and the rest of the curve of the hull was derived by forcing the wooden strips to pass through such points, leaving them free to fit into the rest of desired curve according to their natural tendency"*. Indeed, splines are used in mathematics and statistics to build piecewise polynomials which approximate functions known only at certain points called *knots*, according to a logic resembling the mechanics of wood strips described above. In particular, a spline function passes through the knots and it is free at the other points, with the assumption that it is intrinsically smooth.

3.1.1 Spline

To introduce the *spline* we will start with the *straight line regression model*:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \tag{3.1}$$

Figure 3.1 provides a graphical representations of this model. In particular, Figure 3.1(a) depicts Eq. 3.1, with the line representing the underlying regression function and the points representing a typical disposition of the data for this model. Figure 3.1(b) displays the *corresponding basis* for the above model, and then these are the functions 1 and x.

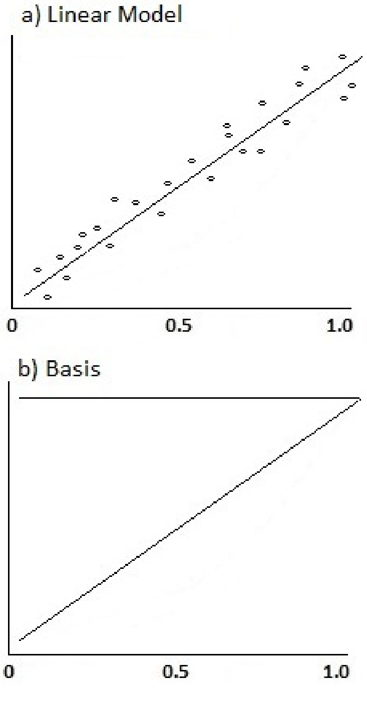


Figure 3.1: Simple linear regression model

Note that the right-hand side of Eq. 3.1 is a linear combination of these functions, which is the reason because we use the word *basis*. In fact, from linear algebra, a *basis* of a vector space is a set V of elements of that space, such that any element of the space can be expressed uniquely as a linear combination of elements of V. For example, $\beta_0 + \beta_1 x$ is a *linear combination* of the basis functions 1 and x. Thus, $[1, x]$ is a basis for the vector space of all linear polynomials in x.

Therefore, the *basis functions* correspond to the columns of the $\boldsymbol{X}$-matrix for fitting the regression are:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

Therefore, the vector of fitted values $\hat{\boldsymbol{y}}$ can be obtained from this matrix and $\boldsymbol{y}$ through the formula

$$\hat{\boldsymbol{y}} = \boldsymbol{X}(\boldsymbol{X'}\boldsymbol{X})^{-1}\boldsymbol{X'}\boldsymbol{y} = \boldsymbol{H}\boldsymbol{y} \tag{3.2}$$

The matrix $\boldsymbol{H}$ is called the *hat matrix* since converts $\boldsymbol{y}$ to $\hat{\boldsymbol{y}}$. A simple extension of the simple linear model is the *quadratic model*:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i \tag{3.3}$$

The Eq. 3.3 is represented in Figure 3.2. It is possible to note that there is an extra basis function: x^2, which corresponds to the addition of the $\beta_2 x_i^2$ term to the model showed in Eq. 3.1.
The $\boldsymbol{X}$-matrix for the quadratic model is:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}$$

and fitted values can be obtained using Eq. 3.2 with this particular $\boldsymbol{X}$.

Now, we see how to accommodate a different type of nonlinear structure. For example, we consider the model showed in Figure 3.3. called *broken stick* model, because it consists of two differently sloped lines that join together at $x = 0.55$.

One mode to handle this type of structure is to introduce a basis function that is zero to the left of 0.55 and then is a positively sloped function from 0.55 onward. The

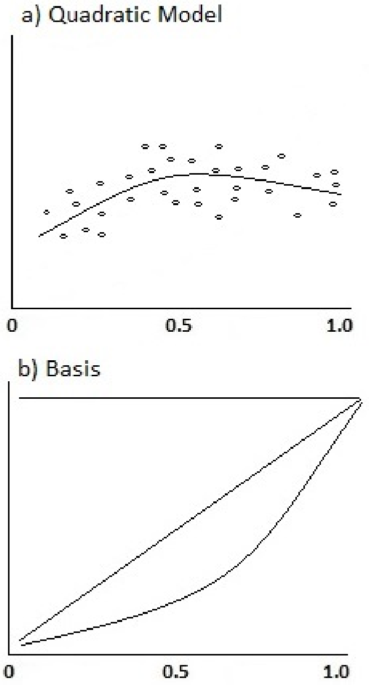


Figure 3.2: Quadratic model

broken line in the top panel of Figure 3.3 can be obtained as a linear combination of the three basis functions in the bottom panel. A compact mathematical way of expressing the new basis function is $(x - 0.55)_+$.

In particular, $(x - 0.55)_+$ is the positive part of the function $(x - 0.55)$ because the '+' sets it to zero for those values of x where $(x-0.55)$ is negative (i.e. $x < 0.55$). A function such as $(x - 0.55)_+$ is also sometimes referred to as a *truncated line* (see Fig. 3.3).

Then, the broken stick model (with a break at $x = 0.55$) is

$$y_i = \beta_0 + \beta_1 x_i + \beta_{11}(x_i - 0.55)_+ + \epsilon_i \tag{3.4}$$

which can be fit using 3.2 with

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 & (x_1 - 0.55)_+ \\ \vdots & \vdots & \\ 1 & x_n & (x_n - 0.55)_+ \end{bmatrix}$$

Now suppose a structure that is more complicated than the broken stick model. About this, Ruppert et al.[20] show an interesting for which there is a straight line

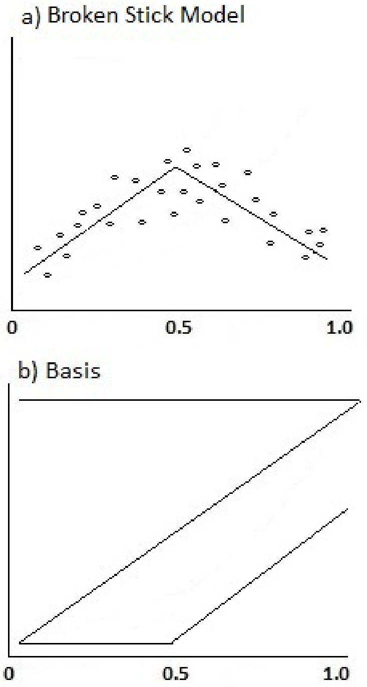


Figure 3.3: Broken stick regression model

structure in the left-hand half, but the right-hand half is prone to a high amount of detailed structure. This model is called *whip model* since the right-hand half is free to move around like the lash of a whip, while the left-hand side corresponds to the whip stiff handle and is linear. Therefore, how could we change the basis? The basis $(x-0.4)_+, (x-0.55)_+, ..., (x-0.85)_+$ will model any *whiplike* structure with a handle between $x=0$ and $x=0.5$. Then we can use ordinary least squares to fit such a model with the $\boldsymbol{X}$-matrix:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 & (x_1-0.4)_+ & (x_1-0.45)_+ & \dots & (x_1-0.85)_+ & \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \\ 1 & x_n & (x_n-0.4)_+ & (x_n-0.45)_+ & \dots & (x_n-0.85)_+ & \end{bmatrix}$$

From this example it is clear that it is possible to handle any complex type of structure simply adding more functions of the form $(x-\kappa)_+$ to the basis or, equivalently, by adding a column of $(x_i-\kappa)_+$ values to the $\boldsymbol{X}$-matrix. The value of κ corresponding to the function $(x-\kappa)_+$ is usually called *knot*. This is because the function is made up of two lines that are *tied together* at $x=\kappa$.

A function such as $(x-0.4)_+$ is called *linear spline basis function* and a set of such functions is called a *linear spline basis*. The spline model is:

$$f(x) = \beta_0 + \beta_1 x + \sum_{i=1}^{K} u_i (x - k_i)_+. \tag{3.5}$$

In general, we choose k points $k_1 < k_2 < ... < k_K$ called *knots* along the x-axis. A function $f(x)$ is constructed so that it passes exactly through the knots and is free at the other points, with the constraint that it presents regular overall behavior.

Usually, if we need smooth and accurate derivatives, it is necessary to increase the order of the spline. Ramsay et al.[15] say that, a practical rule is to impose the order of the spline basis to be at least two higher than the highest order derivative to be used. Therefore, by this rule, a *cubic spline* is a good choice.

Azzalini and Scarpa[19] also note that the degree that is almost universally used is $p = 3$, and we therefore speak of cubic splines. The reason for this is that the human eye cannot perceive discontinuity in the third derivative.

Therefore, after the quadratic model, a further extension of the simple linear model is the *cubic model*:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i \tag{3.6}$$

The $\boldsymbol{X}$-matrix for the cubic model is:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{bmatrix}$$

and fitted values can be obtained using Eq. 3.2 with this particular $\boldsymbol{X}$.

Since the function $(x - k_i)_+^p$ has $p - 1$ continuous derivatives, higher values of p lead to smoother spline functions. In this work we chose the *cubic spline* in order to insure an adequate grade of smoothness. Therefore, starting to the Eq. 3.5, a polynomial spline of degree 3 is based on the following piecewise polynomial

$$m(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{i=1}^{K} u_i (x - k_i)_+^3 \tag{3.7}$$

where K is the number of knots.

In Figure 3.4 we report an example of functional data related to the sterilizer case study. In particular, this plot shows the functional form (cubic spline) of the variable `TChambInt`, with 30 knots, or ten for each macro-phase, in 82 fault free sterilization runs.

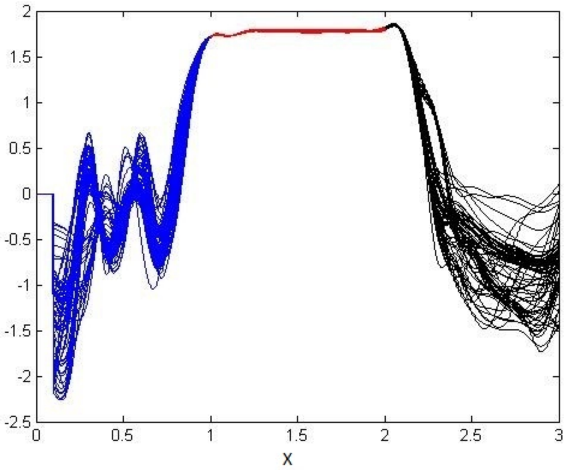


Figure 3.4: Cubic Splines of `TChambInt` variable measured at standardized times x for 82 fault free sterilization runs.

The fundamental block of spline functions are the so called spline basis functions. To see this, let us consider the positive part function, given by $(x)_+ = x$ if $x > 0$ and $(x)_+ = 0$ else. Using this, a linear spline basis si given by $1, x, (x-k_1)_+, ..., (x-k_K)_+$ and any linear combination of such basis is a piecewise linear function with knots at $k_1, ..., k_K$. More generally, a polynomial spline of degree p is based on the following piecewise polynomial that is the generalization of the Eq. 3.5

$$m(x) = \beta_0 + \beta_1 x + ... + \beta_p x^p + \sum_{i=1}^{K} u_i (x - k_i)_+^p \tag{3.8}$$

where K is the number of knots. Moreover, the unknown coefficient vectors $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)'$ and $\boldsymbol{u} = (u_1, ..., u_k)'$ define, respectively, the global and local behavior.

Finally, for more elaborate penalized spline models there are computational advantages to keeping the number of knots relatively low. A reasonable default is to choose the knots to ensure that there are a fixed number of unique observations, say 4-5, between each knot. For large data sets this can lead to an excessive number of knots, so a maximum number of allowable knots (say, 20-40 total) is recommended.

A simple default choice of K that usually works well is

$$K = min\left(\frac{1}{4}c, 35\right) \tag{3.9}$$

where c is number of unique x_i.

However, in our case study, the choice of knots does not affect the detection ability of anomaly by *f-MEWMA* control chart.

3.1.2 Linear Mixed Models

Gurrin et al.[21] say that linear mixed models extend the linear regression model for longitudinal data. In order to see the connection among splines and linear mixed model, let us consider n observations $\boldsymbol{y} = (y_1, ..., y_n)'$ at times $\boldsymbol{x} = (x_1, ..., x_n)'$, such that $\boldsymbol{y} = m(\boldsymbol{x}) + \boldsymbol{\epsilon}$ where $m()$ is as in Equation (3.8) and $\boldsymbol{\epsilon}$ is a Gaussian error with variance covariance matrix $\sigma_\epsilon^2 \boldsymbol{I}$. Moreover, let us introduce the $n \times p$ design matrix of fixed effects:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{1}, & \boldsymbol{x}, & ..., & \boldsymbol{x}^p \end{bmatrix} \tag{3.10}$$

and the $n \times K$ design matrix of random effects

$$\boldsymbol{Z} = \left[(\boldsymbol{x} - \kappa_1 \boldsymbol{1})_+, ..., (\boldsymbol{x} - \kappa_k \boldsymbol{1})_+\right]. \tag{3.11}$$

Following e.g. Gurrin et al.[21] and references therein, the standard form of the linear mixed model is

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{Zu} + \boldsymbol{\epsilon} \tag{3.12}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{u}$ are, respectively, the fixed and random effects introduced in the previous section. Moreover, the random effect vector $\boldsymbol{u}$ is assumed Gaussian with zero mean and variance covariance matrix given by $\boldsymbol{G}$.

3.1.3 Penalized Least Squares

The quality of a nonparametric function estimated by polynomial splines depends on the number of knots, but there are two strategies to bypass this problem:

- The adaptive choice of knots based on model choice strategies.

- The regularization of the estimation problem through the introduction of roughness penalties, that allows to constrain the knots influence.

Usually it is preferred to use the second strategy. The basic idea of *penalized splines* (*P-Splines*) can be synthesized as follows:

1. Approximate a function $f(x)$ with a polynomial spline that uses a large number of knots. This ensures that $f(x)$ can be approximated with enough flexibility to represent even highly complex functions.

2. Insert an additional *penalty term* that prevents overfitting and minimize the *penalized least squares (PLS)* criterion instead of the traditional least squares criterion.

Interestingly, thanks to the above representation, see Green[22], the maximum Gaussian likelihood smoothing spline estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{u}}$ minimize the penalized least squares (PLS) function

$$\boldsymbol{PLS(\beta, u)} = ||\boldsymbol{y} - \boldsymbol{X\beta} - \boldsymbol{Zu}||^{\mathbf{2}} + \alpha||\boldsymbol{u}||^{\mathbf{2}} \tag{3.13}$$

where α represents the ratio of variance components, $\alpha = \sigma_\epsilon^2/\sigma_u^2$, and $||\boldsymbol{u}||$ is the Euclidean norm of the vector $\boldsymbol{u}$. This provides fitted curves $\hat{y}(x) = \hat{m}(x)$ that are spline smoothers, hence formally linking linear mixed models to spline smoothing. Equation (3.13) highlights that $\alpha = \sigma_\epsilon^2/\sigma_u^2$ is a smoothing parameter. If value of α increases then the penalty term receives greater weight and the regression becomes smoother. Optimization of (3.13) is performed using standard software such as R or MATLAB/OCTAVE, which implements Reinsch[23] approach. As a result we get estimates $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{y}}$.

In order to assess the goodness of fit of the spline model used, various quantities are available. From the practical point of view a popular choice is the mean square error:

$$MSE(\hat{\boldsymbol{y}}) = E[diag[(\boldsymbol{y} - \hat{\boldsymbol{y}})(\boldsymbol{y} - \hat{\boldsymbol{y}})']] \tag{3.14}$$

In the application of next section we use cubic smoothing splines with $p = 3$ and $\alpha = 1$.

3.2 Statistical Surveillance

As mentioned, statistical surveillance is the continuing collection, analysis and interpretation of data concerning the health of a system and its main instruments are control charts that are also used in statistical process control. These instruments have been traditionally used to monitor industrial processes. However, they are increasingly being used in many applications fields such as engineering, economics, finance, epidemiology and environmental statistics.

In particular, above instruments help to detect if the process shifts, in order to adopt corrective actions before many nonconforming "units" (in our case sterilization cycles) are performed.

Therefore, in next Section, we introduce the principal control charts and introduce our innovative *f-MEWMA*.

3.2.1 Introduction to Control Charts

In general, a control chart (see Figure 3.5) is a graphical instrument of a quality characteristic that has been measured or computed from a sample versus the sample number or time (Montgomery[24]). Usually, a control chart contains:

- Center line (CL) that represents the average value of the quality characteristic corresponding to the in-control state.
- Two other horizontal lines, called the upper control limit (UCL) and the lower control limit (LCL), are also shown on the chart.

Above control limits are chosen base on the process is in control, then the sample points will fall between them. If these points plot within the control limits, the process is assumed to be in control. However, if a point falls outside of the control limits then that is interpreted as a "signal" that the process is out of control, and investigation and corrective action are required in order to find and eliminate the causes responsible of this behavior.

It is possibile give a general model for a control chart. Let w be a sample statistic that measures some quality characteristic of interest, and suppose that the mean of

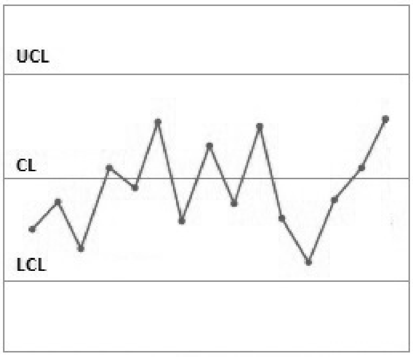


Figure 3.5: A typical control chart

c is μ_c and the standard deviation of w is σ_c (Montgomery[24]). Then the center line, the Upper Control Limit, and the Lower Control Limit become:

$$\text{UCL}= \mu_c + L\sigma_c$$
$$\text{Center line}= \mu_c$$
$$\text{LCL}= \mu_c - L\sigma_c$$

where L is the "distance" of the control limits from the center line, expressed in standard deviation units. This general theory of control charts was first proposed by Walter A. Shewhart, and control charts developed according to these principles are often called **Shewhart control charts**. Montgomery[24] says that an important disadvantage of a Shewhart control chart is that it uses only the information about the process contained in the last sample observation and it ignores any information given by the entire sequence of points. This feature makes the Shewhart control chart relatively insensitive to small process shifts, say, on the order of about 1.5σ or less. This potentially makes Shewhart control charts less useful in phase II monitoring problems.

Two very effective alternatives to the Shewhart control chart may be used when small process shifts are of interest: the cumulative sum (CUSUM) control chart, and the exponentially weighted moving average (EWMA) control chart.

3.2.2 EWMA Control Chart

The *exponentially weighted moving average* (EWMA) control chart is also a good alternative to the Shewhart control chart when we are interested in detecting small shifts. The performance of the EWMA control chart is approximately equivalent to that of the cumulative sum control chart, and in some ways it is easier to set up and operate.

The EWMA control chart was introduced by Roberts[25] and it is very effective with respect to small process shifts. The exponentially weighted moving average is defined as

$$q_i = \lambda y_i + (1 - \lambda) q_{i-1} \tag{3.15}$$

where $0 < \lambda \leq 1$ is a constant and the starting value is the process mean, then that $q_0 = \mu_0$. In some cases the average of preliminary data is used as the starting value of the EWMA, then that $q_0 = \bar{x}$. The weight is $1 - (1 - \lambda)^i$.

If the observations y_i are independent random variables with variance σ^2, then the variance of q_i is

$$\sigma_{q_i}^2 = \sigma^2 \left(\frac{\lambda}{2 - \lambda} \right) [1 - (1 - \lambda)^{2i}] \tag{3.16}$$

Therefore, the EWMA control chart is constructed by plotting q_i versus the sample number i (or time). The center line is μ_0. Instead, the control limits of the EWMA, for i is large, control chart are

$$UCL = \mu_0 + L\sigma \sqrt{\frac{\lambda}{2 - \lambda} [1 - (1 - \lambda)^{2i}]} \tag{3.17}$$

$$LCL = \mu_0 - L\sigma \sqrt{\frac{\lambda}{2 - \lambda} [1 - (1 - \lambda)^{2i}]} \tag{3.18}$$

where L is the width of the control limits. Note that the term $[1 - (1 - \lambda)^{2i}]$ approaches unity as i gets larger. This means that after the EWMA control chart has been running for several time periods, the control limits will approach steady-state values given by

$$UCL = \mu_0 + L\sigma\sqrt{\frac{\lambda}{2-\lambda}} \tag{3.19}$$

$$LCL = \mu_0 - L\sigma\sqrt{\frac{\lambda}{2-\lambda}} \tag{3.20}$$

However, it is recommend to use the exact control limits in equations 3.17 and 3.18 for small values of i, because this improves the performance of the control chart in detecting.

Finally, Montgomery[24] suggests that values of λ between 0.25 and 0.05 work well in practice, with $\lambda = 0.05$, $\lambda = 0.10$, and $\lambda = 0.20$ that are the most popular choices. A good rule of thumb is to use smaller values of λ to detect smaller shifts. Finally, $L = 3$ works reasonably well.

3.2.3 The Multivariate Quality Control Problem

There are many situations in which the simultaneous monitoring or control of two or more related quality characteristics (variables) is necessary. For example, suppose that a sterilizer measures the variables Temperature (y_1) and Pressure (y_2).

Because both quality above variables are measurements, they could be monitored by applying the a control chart to each variable. The process is considered to be in control only if the sample means fall within their respective control limits. In particular, this is equivalent to the pair of means ($\bar{y}_1, \bar{y}_2$) plotting within the shaded region in Fig. 3.6.

However, monitoring these two variables independently can be very misleading. In fact, for example, in Figure 3.6, it is possible to note that one observation appears somewhat different with respect to the others. That point is within in the control limits on both of the univariate charts for (y_1) and (y_2), however, when we examine the two variables simultaneously, the unusual behavior of the point is fairly obvious.

Process-monitoring problems in which several related variables are of interest are sometimes called multivariate quality-control (or process-monitoring) problems Montgomery[24]. The original work in multivariate quality control was done by Hotelling[26].

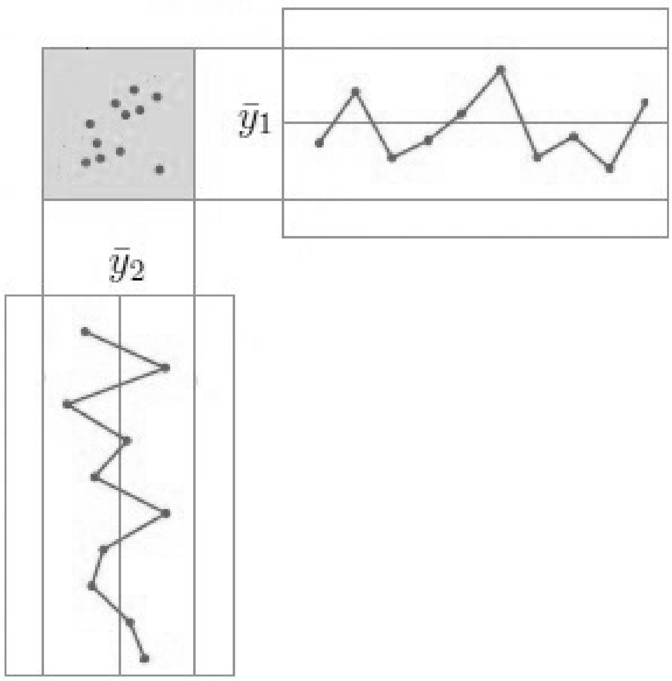


Figure 3.6: Control region by independent control limits.

3.2.4 Description of Multivariate Data

In univariate statistical quality control, it generally use the normal distribution to describe the behavior of a continuous quality characteristic (Montgomery[24]). The univariate normal probability density function is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{3.21}$$

The mean of the normal distribution is μ and the variance is σ^2. It is possible to note that (apart from the minus sign) the term in the exponent can be written in this way

$$(x - \mu)(\sigma^2)^{-1}(x - \mu) \tag{3.22}$$

Above quantity measures the squared standardized distance from x to the mean μ, where the term "standardized" means the distance expressed in standard deviation units.

This same approach can be used in the multivariate normal distribution case. In fact, we suppose to have p variables, called $x_1, x_2, ..., x_v$, that we put in a v-component vector $\mathbf{x}' = [x_1, x_2, ..., x_v]$. Let $\mathbf{m} = [m_1, m_2, ..., m_v]$ be the vector of the means of the above variables, and let $\mathbf{\Sigma}$ the variances-covariances matrix, where the main diagonal elements of are the variances of the x's and the off-diagonal elements are the covariances. Moreover, the squared standardized (generalized) distance from $\mathbf{x}$ to μ is

$$(\mathbf{x} - \mu)'\mathbf{\Sigma^{-1}}(\mathbf{x} - \mu) \tag{3.23}$$

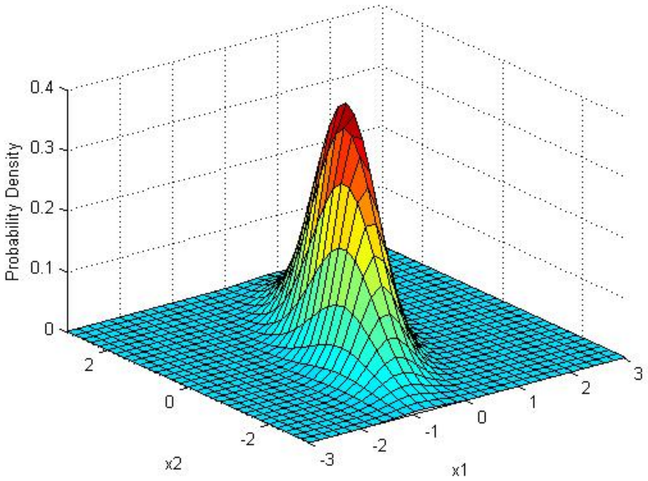


Figure 3.7: A multivariate (bivariate) normal distribution with $v = 2$ variables.

The multivariate normal density function is obtained by replacing the standardized distance shown in equation 3.23 by the multivariate generalized distance in equation 3.22 and changing the constant term with a more general form that makes the area under the probability density function unity regardless of the value of v. Thus, the *multivariate normal* probability density function is

$$f(x) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{3.24}$$

where $-\infty < x_j < +\infty$, $j = 1, 2, ..., v$

In Fig. 3.7 is shown a multivariate normal distribution for $v = 2$ variables (called bivariate normal). It is possibile to note that the density function is a surface.

3.2.5 Multivariate EWMA Control Chart

In recent decades, multivariate statistical methods for monitoring and controlling complex processes have received increasing attention. The first work in multivariate quality control dates back to Hotelling[26], while Montgomery[24] and Woodall and Montgomery[27] provide recent overviews of multivariate control charts and a perspective of applications of statistical process monitoring in different fields.

In order to define the traditional multivariate EWMA of Lowry et al.[4], for each run i, let us suppose that observations $\boldsymbol{y_i}$ are v dimensional random vectors, with *in-control* zero mean and variance covariance matrix $\boldsymbol{\Sigma}$. Also, suppose we want detect a drift intervening at time i_0 in the process mean $E(\boldsymbol{y_i}) \neq \mathbf{0}$ for $i \geq i_0$. In this frame, a MEWMA control chart is based on exponentially smoothed observations:

$$\boldsymbol{Q}_i = \lambda \boldsymbol{y}_i + (1 - \lambda)\boldsymbol{Q}_{i-1} \tag{3.25}$$

where $0 \leq \lambda \leq 1$, $\boldsymbol{Q_0} = \mathbf{0}$. Then the detector is defined by

$$T_i^2 = \boldsymbol{Q_i'} \boldsymbol{\Sigma_Q^{-1}} \boldsymbol{Q_i} \tag{3.26}$$

where $\boldsymbol{\Sigma_Q}$ is the $v \times v$ (asymptotic) variance covariance matrix of $\boldsymbol{Q}_i$

$$\boldsymbol{\Sigma_Q} = \frac{\lambda}{2 - \lambda} \boldsymbol{\Sigma}. \tag{3.27}$$

The MEWMA control chart is then obtained by plotting T_i^2 against i and giving an out-of-control signal if $T_i^2 > h$. The upper control limit $h > 0$ is related to detection delay, false alarms and the so called average run length as discussed Prabhu and Runger[28]. In order to have a simple approximation, it can be observed that, if we assume that $\boldsymbol{y}_i$ is Gaussian distributed, so is, asymptotically, also $\boldsymbol{Q}_i$ and, hence, $\boldsymbol{T}_i^2$ is approximately χ^2 distributed with v degrees of freedom. Hence, h can be initially computed as the 0.999 quantile of the χ^2 distribution.

3.3 f-MEWMA Control Chart

In this section we introduce the f-MEWMA control chart which extends the traditional MEWMA and allows to monitor multiple nonlinear profiles using splines.

Specifically, the proposed approach transforms each sterilization run in a multivariate functional object. Since the random effects $\boldsymbol{u}_i$ are zero mean Gaussian random vectors with zero mean in the fault free case, we apply a MEWMA control chart to $\boldsymbol{u}_i$ in order to detect a generic drift of these objects over time. In particular, the algorithm is iterative for $i = 1, ..., r$ and, for each i, it consists of the following two steps:

1. f-step At run i the observed $n_i \times v$ dimensional profile matrix $\boldsymbol{y}_i$ is transformed in spline form obtaining the estimated spline coefficients. In particular Equation (3.8) is used with $p = 3$ giving v cubic splines, one for each column component of $\boldsymbol{y}_i$. Then the vector $\boldsymbol{b}_i$ is obtained by stacking the v vectors of random coefficients $\hat{\boldsymbol{u}}_{i,j}, j = 1, ..., v$.

2. MEWMA-step Vector $\boldsymbol{b}_i$ is considered as a Gaussian random vector with variance covariance matrix Σ and processed in order to generate *f-MEWMA* control chart. In particular, we calculate the quantity

$$\boldsymbol{Q}_i = \lambda \boldsymbol{b}_i + (1 - \lambda)\boldsymbol{Q}_{i-1}. \tag{3.28}$$

Next, exploiting Equation (3.26), the functional detector T_j^2 is obtained and plotted against i to depict the f-MEWMA control chart, giving a functional out-of-control signal if $T_i^2 > h$ for some $h > 0$.

Notice that, in case we have a univariate profile, i.e. $v = 1$, above algorithm defines a f-EWMA control chart. Moreover, since $\boldsymbol{b}_i$ may be assumed to have a $v \times K$ Gaussian distribution, then, similarly to Section 3.2.5, the threshold h may be computed as the 0.999 quantile of the χ^2 distribution with $v \times K$ degrees of freedom.

3.4 Classification

In statistics and machine learning, *classification* is the problem related to identify to which of a set of categories an observation belongs, on the basis of a training set of data containing observations (or instances) whose category is known. An example would be assigning a given email into "spam" or "non-spam" classes. In the terminology of machine learning, classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available(Alpaydin[29]).

Often, the individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or features. These properties may be categorical (e.g. "A", "B", "AB" or "O", for blood type), ordinal (e.g. "large", "medium" or "small") and so on. Other classifiers work by comparing observations to previous observations by means of a similarity or distance function. An algorithm that implements classification is known as a *classifier.*

A common subclass of classification is probabilistic classification. Algorithms of this nature use statistical inference to find the best class for a given instance (observation). Probabilistic algorithms return a probability of the instance being a member of each of the possible classes. The best class is normally then selected as the one with the highest probability.

In next Section we introduce a probabilistic classification method that exploits the Bayes formula.

3.4.1 Bayes Classifier

Bayesian classification procedures provide a natural way of taking into account any available information about the relative sizes of the sub-populations associated with the different groups within the overall population. Bayesian procedures tend to be computationally expensive and, in the days before Markov chain Monte Carlo computations were developed, approximations for Bayesian clustering rules were devised (Binder[30]).

In our case, we verified the ability to identify the type of fault (anomaly) using probabilistic classification technique, in particular Bayes[31] formula:

$$P(A_i|E) = \frac{P(E|A_i)P(A_i)}{\sum_{j=1}^{n} P(E|A_j)P(A_j)} \tag{3.29}$$

where

- $P(A)$ is the *prior* probability of each fault.

- $P(A_i|E)$ is the *posterior* probability of each fault since a given run (E).

Our classifier is based on functional data, in fact it receives in input the spline coefficients. For each run, we calculated the *posterior probability* of each fault and the diagnosis is made by identifying the the most likely fault. This approach requires the definition of *prior probabilities* of the various anomalies that, in this experimental context, we considered equal to the frequencies used in DOE (see Fig. 4.1).

To obtain the faults classification we used the *leave-one-out* technique. In particular, we exclude one run at a time from the dataset and, then, we classify it according to criterion of the the most likely fault.

Chapter 4

Data, Application and Results

This work is based on a specific requirement of W&H Sterilization to control the health of the sterilizer called *Lisa522 Fully Automatic*. This machine allows dentists to sanitize dental instruments that they use for curing their patients. Thus, it is important to understand if a sterilizer is working well and detect the early onset of anomalies.

Therefore, in this Chapter, we present an exploratory analysis of the variables measured by sterilizers sensors in order to understand the machine behaviour with respect to different types of load and types of fault. Moreover, we show how to implement our *f-MEWMA* control chart and compare it with the results of traditional MEWMA. Finally, we present the results related to the probabilistic runs classifier.

4.1 Exploratory Analysis

Our study is focused on the sterilizer Lisa522 Fully Automatic. As mentioned, at the end of each run this sterilizer produces a *scl* file containing operating parameters (*variables* and *factors*) that we described in Table 2.3.

We are here interested in understanding which are the measurable "physical" consequences of the anomalies introduced in some components of our machines, as described in Section 2 and variables of Table 2.3. Hence the first step is to consider if these anomalies could generate mean differences among good and altered sterilization runs for some variables such as temperature, pressure or sterilization time. To see this, we considered overall means, macro-phase means or micro-phase means for all variables.

Because doubtfully, the above variables have Gaussian distributions, we implemented both standard and nonparametric ANOVA (Kruskal-Wallis test) but we got no clearly significant and stable results, with load playing a major role comparing to the six alteration factors.

This is consistent with the fact that these sterilizing machines have strict quality requirements, ensuring that the dental equipments have been properly sterilized. Hence mean pressure, temperature and time are unchanged under moderate aging and misuse.

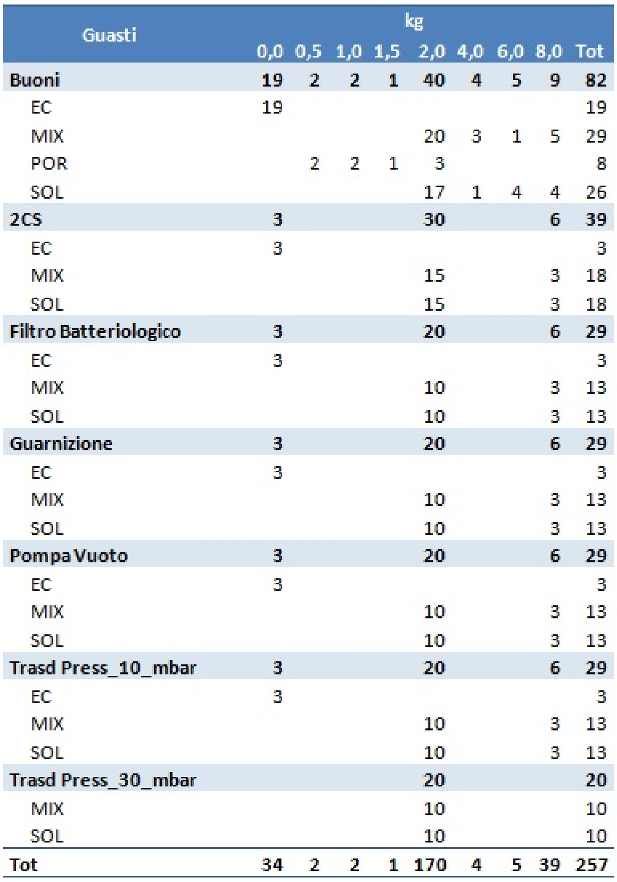
Considering dynamics instead of level, the second approach tried to compare variances. But once again no clearly significant results emerged. If an anomaly is to alter dynamics instead of level or variance, than one could check if correlation among physical quantities, e.g. pressure and temperature, is changed by modifying e.g. door seal. To see this, a correlation analysis between the variables described in Table 2.3 was implemented comparing correlations for good and altered runs. For example the correlation between pressure and temperature has been averaged for the failure free and altered runs. Once again no significant differences among these average correlations were found.

To verify the sterilizer behavior we planned a design of experiments consisting of *good runs*, where all components of sterilizer work properly, and *altered runs* made by tampering, one at a time, the following components of our sterilizer: Door Seal, Pressure Transducer, 2CS, Vacuum Pump and Bacteriological Filter.

Therefore, this section describes the more important aspects of preliminary data analysis. In particular, we implement descriptive analysis for each variable using a dedicated software functions and, by nonparametric ANOVA, we compare the behavior of the sterilizer when it works in normal conditions with respect to when a component is altered.

4.1.1 Data Description

In order to understand the sterilizer behavior, we performed several sterilization runs over three new machines with same characteristics. In particular, we followed a quasi balanced DOE involving six factors which are related to critical sterilizer components, such as door seals, transducers and filters, among others. Moreover confounders such as machine and load have been randomized. Each factor was considered at two levels: fault free vs altered. Although no interactions where considered among above six factors, machine and load where found not interacting significantly with the factors.



Guasti	kg								
	0,0	0,5	1,0	1,5	2,0	4,0	6,0	8,0	Tot
Buoni	**19**	**2**	**2**	**1**	**40**	**4**	**5**	**9**	**82**
EC	19								19
MIX					20	3	1	5	29
POR		2	2	1	3				8
SOL					17	1	4	4	26
2CS	**3**				**30**			**6**	**39**
EC	3								3
MIX					15			3	18
SOL					15			3	18
Filtro Batteriologico	**3**				**20**			**6**	**29**
EC	3								3
MIX					10			3	13
SOL					10			3	13
Guarnizione	**3**				**20**			**6**	**29**
EC	3								3
MIX					10			3	13
SOL					10			3	13
Pompa Vuoto	**3**				**20**			**6**	**29**
EC	3								3
MIX					10			3	13
SOL					10			3	13
Trasd Press_10_mbar	**3**				**20**			**6**	**29**
EC	3								3
MIX					10			3	13
SOL					10			3	13
Trasd Press_30_mbar					**20**				**20**
MIX					10				10
SOL					10				10
Tot	**34**	**2**	**2**	**1**	**170**	**4**	**5**	**39**	**257**

Figure 4.1: DOE for three Lisa 522 using different faults and loads

At the end, we performed 257 runs, with the following characteristics:

- 82 good runs: in this case the sterilizing machine do not present any induced anomaly.

- 175 altered runs: in this case we tampered with critical components, one at a time, almost 30 runs per factor.

Fig. 4.1 shows the composition of design of experiments. The load types used to realize above runs are: Solid (metal only), Porous (porous patches) and Mixed (metal and porous patches). Each run has a different duration which can range between 30 and 45 minutes, according to type of load.

According to market information, we know that the 2 kg is the load weight most used by the dentists. However, we have performed runs with further weights: 0 kg (empty chamber), 0.5 kg, 1.5 kg, 4 kg, 6 kg and 8 kg.

4.1.2 Correlation Analysis

Correlation analysis measures the relationship between two variables, for example `TChambInt` and `PChamb`. To implement this analysis we report the following cases with respect to the load type *Empty*, in order to study the physiological behavior (when the chamber is empty) of our sterilizers:

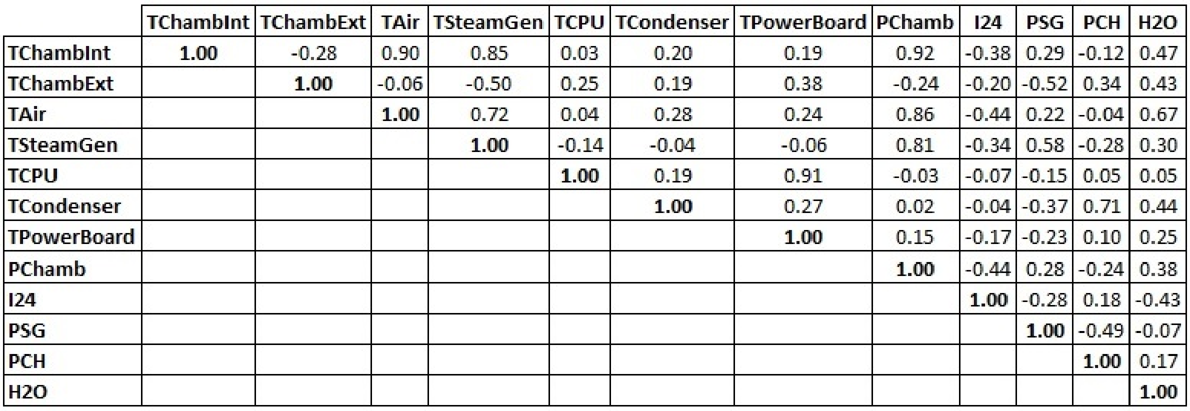
- Correlation between variables with fault= no fault (good runs).
- Correlation between variables with fault=Door Seal.
- Correlation between variables with fault=Pressure Transducer.
- Correlation between variables with fault=2CS.
- Correlation between variables with fault=Vacuum Pump.
- Correlation between variables with fault=Bacteriological Filter.

In particular, this analysis have two aims:

- Calculate the correlation between the variables that the sterilizer measures for each run.
- Compare the correlations between good runs and altered runs.

Therefore, we report the correlation matrices related to each type of fault and highlight the highest correlations choosing a reasonable value $r \geq 0.85$, taking into account that the average of the correlation between `TChambInt` and `PChamb` is 0.92.

Fault: No fault - Load: EC, 0 kg



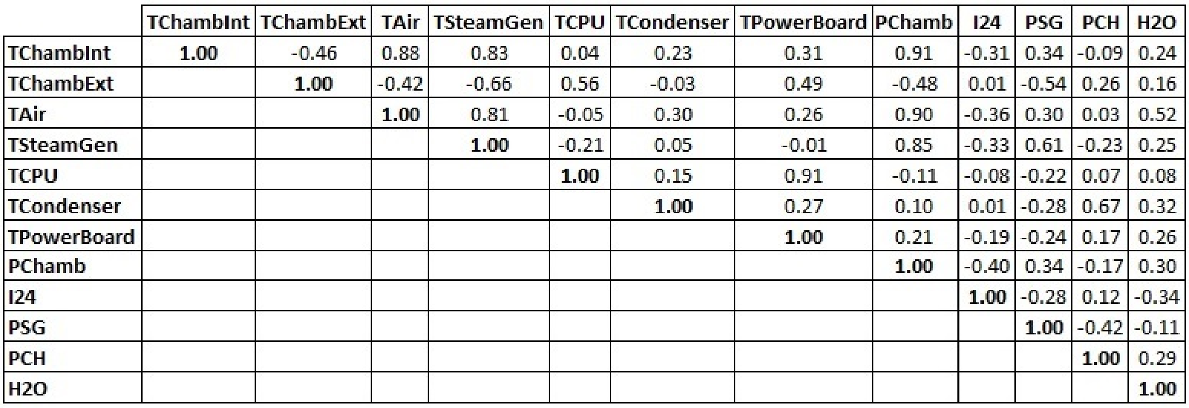
	TChambInt	TChambExt	TAir	TSteamGen	TCPU	TCondenser	TPowerBoard	PChamb	I24	PSG	PCH	H2O
TChambInt	**1.00**	-0.28	0.90	0.85	0.03	0.20	0.19	0.92	-0.38	0.29	-0.12	0.47
TChambExt		**1.00**	-0.06	-0.50	0.25	0.19	0.38	-0.24	-0.20	-0.52	0.34	0.43
TAir			**1.00**	0.72	0.04	0.28	0.24	0.86	-0.44	0.22	-0.04	0.67
TSteamGen				**1.00**	-0.14	-0.04	-0.06	0.81	-0.34	0.58	-0.28	0.30
TCPU					**1.00**	0.19	0.91	-0.03	-0.07	-0.15	0.05	0.05
TCondenser						**1.00**	0.27	0.02	-0.04	-0.37	0.71	0.44
TPowerBoard							**1.00**	0.15	-0.17	-0.23	0.10	0.25
PChamb								**1.00**	-0.44	0.28	-0.24	0.38
I24									**1.00**	-0.28	0.18	-0.43
PSG										**1.00**	-0.49	-0.07
PCH											**1.00**	0.17
H2O												**1.00**

Figure 4.2: Correlation matrix for good runs.

The highest correlations (≥ 0.85) are between the variables:

- `TChambInt` and `TAir`, $r = 0.90$.
- `TChambInt` and `TSteamGen`, $r = 0.85$.
- `TChambInt` and `PChamb`, $r = 0.92$.
- `TCPU` and `TPowerBoard`, $r = 0.91$.
- `PChamb` and `TAir`, $r = 0.86$.

Fault: Door Seal - Load: EC, 0 kg



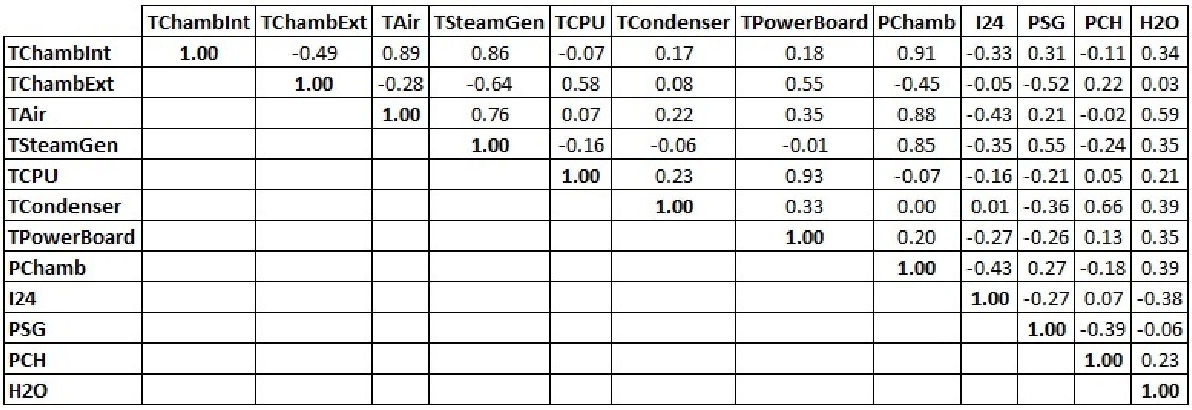
	TChambInt	TChambExt	TAir	TSteamGen	TCPU	TCondenser	TPowerBoard	PChamb	I24	PSG	PCH	H2O
TChambInt	**1.00**	-0.46	0.88	0.83	0.04	0.23	0.31	0.91	-0.31	0.34	-0.09	0.24
TChambExt		**1.00**	-0.42	-0.66	0.56	-0.03	0.49	-0.48	0.01	-0.54	0.26	0.16
TAir			**1.00**	0.81	-0.05	0.30	0.26	0.90	-0.36	0.30	0.03	0.52
TSteamGen				**1.00**	-0.21	0.05	-0.01	0.85	-0.33	0.61	-0.23	0.25
TCPU					**1.00**	0.15	0.91	-0.11	-0.08	-0.22	0.07	0.08
TCondenser						**1.00**	0.27	0.10	0.01	-0.28	0.67	0.32
TPowerBoard							**1.00**	0.21	-0.19	-0.24	0.17	0.26
PChamb								**1.00**	-0.40	0.34	-0.17	0.30
I24									**1.00**	-0.28	0.12	-0.34
PSG										**1.00**	-0.42	-0.11
PCH											**1.00**	0.29
H2O												**1.00**

Figure 4.3: Correlation matrix for Door Seal runs.

The highest correlations ($r \geq 0.85$) are between the variables:

- `TChambInt` and `TAir`, $r = 0.88$.
- `TChambInt` and `PChamb`, $r = 0.91$.
- `TCPU` and `TPowerBoard`, $r = 0.91$.
- `PChamb` and `TAir`, $r = 0.90$.
- `PChamb` and `TSteamGen`, $r = 0.85$.

Fault: Pressure Transducer - Load: EC, 0 kg



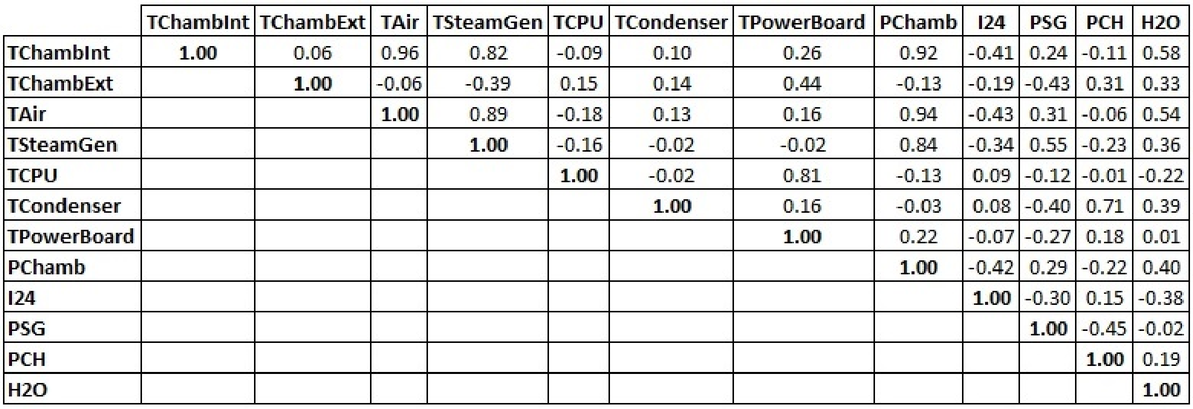
	TChambInt	TChambExt	TAir	TSteamGen	TCPU	TCondenser	TPowerBoard	PChamb	I24	PSG	PCH	H2O
TChambInt	**1.00**	-0.49	0.89	0.86	-0.07	0.17	0.18	0.91	-0.33	0.31	-0.11	0.34
TChambExt		**1.00**	-0.28	-0.64	0.58	0.08	0.55	-0.45	-0.05	-0.52	0.22	0.03
TAir			**1.00**	0.76	0.07	0.22	0.35	0.88	-0.43	0.21	-0.02	0.59
TSteamGen				**1.00**	-0.16	-0.06	-0.01	0.85	-0.35	0.55	-0.24	0.35
TCPU					**1.00**	0.23	0.93	-0.07	-0.16	-0.21	0.05	0.21
TCondenser						**1.00**	0.33	0.00	0.01	-0.36	0.66	0.39
TPowerBoard							**1.00**	0.20	-0.27	-0.26	0.13	0.35
PChamb								**1.00**	-0.43	0.27	-0.18	0.39
I24									**1.00**	-0.27	0.07	-0.38
PSG										**1.00**	-0.39	-0.06
PCH											**1.00**	0.23
H2O												**1.00**

Figure 4.4: Correlation matrix for Pressure Transducer runs.

The highest correlations (≥ 0.85) are between the variables:

- `TChambInt` and `TAir`, $r = 0.89$.
- `TChambInt` and `TSteamGen`, $r = 0.86$.
- `TChambInt` and `PChamb`, $r = 0.91$.
- `TCPU` and `TPowerBoard`, $r = 0.93$.
- `PChamb` and `TAir`, $r = 0.88$.

Fault: 2CS - Load: EC, 0 kg



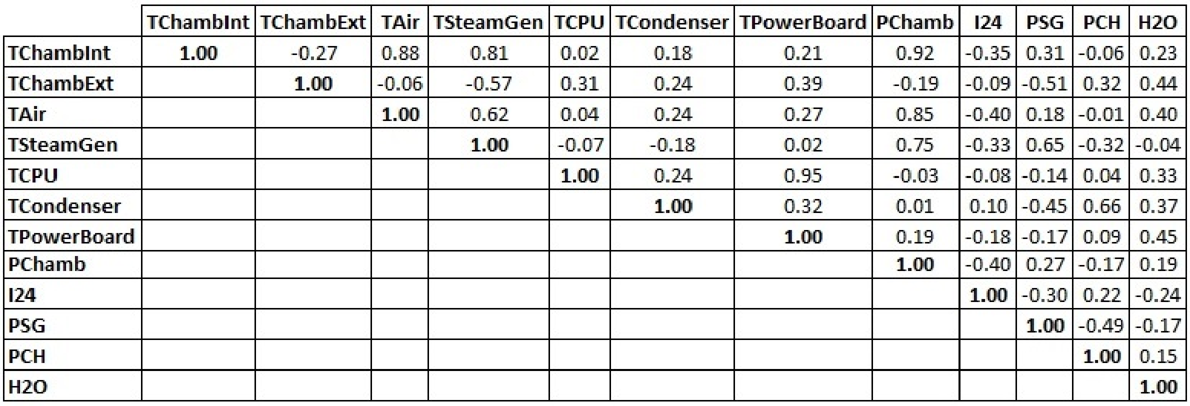
	TChambInt	TChambExt	TAir	TSteamGen	TCPU	TCondenser	TPowerBoard	PChamb	I24	PSG	PCH	H2O
TChambInt	**1.00**	0.06	0.96	0.82	-0.09	0.10	0.26	0.92	-0.41	0.24	-0.11	0.58
TChambExt		**1.00**	-0.06	-0.39	0.15	0.14	0.44	-0.13	-0.19	-0.43	0.31	0.33
TAir			**1.00**	0.89	-0.18	0.13	0.16	0.94	-0.43	0.31	-0.06	0.54
TSteamGen				**1.00**	-0.16	-0.02	-0.02	0.84	-0.34	0.55	-0.23	0.36
TCPU					**1.00**	-0.02	0.81	-0.13	0.09	-0.12	-0.01	-0.22
TCondenser						**1.00**	0.16	-0.03	0.08	-0.40	0.71	0.39
TPowerBoard							**1.00**	0.22	-0.07	-0.27	0.18	0.01
PChamb								**1.00**	-0.42	0.29	-0.22	0.40
I24									**1.00**	-0.30	0.15	-0.38
PSG										**1.00**	-0.45	-0.02
PCH											**1.00**	0.19
H2O												**1.00**

Figure 4.5: Correlation matrix for 2CS runs.

The highest correlations (≥ 0.85) are between the variables:

- `TChambInt` and `TAir`, $r = 0.96$.
- `TAir` and `TSteamGen`, $r = 0.89$.
- `TChambInt` and `PChamb`, $r = 0.92$.
- `PChamb` and `TAir`, $r = 0.94$.

Fault: Vacuum Pump - Load: EC, 0 kg



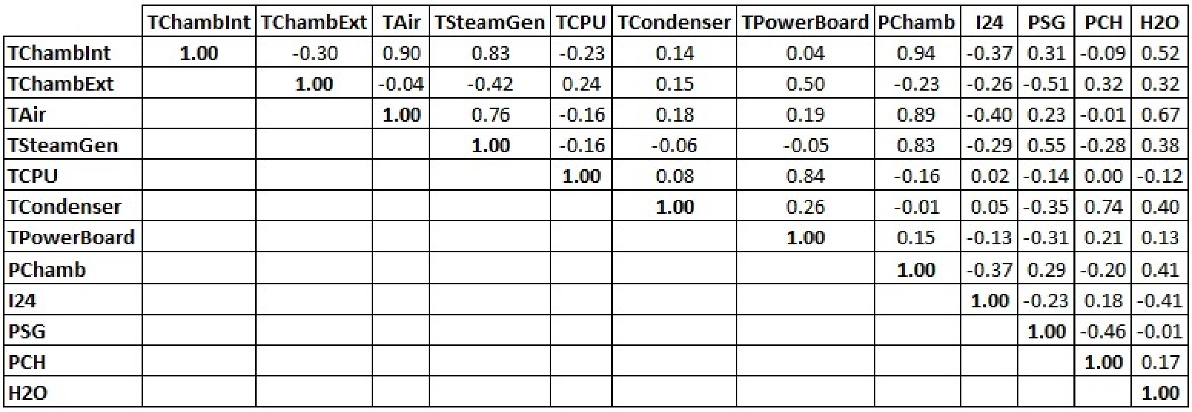
	TChambInt	TChambExt	TAir	TSteamGen	TCPU	TCondenser	TPowerBoard	PChamb	I24	PSG	PCH	H2O
TChambInt	**1.00**	-0.27	0.88	0.81	0.02	0.18	0.21	0.92	-0.35	0.31	-0.06	0.23
TChambExt		**1.00**	-0.06	-0.57	0.31	0.24	0.39	-0.19	-0.09	-0.51	0.32	0.44
TAir			**1.00**	0.62	0.04	0.24	0.27	0.85	-0.40	0.18	-0.01	0.40
TSteamGen				**1.00**	-0.07	-0.18	0.02	0.75	-0.33	0.65	-0.32	-0.04
TCPU					**1.00**	0.24	0.95	-0.03	-0.08	-0.14	0.04	0.33
TCondenser						**1.00**	0.32	0.01	0.10	-0.45	0.66	0.37
TPowerBoard							**1.00**	0.19	-0.18	-0.17	0.09	0.45
PChamb								**1.00**	-0.40	0.27	-0.17	0.19
I24									**1.00**	-0.30	0.22	-0.24
PSG										**1.00**	-0.49	-0.17
PCH											**1.00**	0.15
H2O												**1.00**

Figure 4.6: Correlation matrix for Vacuum Pump runs.

The highest correlations (≥ 0.85) are between the variables:

- `TChambInt` and `TAir`, $r = 0.88$.
- `TCPU` and `TPowerBoard`, $r = 0.95$.
- `TChambInt` and `PChamb`, $r = 0.92$.
- `PChamb` and `TAir`, $r = 0.85$.

Fault: Bacteriological Filter - Load: EC, 0 kg



	TChambInt	TChambExt	TAir	TSteamGen	TCPU	TCondenser	TPowerBoard	PChamb	I24	PSG	PCH	H2O
TChambInt	**1.00**	-0.30	0.90	0.83	-0.23	0.14	0.04	0.94	-0.37	0.31	-0.09	0.52
TChambExt		**1.00**	-0.04	-0.42	0.24	0.15	0.50	-0.23	-0.26	-0.51	0.32	0.32
TAir			**1.00**	0.76	-0.16	0.18	0.19	0.89	-0.40	0.23	-0.01	0.67
TSteamGen				**1.00**	-0.16	-0.06	-0.05	0.83	-0.29	0.55	-0.28	0.38
TCPU					**1.00**	0.08	0.84	-0.16	0.02	-0.14	0.00	-0.12
TCondenser						**1.00**	0.26	-0.01	0.05	-0.35	0.74	0.40
TPowerBoard							**1.00**	0.15	-0.13	-0.31	0.21	0.13
PChamb								**1.00**	-0.37	0.29	-0.20	0.41
I24									**1.00**	-0.23	0.18	-0.41
PSG										**1.00**	-0.46	-0.01
PCH											**1.00**	0.17
H2O												**1.00**

Figure 4.7: Correlation matrix for Bacteriological Filter runs.

The highest correlations ($r \geq 0.85$) are between the variables:

- `TChambInt` and `TAir`, $r = 0.90$.
- `TChambInt` and `PChamb`, $r = 0.94$.
- `PChamb` and `TAir`, $r = 0.89$.

Comments

For each type of fault, it is possible to note that the highest correlations concern the variables `TChambInt` and `PChamb`, `TChambInt` and `TAir`, `PChamb` and `TAir`.

Moreover, it is interesting to note that, in the altered runs, the correlation between `TChambInt` and `TSteamGen` drops below the threshold $r = 0.85$.

4.1.3 Variables Distribution

Many commonly used inferential statistics operate under the assumption that the population is normally distributed. Then, a preliminary analysis is to test the null hypothesis that the data are normally distributed.

Therefore, in this section we perform a distribution analysis of the variables that the sensors of the sterilizer measure during each run. In this case we consider runs without fault and with load 0 kg (physiological behavior). In particular, we choose the variables `Duration` and `H2O` because the first represents the velocity of sterilization and the second is a parameter that measures the consumption of water. With respect to water, the W&H patented 2CS condensate collecting system that allows to save water in each sterilization cycle.

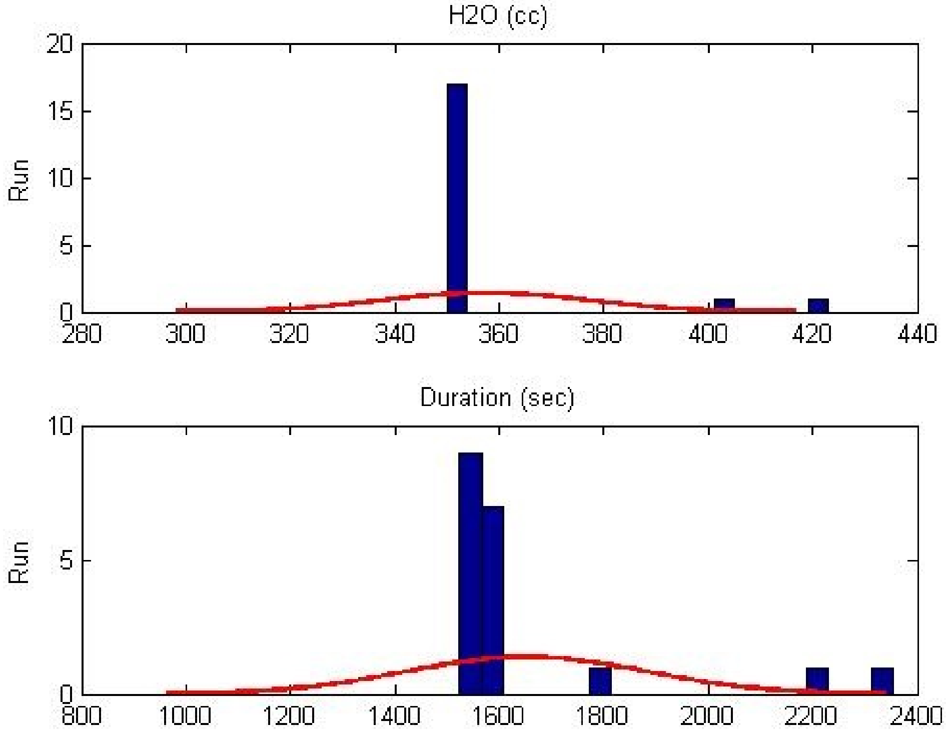


Figure 4.8: Distribution of variables `H2O` and `Duration`.

Figure 4.8 shows that the variables do not assume a Normal distribution. This graphical evaluation is confirmed by the Kolmogorov-Smirnov test ($\alpha = 5\%$). In fact, the p-values, for each variable, are lower than 0.05, therefore the null hypothesis "data comes from a standard normal distribution" is rejected.

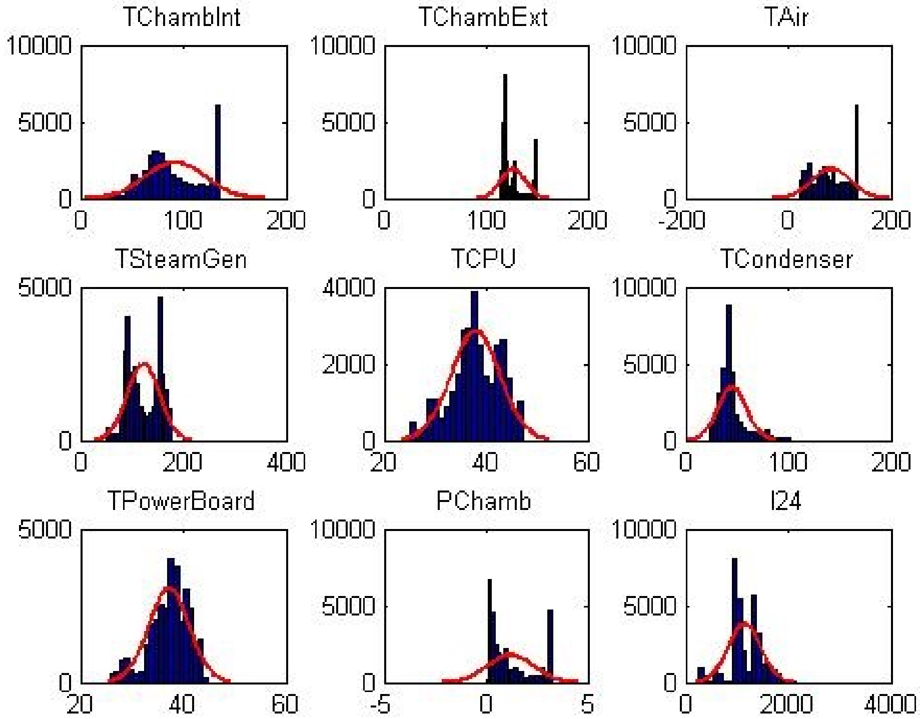


Figure 4.9: Distribution of variables measured by sensors of Steam Sterilizer

Figure 4.9 shows that the variables do not assume a Normal distribution. These initial results are useful to orientate the analysis in the next section where we will use the Kruskal-Wallis test (nonparametric ANOVA) to check the behavior differences of the sterilizer according to the type of load.

4.1.4 Load and Faults Analysis

In the next subsections we verify the sterilizer behavior in the phases pre-plateau (PV1, PP1, PV2, PP2, PV3, PPH), because the other phases have constant duration. Therefore, we consider the variable `Duration` because it is is affected by type of load. Thus, we performed this control using *Kruskal-Wallis* ($\alpha = 0.05$) test, because the distributions of each variable are not Normal. *Kruskal-Wallis* test have the following system of hypothesis:

- H_0: data comes from the same distribution.
- H_1: not all samples come from the same distribution.

In particular, we examine two kinds of load:

- EC - 0 kg: the chamber of sterilizer is empty.
- MIX - 2 kg: the chamber of sterilizer contains a 2 kg of mixed load.

To implement this analysis we used the *Empty* load type, in order to study the physiological behavior of our sterilizers. Moreover, we choose a MIX load because it is the most used by dentists. In fact, in this mode, they can be sterilized together both metal tools and porous patches. With respect to weight, 2 kg is that most used by dentists .

Variable: Duration - Fault: Door Seal (code 10) - Load: EC, 0 kg

Below we report the boxplots related to the `Duration` of each phase, with respect *Door Seal* fault. The title of each boxplot indicates the name of the studied phase, instead the x-axis reports *code 0* for good runs and *code 10* for *Door Seal* fault.

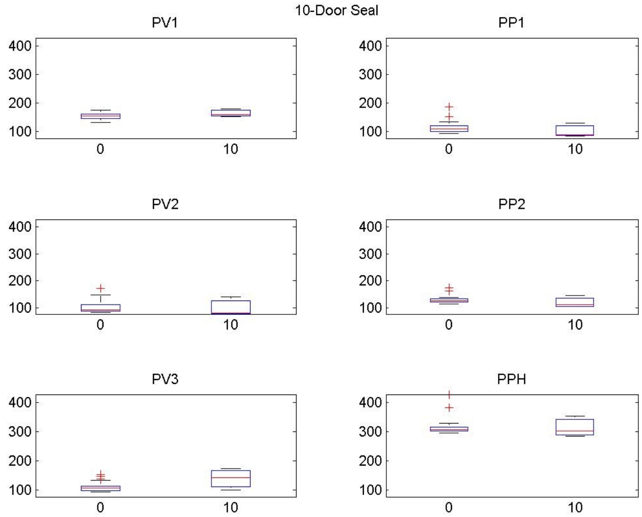
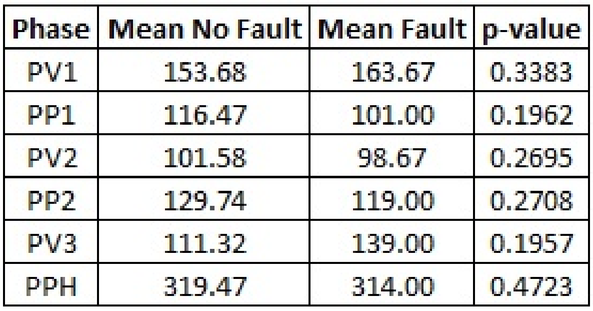


Figure 4.10: Boxplots for Duration - Fault: Door Seal - Load: EC 0 kg.

The following Table shows the durations average of each phase both for good runs and for *Door Seal* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	153.68	163.67	0.3383
PP1	116.47	101.00	0.1962
PV2	101.58	98.67	0.2695
PP2	129.74	119.00	0.2708
PV3	111.32	139.00	0.1957
PPH	319.47	314.00	0.4723

Figure 4.11: Average of Duration - Fault: Door Seal fault - Load EC 0 kg

The returned p-values indicate that Kruskal-Wallis test does not reject the null hypothesis for each phase, at a 5% significance level. However, it is possible to note that, on average, the injection phases of altered runs are longer than good runs ones.

Variable: Duration - Fault: Pressure Transd (code 21) - Load: EC, 0 kg

Below we report the boxplots related to the `Duration` of each phase, with respect *Pressure Transducer* fault. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 21* for *Pressure Transducer* fault.

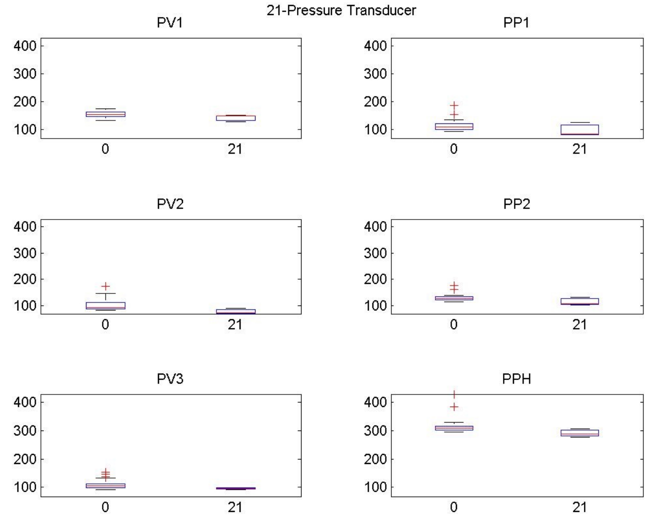
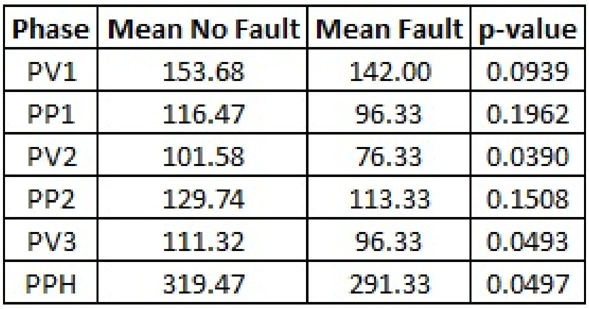


Figure 4.12: Boxplots for Duration - Fault: Pressure Transducer - Load: EC 0 kg.

The following Table shows the average durations of each phase both for good runs and for *Pressure Transducer* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	153.68	142.00	0.0939
PP1	116.47	96.33	0.1962
PV2	101.58	76.33	0.0390
PP2	129.74	113.33	0.1508
PV3	111.32	96.33	0.0493
PPH	319.47	291.33	0.0497

Figure 4.13: Average of Duration - Fault: Pressure Transducer fault - Load EC 0 kg

The returned p-values indicate that Kruskal-Wallis test rejects the null hypothesis for the phases PV2 and PV3, at a 5% significance level. Moreover, it is possible to note that, on average, the phases of altered runs are shorter than good runs ones.

Variable: Duration - Fault: 2CS (code 30) - Load: EC, 0 kg

Below we report the boxplots for the `Duration` of each phase, with respect *2CS*. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 30* for *2CS* fault.

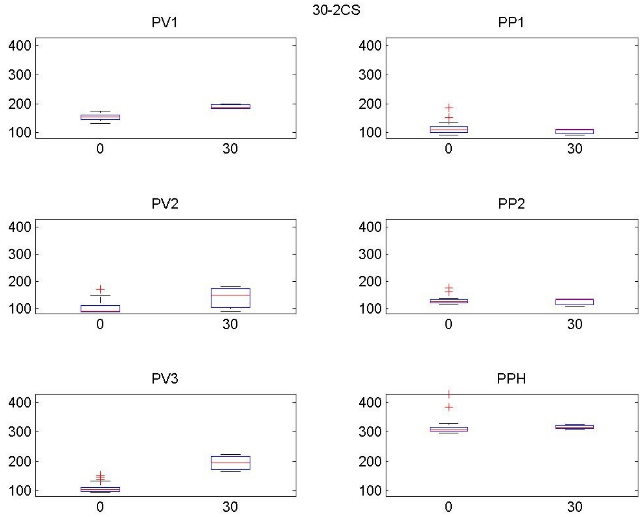
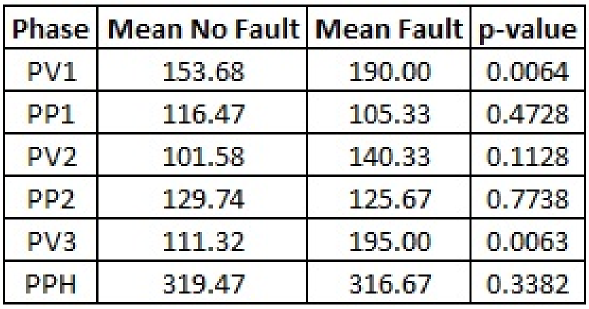


Figure 4.14: Boxplots for Duration - Fault: 2CS - Load: EC 0 kg.

The following Table shows the average durations of each phase both for good runs and for *2CS* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	153.68	190.00	0.0064
PP1	116.47	105.33	0.4728
PV2	101.58	140.33	0.1128
PP2	129.74	125.67	0.7738
PV3	111.32	195.00	0.0063
PPH	319.47	316.67	0.3382

Figure 4.15: Average of Duration - Fault: 2CS - Load EC 0 kg

The returned p-values indicate that Kruskal-Wallis test rejects the null hypothesis for the phases PV1 and PV3, at a 5% significance level. Moreover, it is possible to note that, on average, the vacuum phases of altered runs are longer than good runs ones.

Variable: Duration - Fault: Vacuum Pump (code 40) - Load: EC, 0 kg

Below we report the boxplots for the `Duration` of each phase, with respect *Vacuum Pump*. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 40* for *Vacuum Pump* fault.

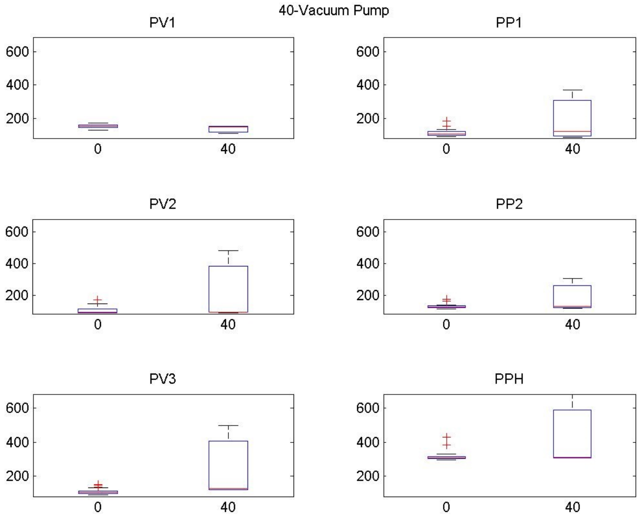
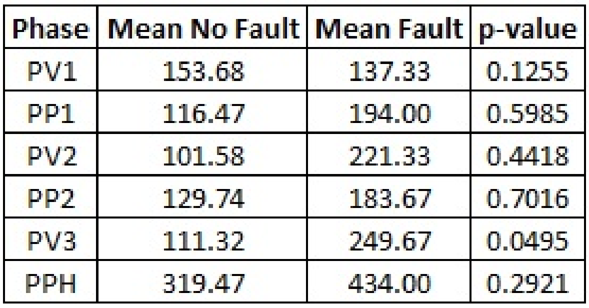


Figure 4.16: Boxplots for Duration - Fault: Vacuum Pump - Load: EC 0 kg.

The following Table shows the average durations of each phase both for good runs and for *Vacuum Pump* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	153.68	137.33	0.1255
PP1	116.47	194.00	0.5985
PV2	101.58	221.33	0.4418
PP2	129.74	183.67	0.7016
PV3	111.32	249.67	0.0495
PPH	319.47	434.00	0.2921

Figure 4.17: Average of Duration - Fault: Vacuum Pump - Load EC 0 kg

The returned p-values indicate that Kruskal-Wallis test rejects the null hypothesis for the phase PV3, at a 5% significance level. Moreover, it is possible to note that, on average, the phases of altered runs, except PV1, are longer than good runs.

Variable: Duration - Fault: Bacteriol. Filter (code 50) - Load: EC, 0 kg

Below we report the boxplots for the `Duration` of each phase, with respect *Bacteriological Filter*. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 50* for *Bacteriological Filter* fault.

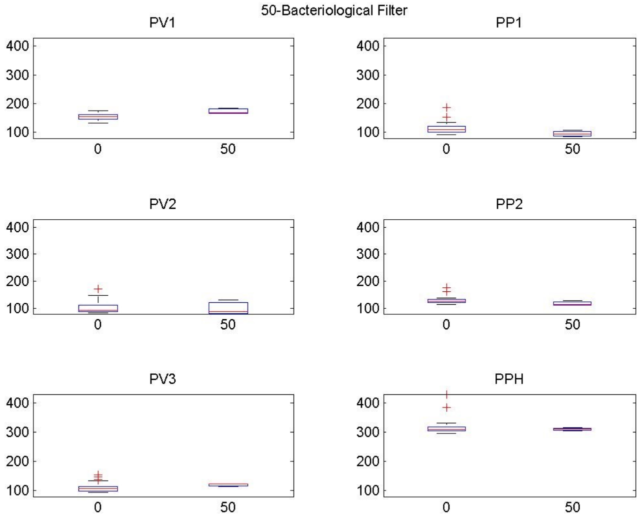
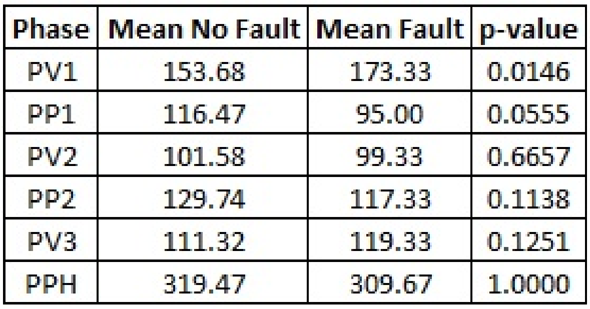


Figure 4.18: Boxplots for Duration - Fault: Bacteriological Filter - Load: EC 0 kg.

The following Table shows the average durations of each phase both for good runs and for *Bacteriological Filter* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	153.68	173.33	0.0146
PP1	116.47	95.00	0.0555
PV2	101.58	99.33	0.6657
PP2	129.74	117.33	0.1138
PV3	111.32	119.33	0.1251
PPH	319.47	309.67	1.0000

Figure 4.19: Average of Duration - Fault: Bacteriological Filter - Load EC 0 kg

The returned p-values indicate that Kruskal-Wallis test rejects the null hypothesis for the phase PV1, at a 5% significance level.

Variable: Duration - Fault: Door Seal (code 10) - Load: MIX, 2 kg

Below we report the boxplots for the `Duration` of each phase, with respect *Door Seal*. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 10* for *Door Seal* fault.

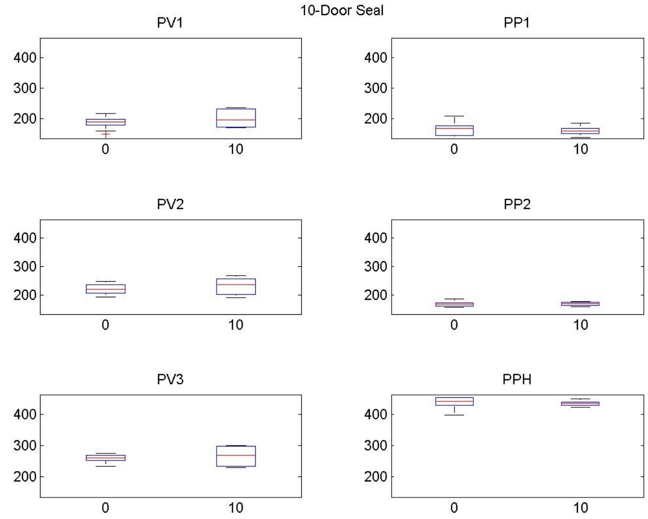
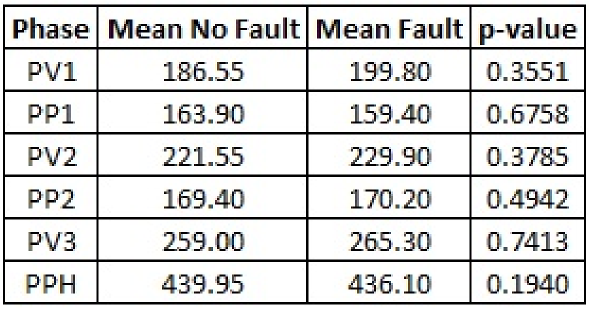


Figure 4.20: Boxplots for Duration - Fault: Door Seal - Load: EC 0 kg.

The following Table shows the average durations of each phase both for good runs and for *Door Seal* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	186.55	199.80	0.3551
PP1	163.90	159.40	0.6758
PV2	221.55	229.90	0.3785
PP2	169.40	170.20	0.4942
PV3	259.00	265.30	0.7413
PPH	439.95	436.10	0.1940

Figure 4.21: Average of Duration for Door Seal- Load MIX 2 kg

The returned p-values indicate that Kruskal-Wallis test does not reject the null hypothesis for each phase, at a 5% significance level.

Variable: Duration - Fault: Press Transd (code 21) - Load: MIX, 2 kg

Below we report the boxplots for the `Duration` of each phase, with respect *Pressure Transducer*. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 21* for *Pressure Transducer* fault.

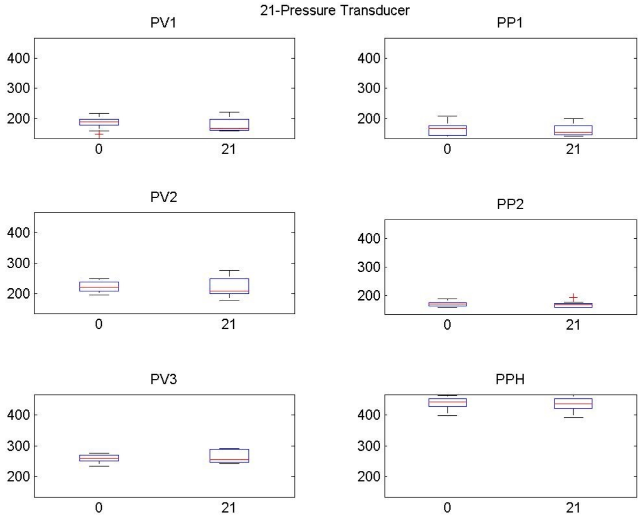
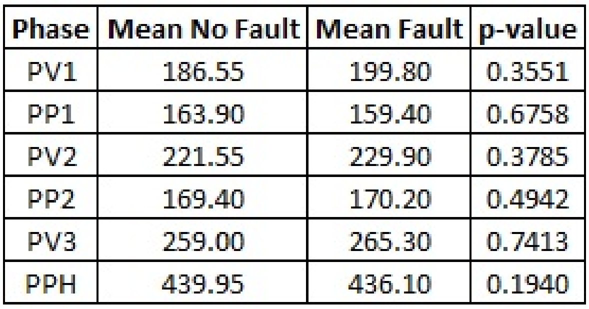


Figure 4.22: Boxplots for Duration - Fault: Pressure Transducer - Load: MIX 2 kg.

The following Table shows the average durations of each phase both for good runs and for *Pressure Transducer (code 21)* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	186.55	199.80	0.3551
PP1	163.90	159.40	0.6758
PV2	221.55	229.90	0.3785
PP2	169.40	170.20	0.4942
PV3	259.00	265.30	0.7413
PPH	439.95	436.10	0.1940

Figure 4.23: Average of Duration - Fault: Pressure Transducer- Load MIX 2 kg

The returned p-values indicate that Kruskal-Wallis test does not reject the null hypothesis for each phase, at a 5% significance level. Moreover, it is possible to note that, on average, the phases of altered runs, except PP1, are longer than good runs.

Variable: Duration - Fault: 2CS (code 30) - Load: MIX, 2 kg

Below we report the boxplots for the `Duration` of each phase, with respect *2CS*. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 30* for *2CS* fault.

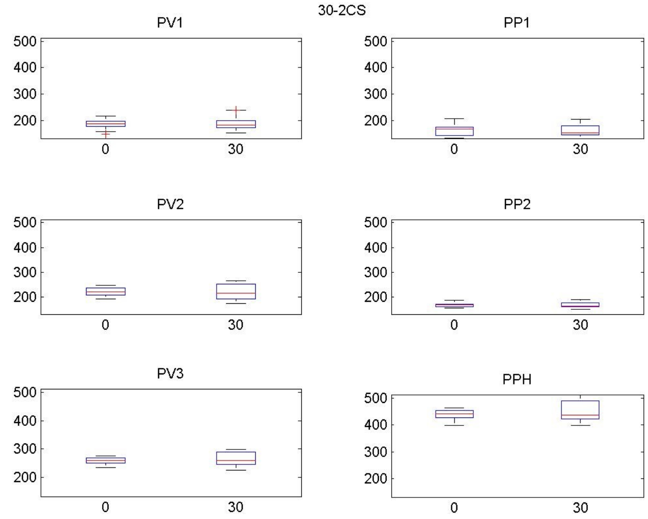
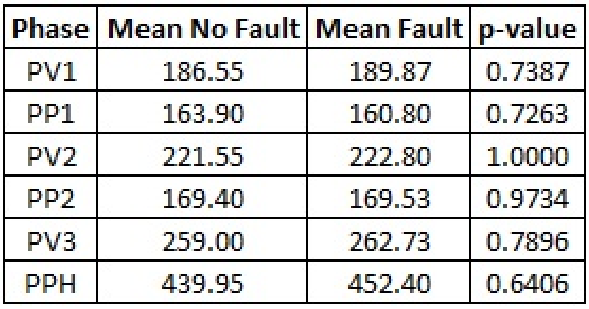


Figure 4.24: Boxplots for Duration - Fault: 2CS - Load: EC 0 kg.

The following Table shows the average durations of each phase both for good runs and for *2CS* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	186.55	189.87	0.7387
PP1	163.90	160.80	0.7263
PV2	221.55	222.80	1.0000
PP2	169.40	169.53	0.9734
PV3	259.00	262.73	0.7896
PPH	439.95	452.40	0.6406

Figure 4.25: Average of Duration - Fault: 2CS- Load MIX 2 kg

The returned p-values indicate that Kruskal-Wallis test does not reject the null hypothesis for each phase, at a 5% significance level. Moreover, it is possible to note that, on average, the phases of altered runs, except PP1, are longer than good runs.

Variable: Duration - Fault: Vacuum Pump (code 40) - Load: MIX, 2 kg

Below we report the boxplots for the `Duration` of each phase, with respect *Vacuum Pump*. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 40* for *Vacuum Pump* fault.

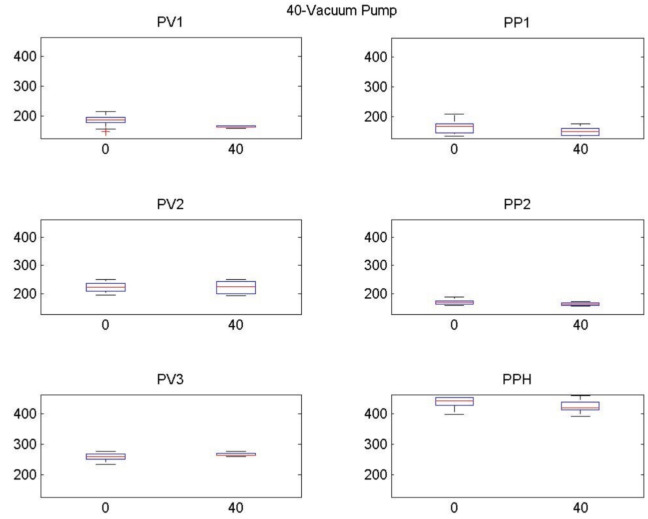
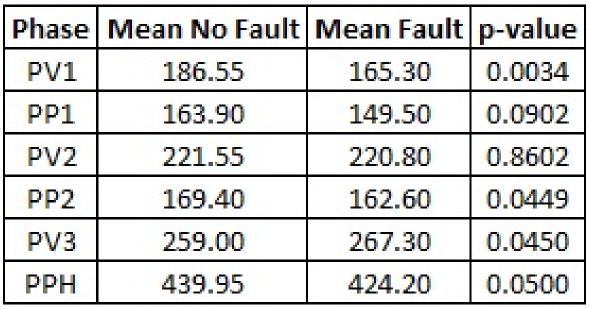


Figure 4.26: Boxplots for Duration - Fault: Vacuum Pump - Load: EC 0 kg.

The following Table shows the average durations of each phase both for good runs and for *Vacuum Pump* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	186.55	165.30	0.0034
PP1	163.90	149.50	0.0902
PV2	221.55	220.80	0.8602
PP2	169.40	162.60	0.0449
PV3	259.00	267.30	0.0450
PPH	439.95	424.20	0.0500

Figure 4.27: Average of Duration - Fault: Vacuum Pump - Load MIX 2 kg

The returned p-values indicate that Kruskal-Wallis test rejects the null hypothesis for the phases PV1, PP2 and PV3, at a 5% significance level. Moreover, it is possible to note that, on average, the phases of altered runs, except PV3, are shorter than good runs.

Variable: Duration - Fault: Bacter. Filter (code 50) - Load: MIX, 2 kg

Below we report the boxplots for the `Duration` of each phase, with respect *Bacteriological Filter*. The title of each boxplot indicates the name of the phase, instead the x-axis reports *code 0* for good runs and *code 50* for *Bacteriological Filter* fault.

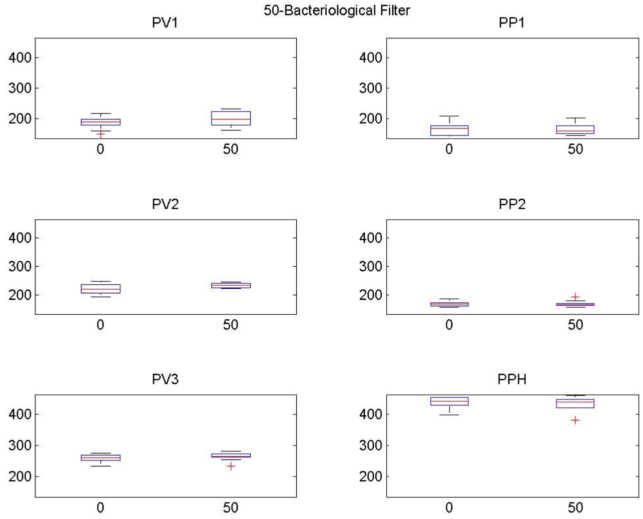
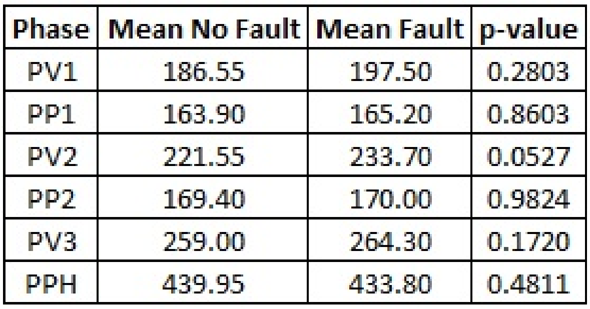


Figure 4.28: Boxplots for Duration - Fault: Bacteriological Filter - Load: EC 0 kg.

The following Table shows the average durations of each phase both for good runs and fo *Bacteriological Filter* runs, and the p-value of the Kruskal-Wallis test performed between good and altered runs.



Phase	Mean No Fault	Mean Fault	p-value
PV1	186.55	197.50	0.2803
PP1	163.90	165.20	0.8603
PV2	221.55	233.70	0.0527
PP2	169.40	170.00	0.9824
PV3	259.00	264.30	0.1720
PPH	439.95	433.80	0.4811

Figure 4.29: Average of Duration - Fault: Bacteriological Filter- Load MIX 2 kg

The returned p-values indicate that Kruskal-Wallis test does not reject the null hypothesis for each phase, at a 5% significance level. Moreover, it is possible to note that, on average, the phases of altered runs, except PPH, are longer than good runs.

Therefore, since no changes related to level, variability and linear dynamics are observed, if an effect is to be observed, this is related to curvature and functional data analysis could help. We develop this in connection to control chart, since this approach has a practical utility in maintenance management.

4.2 Control Chart Results

Nov we consider both traditional multivariate control charts for level changes and functional control charts for curvature changes, applied to the seven variables of Table 2.3, in particular: `TChambInt`, `TChambExt`, `TSteamGen`, `TCondenser`, `PChamb`, `PowerSteamGen` and `I24`. To do this, we used the 82 failure free runs in order to estimate the *in-control* parameters for both the MEWMA on levels of Section 3.2.5 and the f-MEWMA of Section 3.3.

In particular data and random effects have been standardized using *in-control* values to assure zero mean before machine manumission and the exponential smoothing factor has been kept constant in all cases, with $\lambda = 0.05$, to have a good power against small shifts. In order to show the results of tampering with the door seal component, we attach the 29 altered runs, after the above 82 failure free runs, simulating a failure intervened at run $i_0 = 83$. First we focus on the level surveillance with data vector $\boldsymbol{y}_i$ given by the 7-dimensional run mean, for each run i.

4.2.1 MEWMA

First of all we report the results of traditional MEWMA control chart. In particular the Figure 4.30 shows the matrices flow related to MEWMA for Lisa:

- **L-block**: this block we collect the data and create the dataset *Lisa_ TOT*. In particular this dataset has n rows (sum of the observations of all runs) and v columns (number of variables).

- **M-block**: in this block we process the dataset coming from *L-block* and returns the *MEWMA* control charts. In particular, the output is a vector with dimension a x b, where a is the number of faults and b is the number of statistics T^2 that identify each run.

Figure 4.31 depicts the MEWMA detector T_i^2 against run index i, with vertical line which identifies the first altered run. In the same Figure the horizontal line represents the threshold h. As expected it is apparent that the MEWMA control chart for levels is not able to detect the anomaly induced on door seal.

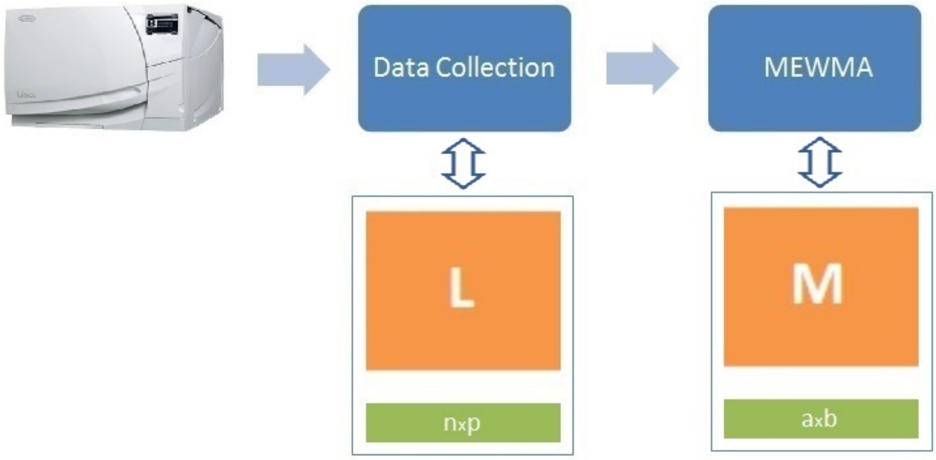


Figure 4.30: MEWMA algorithm without functional data

Finally the MEWMA model (with $\lambda = 0.05$), is defined as follows:

$$\boldsymbol{Q}_i = 0.05\boldsymbol{y}_i + (1 - 0.05)\boldsymbol{Q}_{i-1}$$

where $\boldsymbol{y}_i$ is a vector that contains the output values of each phase and i represents the i-th run. The quantity plotted on the control chart is

$$T_i^2 = \boldsymbol{Q_i'}\boldsymbol{\Sigma_Q^{-1}}\boldsymbol{Q_i}$$

where the covariance matrix is

$$\boldsymbol{\Sigma_Q} = \frac{0.05}{2 - 0.05}\boldsymbol{\Sigma}$$

and Σ is an identity matrix.

Figure 4.31 shows that the EWMA control chart, with respect to the $v = 7$ above cited variables, is not able to detect the anomaly related to the Door Seal anomaly.

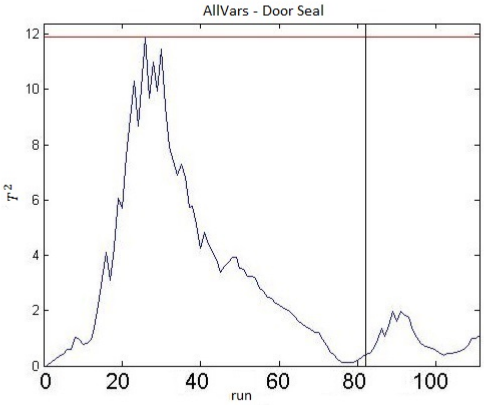


Figure 4.31: MEWMA on levels for Door Seal anomaly; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

Figure 4.32, related to the variable `TChambInt`, and Figure 4.33, related to the variable `PChamb`, show that the EWMA control chart is not able to detect the anomalies with respect to all components/faults described in Section 2.4.

In next Section we consider the curvature surveillance. To do this, we used a cubic spline for each of the $v = 7$ sterilizer variables and each run i, with $K = 30$ evenly spaced knots.

4.2.2 f-MEWMA

Using the estimated random effects $\hat{u}_i$, we computed the overall 7-dimensional f-MEWMA detector. Moreover we computed seven univariate f-EWMA detectors, one for each variable. Figure 4.36 depicts the multivariate detector on the top panel, and two representatives out of the seven univariate detectors on the bottom panels. It is apparent that the overall detector signals the door seal anomaly quite fast. From the bottom panels, it is also clear that the curvature of the chamber internal temperature profile is much affected by the considered manumission. On the other side the door seal defect does not affect the curvature of the condenser temperature profile.

In order to assess model adequacy, Figure 4.34 shows the MSE for the 111 runs of `TChambInt`. Recalling that data have be standardized to unit variance, it can

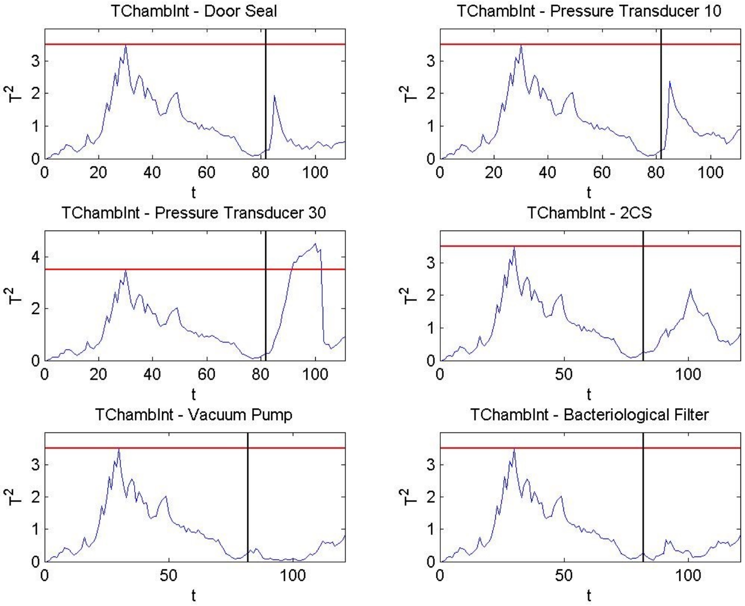


Figure 4.32: EWMA for TChambInt with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

be observed that these MSE's are quite small and insensitive to door seal anomaly, confirming that the penalized cubic splines with $K = 30$ knots do a good job for these data. Similar results were obtained for the other six variables.

Below, we report the results of the control chart *f-MEWMA* that use functional data. In particular the Figure 4.35 shows the matrices flow that is divided in three blocks:

- **L-block**: in this block we collect the data and create the dataset *Lisa_ TOT*. In particular we have n rows (sum of the observations of all runs) and v columns (number of variables). In our case, the final dimension of *L-matrix* is 684277 x 7.

- **B-block**: this block allows the spline transformation of the dataset coming from *L-block* and generates called *FD_ Lisa*. In particular this dataset has r rows (number of runs) and k columns (number of coefficients). In our case, $r = g + f$, where $g = 82$ represents the *good runs* and f is the number of altered runs that changes according to type of fault (see DOE in Figure 4.1).

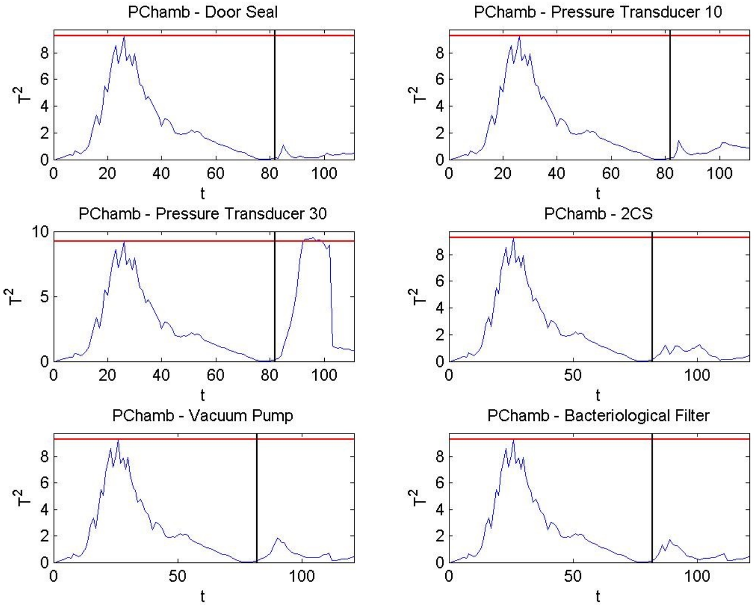


Figure 4.33: EWMA for PChamb with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

- **M-block**: this block processes the dataset coming from *B-block* and returns the *f-MEWMA* control charts. In particular, the output is a matrix with dimension $(a = 6)$ x $b = r$, where a represents the number of faults and b is the number of statistics T^2 that identify each run (i.e., the number of run).

Figure 4.36 shows the plots related to *f-MEWMA* control chart that considers the above 7 variables with respect to type of faults described in Section 2. It s possible to note that our *f-MEWMA* control chart is very effective to detect the anomaly related to Door Seal. Moreover, we also show the *f-EWMA* for each variable and fault in order to see their anomalies detecting performance.

Figure 4.37, related to the variable `TChambInt`, and Figure 4.38, related to the variable `PChamb`, show that, for each type of fault, *f-EWMA* control chart detects the anomalies with respect to components described in Section 2.4.

Figure 4.40 shows that, for each type of fault, *f-EWMA* control chart, related to the variable `I24`, is not able to detects the anomalies related to components described in Section 2.4. Figure 4.41 shows that, for each type of fault, *f-EWMA*

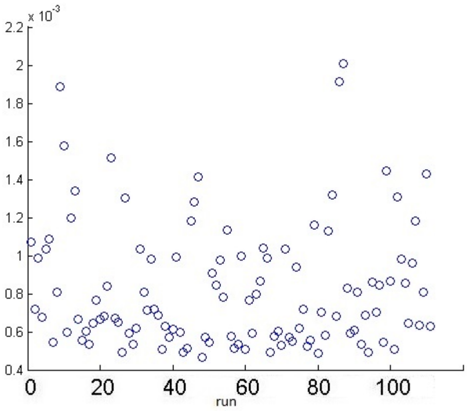


Figure 4.34: Mean square error to evaluate the spline model goodness of fit for `TChambInt` variable; abscissa is the run index i, ordinate is MSE.

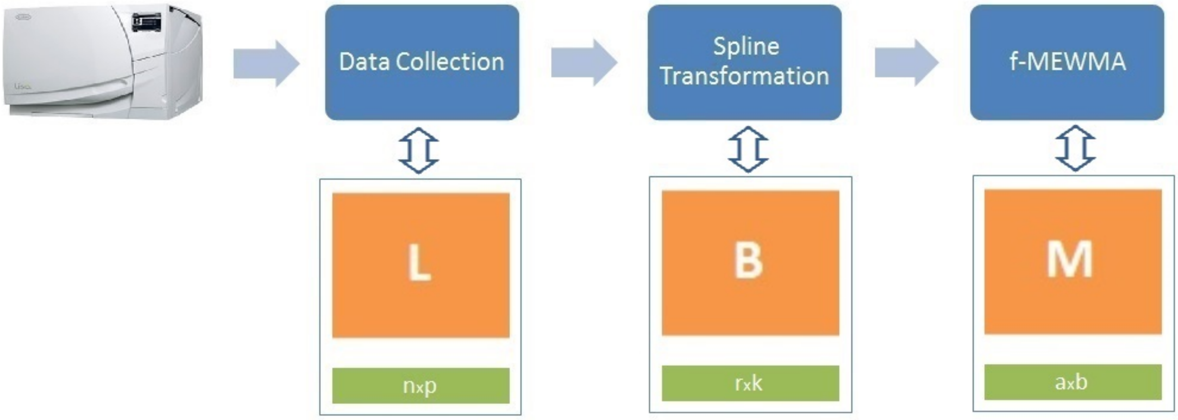


Figure 4.35: f-MEWMA algorithm. The three block represent the data collection, the data transformation in spline and the monitoring of system health, respectively.

control chart, related to the variable `TSteamGen`, is not able to detects the anomalies related to components described in Section 2.4. Figure 4.42 shows that, for each type of fault, *f-EWMA* control chart, related to the variable `TCondenser`, is not able to detects the anomalies related to components described in Section 2.4. Figure 4.43 shows that, for each type of fault, *f-EWMA* control chart is not able to detects the anomalies related to components described in Section 2.4. Figure 4.39 shows that, for each type of fault, *f-EWMA* control chart detects, related to the variable `PowerSteamGen`, the anomalies with respect to components described in Section 2.4.

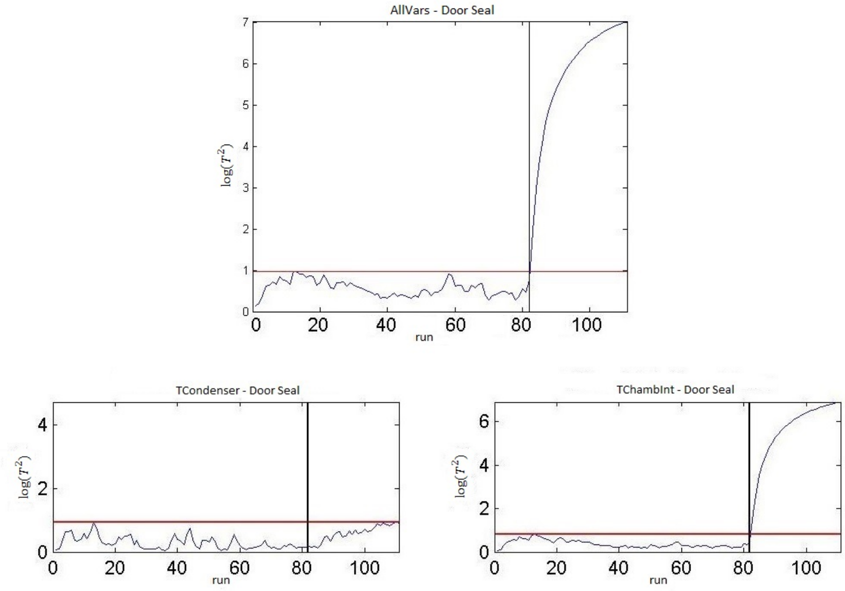


Figure 4.36: f-MEWMA for Door Seal anomaly; top panel is the 7-variate f-MEWMA; bottom left panel is the `TCondeser` f-EWMA; bottom right panel is the `TChambInt` f-EWMA; in all panels, abscissa is experimental run, ordinate is detector T_i^2, vertical line identifies the first altered run and horizontal line represents the threshold h.

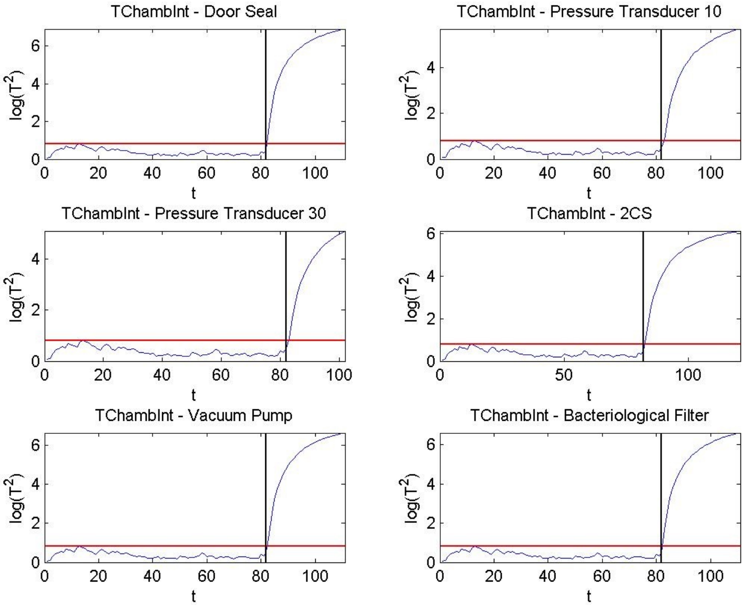


Figure 4.37: f-EWMA for TChambInt with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

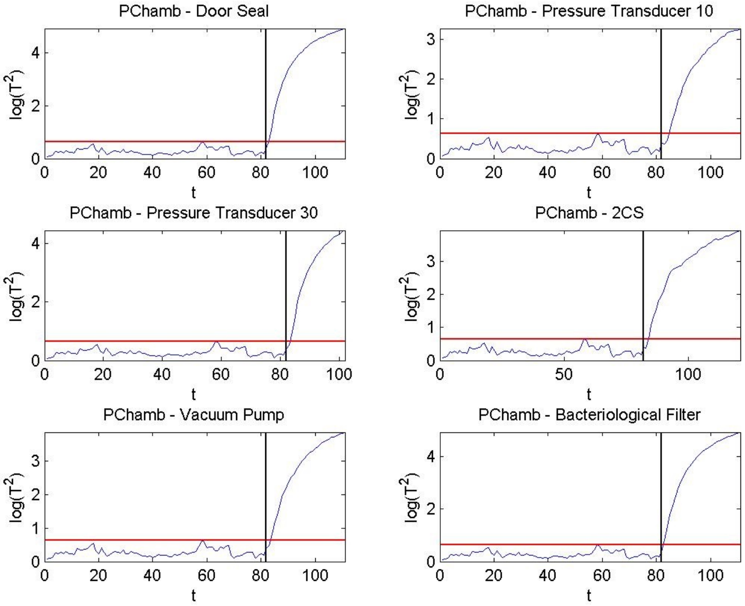


Figure 4.38: f-EWMA for PChamb with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

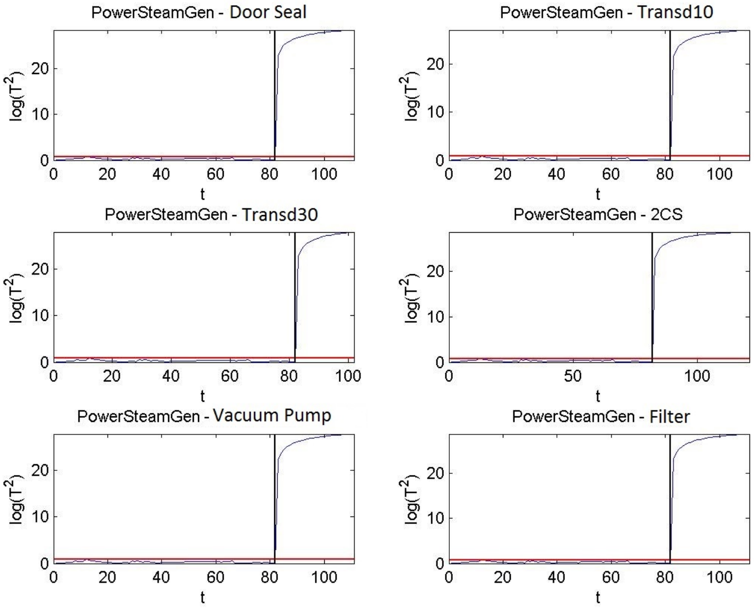


Figure 4.39: f-EWMA for PowerSteamGen with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

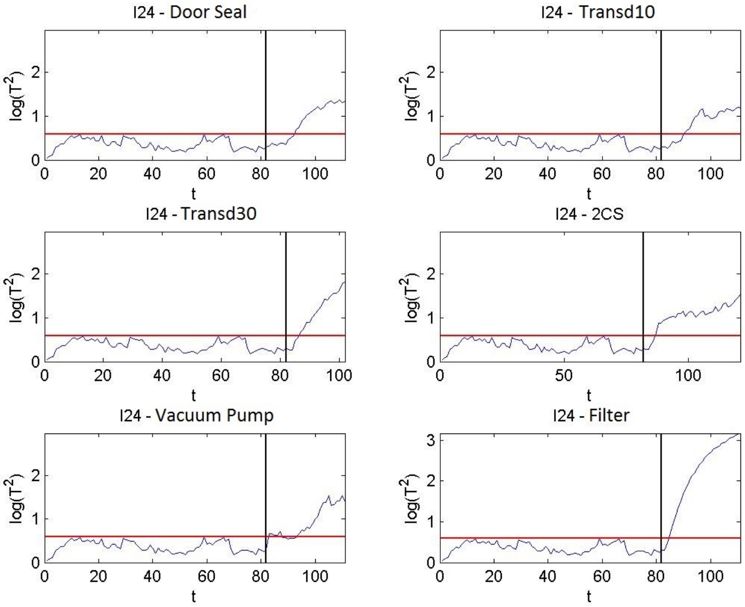


Figure 4.40: f-EWMA for I24 with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

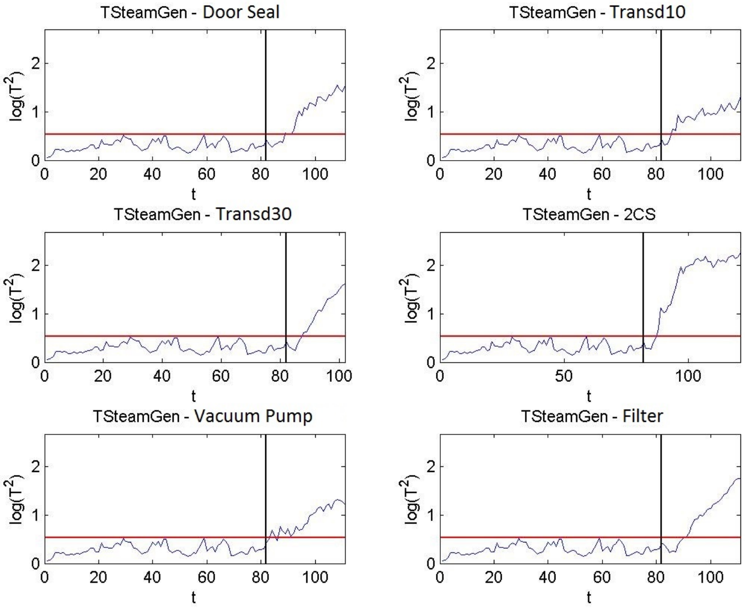


Figure 4.41: f-EWMA for TSteamGen with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

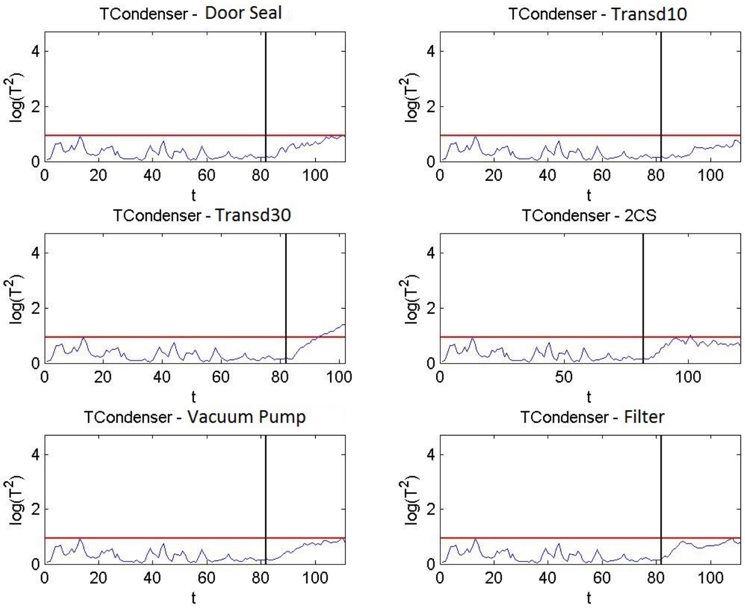


Figure 4.42: f-EWMA for TCondenser with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

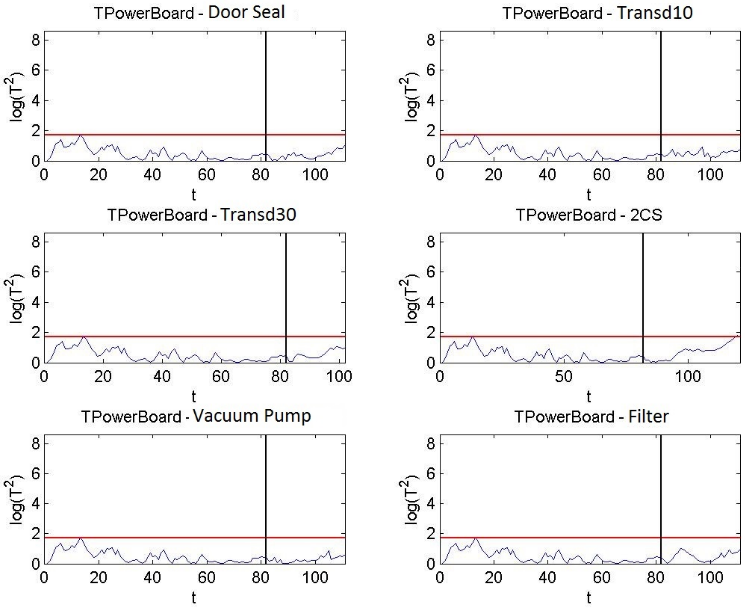
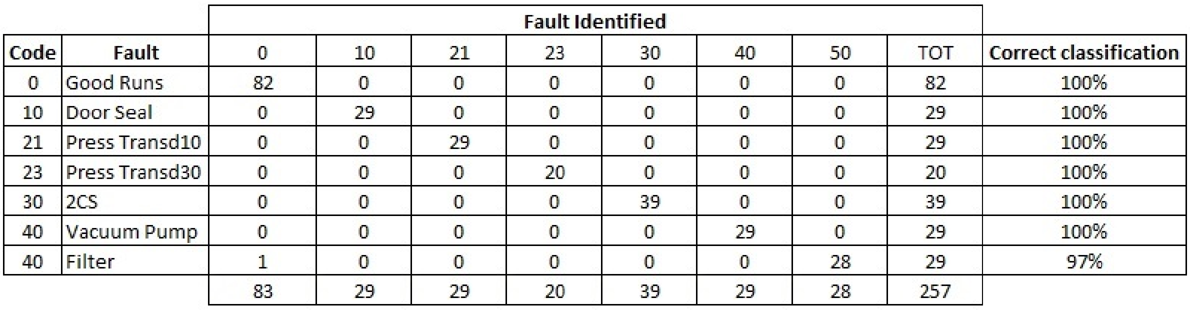


Figure 4.43: f-EWMA for TPowerBoard with respect to all faults; abscissa is experimental run, ordinate is detector T_i^2; vertical line identifies the first altered run; horizontal line represents the threshold h.

4.3 Runs Classifier

Now, we report the results of runs classification, with respect to faults, using probabilistic classification technique.



		Fault Identified								
Code	Fault	0	10	21	23	30	40	50	TOT	Correct classification
0	Good Runs	82	0	0	0	0	0	0	82	100%
10	Door Seal	0	29	0	0	0	0	0	29	100%
21	Press Transd10	0	0	29	0	0	0	0	29	100%
23	Press Transd30	0	0	0	20	0	0	0	20	100%
30	2CS	0	0	0	0	39	0	0	39	100%
40	Vacuum Pump	0	0	0	0	0	29	0	29	100%
40	Filter	1	0	0	0	0	0	28	29	97%
		83	29	29	20	39	29	28	257	

Figure 4.44: Runs classification using probabilistic classification technique

The percentage of correct identification is nearly 100% when using all variables. However, using only the variables `TChambInt` and `PChamb`, this percentage does not drop below 95%.

4.4 Data Flow and Software

In this Section we show the functions that we have developed by MATLAB/OCTAVE environment. In particular, we have three type of them:

- Data Collection Functions: these functions, for each run, allow to import the log files generate by Lisa and convert them in MATLAB/OCTAVE dataset called *Lisa_TOT*.
- f-MEWMA Building Functions: these functions allow to convert each sterilization cycle in a multivariate functional object, and, then create the *f-MEWMA* control chart.
- Classifier Functions: these functions allow to classify each sterilization run with respect to the faults using a bayesian approach.

4.4.1 Data Collection Functions

At the end of each run the sterilizer generates a *scl* file that contains the operating parameters. The *scl* files are saved by sterilizer into an SD card. Figure 4.45 shows the data collection process that allows to create a dataset.

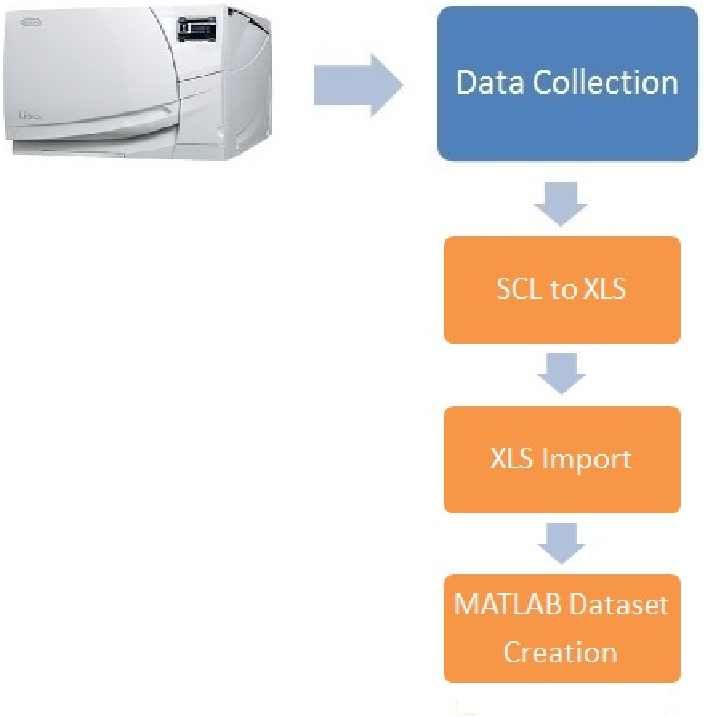


Figure 4.45: Data collection flow

As showed by the Figure 4.45, the process of data collection is composed by three steps:

- *SCL to XLS*: Each *scl* file is converted in a *xls* file by a devoted *VBA* function.
- *XLS Import*: Each *xls* file is imported in MATLAB/OCTAVE by a devoted function.
- *Dataset Creation*: Thanks to a devoted functions is created a dataset for each *xls* and then all datasets are stacked in order to create the final dataset *Lisa_TOT*.

Therefore, *Lisa_TOT* is a dataset that contains the raw data that we process using spline method. In particular, it is possible to divide the information that contains *Lisa_ TOT* in three areas:

1. First Area: in this area there are information that identifies each run (i.e., ID Run, type of load, kgs, etc.).
2. Second Area: in this area there are the operating parameters (variables) that the sterilizer has measured during each run (see Table 2.3).
3. Third Area: in this area there are the dummy variables that identify electrovalves, phases and steps are active in each run.

In particular, Figure 4.46 shows in detail the functions that we have created to implement the process of data collection.

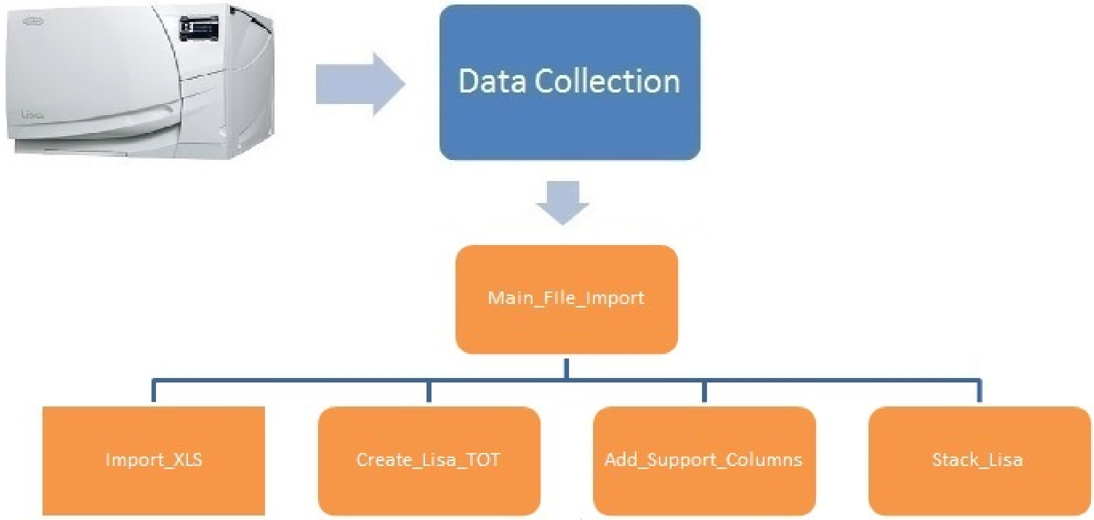


Figure 4.46: Devoted functions for data collection process

Below we explain the devoted functions that import the *xls* files and create the dataset *Lisa_ TOT*:

Main_File_Import

This is the main script that governs the whole data collection process and recalls the devoted functions.

Import_XLS

This function imports the *xls* files (i.e. each run).

Create_Lisa_TOT

This function creates a dataset for each run.

Add_Support_Columns

This function adds information columns for each dataset of each run.

Stack_Lisa

This function stacks the individual dataset of each run and creates the dataset *Lisa_TOT*.

4.4.2 f-MEWMA Building Functions

In this section we describe the functions flow, as depicted in Figure 4.47, that allow to generate the control charts *f-MEWMA*. We already explained the Data Collection step in Section4.4.1.

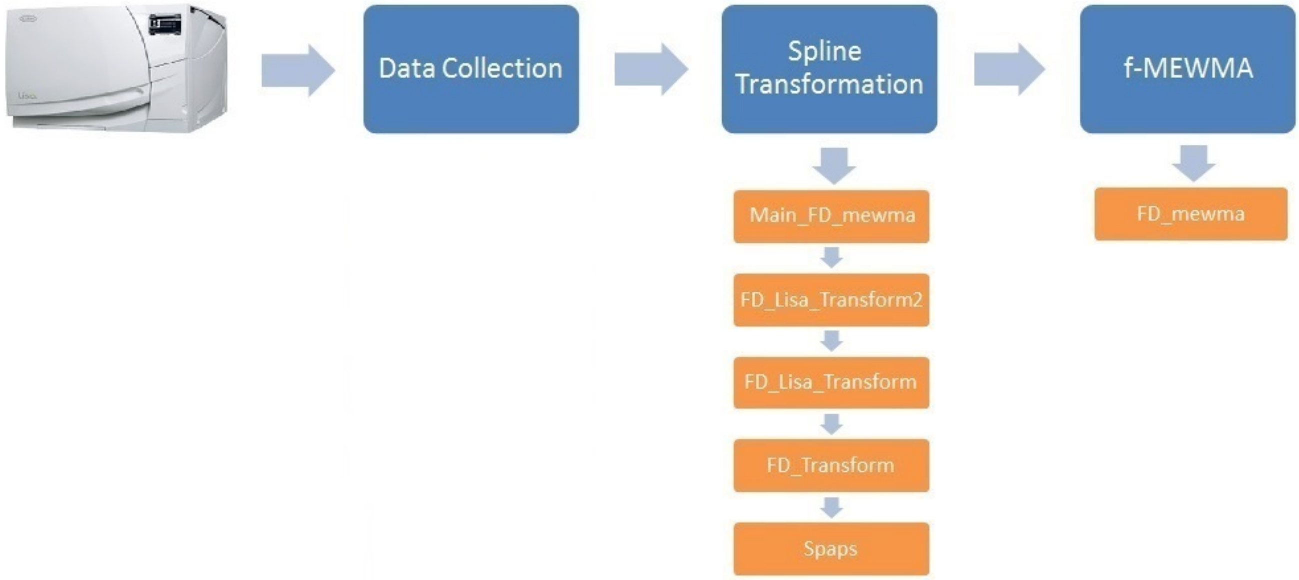


Figure 4.47: Functions flow.

Below we explain the functions that transform the raw data in *spline*. In particular, the step *spline Transformation* includes the following functions:

Main_FD_mewma

This is the main script that governs the whole *f-MEWMA* generation process. In particular, this script initializes the *structure array* `FD_setup` that is a data type that groups related data using data containers called *fields*. The fields of `FD_setup` are the following:

- `n_phases`: number of macro-phases in which is divided the sterilization cycle; specifically, we identify 3 macro-phases (Start, Plateau, Dry). In particular,

  each run has a different duration which are included between 30 and 45 minutes, according to type of load and, moreover, we divided it in three macro-phases. Therefore, in order to have runs with homogeneous lengths, sterilization time, called x, is standardized to have unit length for each macro-phase as follows:

  1. Preparation phase, the machine performs vacuum and steam injection phases ($0 < x < 1$);
  2. Sterilization phase, the machine performs the high temperature sterilization ($1 < x < 2$);
  3. Dry phase, the machine performs load drying ($2 < x < 3$).

  Note that each macro-phase can be divided in a number of micro-phases that we described in Tab 2.1.

- `n_knots`: number of knots (10) for each macro-phase.
- `y_tol`: tolerance; it helps to connect knots and works together knots weights.

Finally, this script invokes, for each selected variable of `Lisa_TOT`, the functions `FD_Lisa_Transform2` and `FD_mewma`.

FD_Lisa_Transform2

This function selects a fault at a time and constructs the dataset `FD_Lisa`.

The input parameters are:

- `Lisa_TOT`: described in section 2.3.
- `variabili`: represents the selected variables.
- `faults`: represents the code of each fault (10; 21; 23; 30; 40; 50).
- `FD_setup`: described in subsection *Main_FD_mewma*.

The output parameters are:

- `FD_Lisa`: this is the dataset the contains, for each run, the following information: identifier of run, type of fault, type of load and weight of load (kg). Moreover, it contains, for each variable, the spline coefficients.
- `col_vn1`: this is an integer variable that indicate the number of column of `FD_Lisa` where start the coefficients of splines.

FD_Lisa_Transform

This function calculates the number of run for each fault, controls and considers only the variables with variance different from zero, and prepares the splines vector of coefficients for each run.

The input parameters are:

- `Lisa_TOT_fault`: this is the dataset *Lisa_TOT* that contain only the runs of a particular fault.
- `variables`: this is a cell that contains the selected variables (e.g., `TChambInt` and `PChamb`).
- `FD_setup`: described in subsection *Main_FD_mewma*.

The output parameters are:

- `Y_COEFS`: this is the dataset that contains the coefficients of splines for each run and variable.

- `Bhead`: this is a dataset the contains, for each run, some information as: identifier of run, type of fault, type of load and weight (kg).

FD_Transform

This function estimates the spline coefficients. In particular, to make this it invokes a devote function.

The input parameters are:

- `Lisa_TOT_fault`: described in subsection *FD_Lisa_Transform*.
- `col_vn1`: described in subsection *FD_Lisa_Transform2*.
- `runs`: this vector contains the runs identifiers.
- `FD_setup`: described in subsection *Main_FD_mewma*.
- `faults`: this vector contains the faults identifiers.

The output parameters are:

- `y_pp`: returns the spline coefficients.
- `Bhead`: described in subsection *FD_Lisa_Transform*.

SPAPS

To estimate spline coefficients we used a devoted function of MATLAB *spaps*. This function returns the B-form of the smoothest function f that lies within the given tolerance of the given data points. Below we report the syntax:

$$[sp] = spaps(x, y, tol) \tag{4.1}$$

where the input parameters are:

- x: Knots.
- y: Observed data.
- tol: Tolerance. When tol is nonnegative, then the spline f is determined as the unique minimizer of the expression $\rho E(f) + F(D^m f)$, with the smoothing parameter ρ so chosen that $E(f)$ equals tol.

instead the output parameter is:

- *sp*: returns the *B-form* of the smoothest function f that lies within the given tolerance *tol* of the given data points $(x(j), y(:, j))$, where $j = 1 : length(x)$. The data values $y(:, j)$ may be scalars, vectors, matrices, even ND-arrays. Data points with the same data site are replaced by their weighted average, with its weight the sum of the corresponding weights, and the tolerance *tol* is reduced accordingly.

The distance of the function f from the given data is measured by

$$E(f) = \sum_{i=1}^{n} w(j)|y(:, j) - f(x(j))|^2 \tag{4.2}$$

with the default choice for the weights w making $E(f)$ the composite trapezoidal rule approximation. Further, *smoothest* means that the following roughness measure is minimized:

$$E(f) = F(D^m f) = \int_{min(x)}^{max(x)} \lambda(t)|D^m f(t)|dt \tag{4.3}$$

where $D^m f$ denotes the m-th derivative of f. The default value for m is 2; the default value for the roughness measure weight λ is the constant 1 and this makes f a cubic smoothing spline.

splinefit

In OCTAVE environment, in order to transform the row data in spline form, it is possible to use the $splinefit$ function. In fact, it allows to fit a piecewise cubic spline with breaks (knots) breaks to the noisy data, x and y.

$$pp = splinefit(x, y, breaks) \tag{4.4}$$

where the input parameters are:

- x: number of knots.

- y: Observed data.

- *breaks*: knots.

instead the output parameter is:

- *pp*: the fitted spline is returned as a piecewise polynomial, pp, and may be evaluated using ppval. The splines are constructed of polynomials with degree order. The default is a cubic, order=3.

FD_mewma

This function creates the *f-MEWMA* control charts for each fault.
The input parameters are:

- `FD_Lisa`: described in subsection *FD_Lisa_Transform2.*
- `vett_m0`: this vector contains the averages of good runs for each variable.
- `vett_var0`: this vector contains the variances of good runs for each variable.
- `lambda`: this is a constant belongs between 0 (not included) and 1.
- `variables`: described in subsection *FD_Lisa_Transform2.*
- `faults`: described in subsection *FD_Transform.*

The output parameters are:

- `XX`: this matrix contains the standardized values of the good runs.
- `T2`: this is a matrix that contains the values of MEMWA to visualize in a control chart.

4.4.3 Matrices Flow

In this section we describe the matrices flow (Figure 4.48) that represents the sequence of matrices generated from the above MATLAB/OCTAVE functions. The final vector is generated by *M-block* and contains the *f-MEWMA* control chart point T^2.

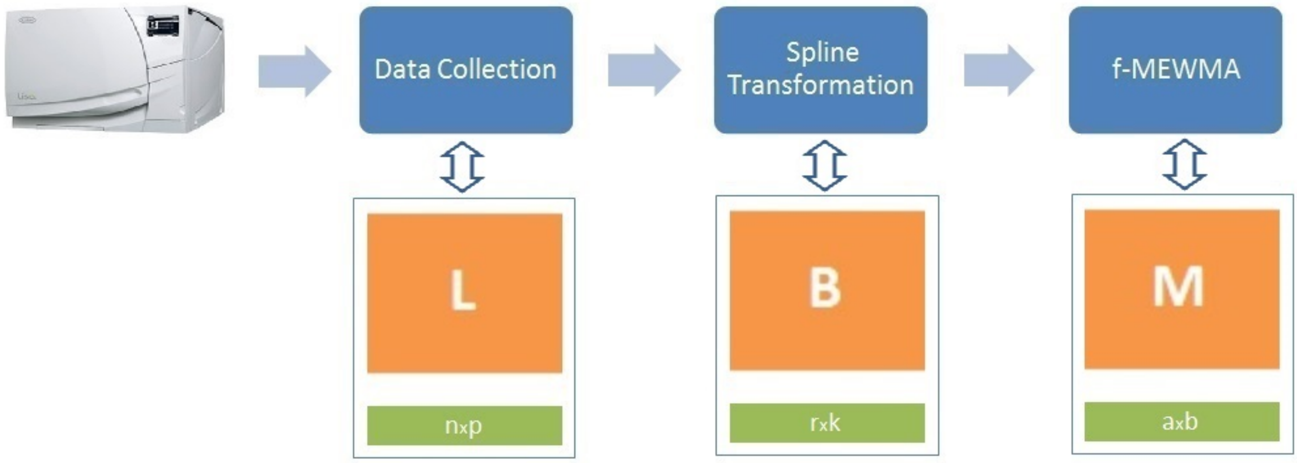


Figure 4.48: Matrices flow.

L-Matrix

This matrix represents the dataset *Lisa_TOT*, that is a dataset that contains the raw data that are processed into the step *Spline Tranformation*.

The *L-matrix* have dimension n x k, in particular:

- **n** (rows): represents the operating time (expressed in seconds) of the sterilization cycles set. In general, n is the sum of the observations of all runs.
- **p** (columns): represents the number of selected variables (e.g., `TChambInt` and `PChamb`). The variables are described in Table 2.3.

B-Matrix

This matrix contains the coefficients of the spline. In particular:

- **r** (rows): represents the number of runs. In detail, $r = g+f$, where g represents the number of *good runs* and f is the number of altered runs that changes according to type of fault (see DOE in Figure 4.1).
- **c** (columns): represents the number of spline coefficients.

M-Matrix

This matrix contains the T^2: a value for each run. These values are plotted on *f-MEWMA* control chart.

The *M-matrix* have dimension a x b, in particular:

- **a** (rows): represents the faults that we described in Section 2.4.
- **b** (columns): represents the number of runs.

Classifier Functions

Below we explain the functions that implement this functional probabilistic classifier.

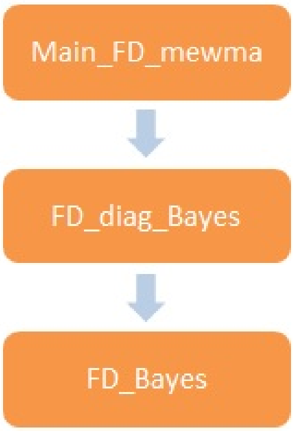


Figure 4.49: Classifier Functions.

FD_diag_Bayes

This function, starting from `FD_Lisa`, calculates the means (centroids) and the variances of each spline coefficient for each fault, excluding a run at a time. This function, for each run, invokes the `FD_Bayes` function.

The input parameters are:

- `FD_Lisa`: described in subsection `FD_Lisa_Transform2`.
- `col_vn1: described in subsection FD_Lisa_Transform2.`
- `variabili: selected variables.`
- `faults: selected faults.`

The output parameters are:

- Guasto_hat:

- P_gS: posterior probability of each run related to particular fault.

FD_Bayes

This function, for each run, calculates the multivariate normal probability density function. This job is implemented for each fault. Therefore, for each fault, of each run, it is calculates $P(E|A_i)$ of the Eq. 3.29. At the end, the function FD_Bayes applies the Bayes formula and calculates the posterior probability $P(A_i|E)$, that is the probability of a determinate fault given the i-th run. Finally, this function calculates the maximum of the posterior probabilities related to each fault for each run.

The input parameters are:

- S: spline coefficients

- mhu_g: means (centroids) of spline coefficients.

- SIGMA_g: variance of spline coefficients.

- P_g: prior probability.

The output parameters are:

- g: identifies the number of fault (it is a counter).

- P_gS: posterior probability.

Therefore, this procedure allows to associate, with a certain probability, the i-th sterilizazion cycle to a determinate fault.

Chapter 5

Conclusions

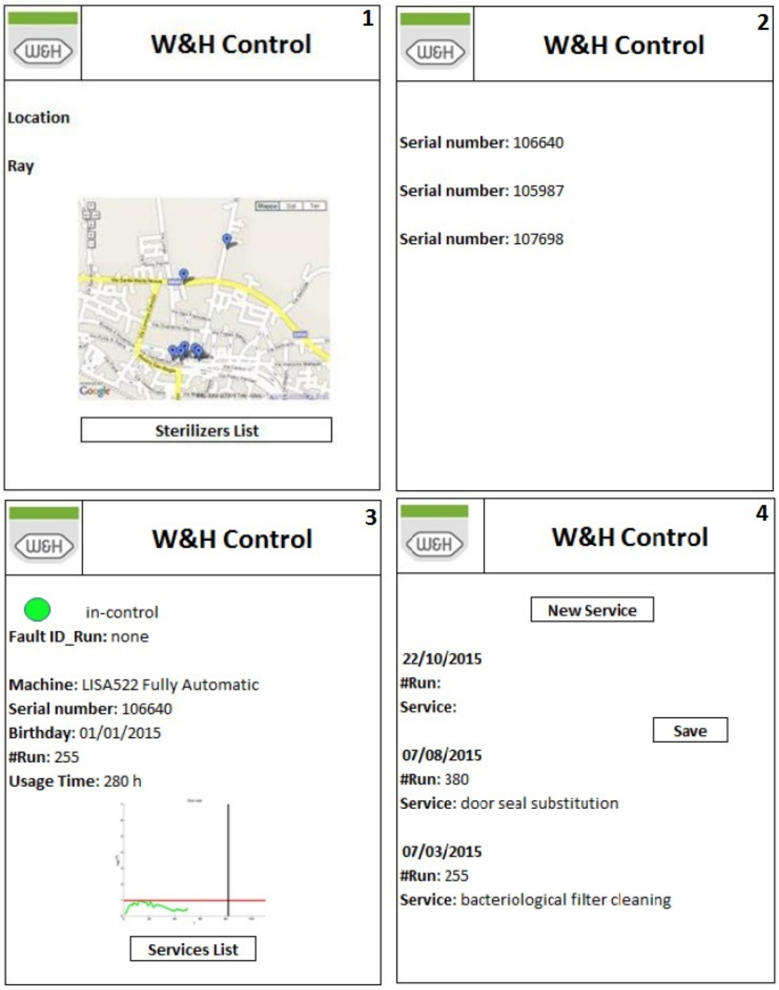
As mentioned, this work is based on a specific requirement of W&H Sterilization to control the health of the sterilizer called *Lisa522 Fully Automatic*. This machine allows dentists to sanitize dental instruments that they use for curing their patients. Thus, it is important to understand if a sterilizer is working well and detect the early onset of anomalies.

Therefore, in this work we considered the statistical surveillance of profile curvature, which is relevant for technological systems whose health is given by nonlinear profiles and deterioration or aging is not related to level or variablity changes, rather it affects profile curvature. In this frame, we proposed the innovative functional Multivariate Exponentially Weighted Moving Average control chart, in short *f-MEWMA*, which is able to manage nonlinear profiles, to monitor the behaviour of a system over time and to highlight early on the onset of anomalies.

In our case the system is a sterilizing machine that, for each sterilization run, measures several variables in high frequency. Each run is transformed in a multivariate functional object and an *f-MEWMA* control chart monitors the variation of these objects over time. According to our experimental data, the new control chart shows an excellent ability to detect anomalies. Finally, an important property of *f-MEWMA* is that it is a generic instrument, therefore, it can be used in various application fields.

About future developments, From the methodological point of view some points are still open and, in light of the interesting results, they are worth to be studied in some details. In particular, further research is necessary to understand the Average Run Length (number of points that, on average, will be plotted on a control chart before an out of control condition is indicated) of *f-MEWMA*. The case of serially correlated runs is also important because is the case raising by delivered machine working in medical centers instead of laboratory data. Moreover the approximation to the χ^2 distribution for the functional detector needs some insights.

Further, the second phase of this work involves the remote control of W&H health sterilizers. In particular, our algorithm may be implemented both on stand alone clients or on a centralized web server. We suggest the second solution in order also to keep a unified database giving valuable knowledge about *LISA* behaviour.



Location

Ray

Serial number: 106640

Serial number: 105987

Serial number: 107698

● in-control
Fault ID_Run: none

Machine: LISA522 Fully Automatic
Serial number: 106640
Birthday: 01/01/2015
#Run: 255
Usage Time: 280 h

22/10/2015
#Run:
Service:

07/08/2015
#Run: 380
Service: door seal substitution

07/03/2015
#Run: 255
Service: bacteriological filter cleaning

Figure 5.1: Lisa App screens

Implementation of our algorithm on a new generation of *LISA* instruments requires a straightforward calibration. This includes:

- the assessment and definition of a set of alternative setups;
- the implementation of a auto-learning procedure working on each new product.

Finally, Fig. 5.1 show the screens of Lisa App that we want to develop in order to allow to maintainers to control the health state of each sold sterilizer.
In particular:

- **Screen 1**: This screen allows to search the sterilizers within a given ray o research (5-10-20 km).
- **Screen 2**: This screen shows the list of sterilizers that are within the selected geographical area.
- **Screen 3**: This screen shows, by a semaphore, the health state of a selected sterilizer (Green: OK; Yellow: Anomaly; Red: Fault).
- **Screen 4**: This screen shows the services of a sterilizer and allows to add new maintenances.

Bibliography

[1] D. Bayart. Walter andrew shewhart. In *Statisticians of the Centuries (ed. C. Heyde and E. Seneta)*, 2001.

[2] T. Lazariv and W. Schmid. Variance charts for time series: A comparison study. *Frontiers in Statistical Quality Control*, pages 77–94, 2015.

[3] N.F. Zhang and A.L. Pintar. Monitoring process variability for stationary process data. *Quality and Reliability Engineering International*, pages 1383–1396, 2015.

[4] C. Lowry, W. Woodall, C. Champ, and S. Rigdon. A multivariate exponentially weighted moving average control chart. *Technometrics*, pages 46–53, 1992.

[5] A. Fassó and S. Locatelli. Asymmetric monitoring of multivariate mata with nonlinear dynamics. *AstA - Advances in Statistical Analysis.*, pages 23–37, 2007.

[6] R. Sparks. Social network monitoring: Aiming to identify periods of unusually increased communications between parties of interest. *Frontiers in Statistical Quality Control*, pages 3–13, 2015.

[7] V. Golosnoy and W. Schmid. Ewma control charts for monitoring optimal portfolio weights. *Sequential Analysis: Design Methods and Applications*, pages 195–224, 2007.

[8] P. Tse, J. Hu, K. Shrivastava, and K. Tsui. A multivariate control chart for detecting a possible outbreak of disease. In *Proceedings of the 7th World Congress on Engineering Asset Management*, pages 593–603, 2012.

[9] S.W. Han and H. Zhong. A comparison of mcusum-based and mewma-based spatiotemporal surveillance under non-homogeneous populations. *Quality and Reliability Engineering International*, pages 1449–1472, 2015.

[10] S. Fan, N. Yao, Y. Chang, and C. Jen. Statistical monitoring of nonlinear profiles by using piecewise linear approximation. *Journal of Process Control*, pages 1217–1229, 2011.

[11] S. Chang and S. Yadama. Statistical process control for monitoring nonlinear profiles using wavelet filtering and b-spline approximation. *International Journal of Production Research*, pages 1049–1068, 2010.

[12] J. Cano, J. Moguerza, S. Psarakisb, and A. Yannacopoulosb. Using statistical shape theory for the monitoring of nonlinear profiles. *Applied Stochastic Models in Business and Industry*, pages 160–167, 2014.

[13] C. Zou and P. Qiu. Multivariate statistical process control using lasso. *Journal of the American Statistical Association*, pages 1586–1596, 2014.

[14] P. Qiu, C. Zou, and Z. Wang. Nonparametric profile monitoring by mixed effects modeling. *Technometrics*, pages 265–277, 2010.

[15] J. Ramsay, G. Hooker, and S. Graves. *Functional Data Analysis with R and MATLAB*. Springer, USA, 2009.

[16] J. Ramsay and B. Silverman. *Applied Functional Data Analysis. Methods and Case Studies*. Springer, 2002.

[17] M. Bohorquez, R. Giraldo, and J. Mateu. Optimal sampling for spatial prediction of functional data. *Statistical Methods and Applications*, pages 1–16, doi:10.1007/s10260–015–0340–9, 2015.

[18] A. Fassó, R. Ignaccolo, F. Madonna, B. Demoz, and M. Franco-Villoria. Statistical modelling of collocation uncertainty in atmospheric thermodynamic profiles. *Atmospheric Measurement Techniques*, pages 1803–1816, 2015.

[19] A. Azzalini and B. Scarpa. *Data Analysis and Data Mining*. Oxford, UK, 2012.

[20] D. Ruppert, M. Wand, and R. Carrol. *Semiparametric Regression.* Cambridge, UK, 2003.

[21] L. Gurrin, K. Scurrah, and M. Hazelton. Tutorial in biostatistics: spline smoothing with linear mixed models. *Statistics in Medicine*, pages 3361–3381, 2005.

[22] P. Green. Penalized likelihood for general semi-parametric regression models. *International Statistical Review*, pages 245–259, 1987.

[23] C. Reinsch. Smoothing by spline functions. *Numerische Mathematik*, pages 177–183, 1967.

[24] D. Montgomery. *Introduction Statistical Quality Control.* John Wiley & Sons, USA, 2009.

[25] S. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, pages 97–102, 1959.

[26] H. Hotelling. *Multivariate Quality Control.* McGraw-Hill, 1947.

[27] W. Woodall and W. D. Montgomery. Some current directions in the theory and application of statistical process monitoring. *Journal of Quality Technology*, pages 78–94, 2014.

[28] S. Prabhu and C. Runger. Designing a multivariate ewma control chart. *Journal of Quality Technology*, pages 8–15, 1997.

[29] E. Alpaydin. *Introduction to Machine Learning.* MIT Press, 2010.

[30] D. Binder. Approximations to bayesian clustering rules. *Biometrika*, pages 275–285, 1981.

[31] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions*, pages 370–418, 1763.