Universitat Autònoma de Barcelona

Multi-objective Optimization and Multicriteria design of PI /PID controllers

Helem Sabina Sánchez Corrales

Universitat Autònoma
de Barcelona

Multi-objective Optimization and Multicriteria design of PI /PID controllers.

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR ENGINEER AT
UNIVERSITAT AUTÒNOMA DE BARCELONA

by

Helem Sabina Sánchez Corrales

June, 2016

Dr. Ramón Vilanova i Arbós, associate professor at Universitat Autònoma de Barcelona,

CERTIFIES:

That the doctoral thesis entitled "**Multi-objective Optimization and Multicriteria design of PI /PID controllers.**" by Helem Sabina Sánchez Corrales, presented in partial fulfillment of the requirements for the degree of Doctor in Telecommunications and Systems Engineering of Department of Telecommunications and Systems Engineering, has been developed and written under his supervision.

Dr. Ramón Vilanova i Arbós
Director

Helem Sabina Sánchez Corrales
Author

Bellaterra, June 2016.

This work was partially supported by:

Universitat Autònoma de Barcelona through the PIF fellowship 461-01-4/2013.
The Spanish CICYT program under grant DPI2010-15230.
The Spanish Ministry of Economy and Competitiveness program under grant DPI2013-47825-C3-1-R.

Preface

Nowadays, the proportional integral and proportional integral derivative are the most used control algorithm in the industry. Moreover, the fractional controller have received attention recently for both, the research community and from the industrial point of view. Owing to this, in this thesis some of the scenarios involves the tuning of these controllers by using the Multi-objective Optimization Design (MOOD) procedure. This procedure focus on providing reasonable trade-off among the conflictive objectives and brings the designer the possibility to appreciate the comparison of the design objectives, this characteristic can be useful for controller tuning.

It is well-known that the controller design is a challenge for the control engineer, satisfying a set of requirements or constraints such as performance, robustness, control effort usage, reliability and others is not an easy task. Sometimes, the improvement of one objective is at the expense of worsening another. This kind of problems where the designer have to deal with the fulfillment of multiple objectives are known as Multi-Objective Problems. Such problems can be addressed using a simultaneous optimization of all targets (multi-objective optimization). This implies to seek for a Pareto optimal solution in which the objectives have been improved as possible without giving anything in exchange (select a design alternative).

Therefore, the first part of this thesis presents the research problem, fundamentals on control system and background on multi-objective optimization. The second part presents the contributions on the MOOD procedure, (i) the Nash Solution as a Multi-criteria decision making technique and (ii) a Multi-stage approach for the Multi-objective optimization process. Hereafter, in the third part we present the contributions for controllers tuning: proportional integral, proportional integral derivative and fractional controller. Finally, some general conclusions and ideas for further research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Proportional-Integral (PI) and Proportional-Integral-Derivative (PID) controller remains as a reliable and practical control solution for several industrial processes. Owing to this the scientific community has shown continued interest in new tuning methods to improve its performance; but also guaranteeing reasonable stability margins for a wide variety of processes [9, 121]. One of the main advantages of PI-PID controllers is their ease of implementation as well as their tuning, giving a good trade-off between simplicity and implementation cost [108].

Since Ziegler and Nichols presented their PID controller tuning rules, a great number of other procedures have been developed, especially those using a low order overdamped model representation of the controlled process, see [69]. Some of them consider only the control system performance of the closed-loop [51, 92]. Hereafter, it was observed that if the designer (control engineer) only considers the performance this leads to control systems with low robustness, due to this fact this specification was introduced into the controller design [10, 44].

It is worth stressing to mention, that the control system design procedure is usually based on use of of low linear models identified at the control system normal operation point [123]. Considering that most of the industrial process must have non-linear characteristics, the designer should take into account the capability of the control system to deal with changes in the controlled process (certain level of robustness).

Moreover, when the changes in the process operating conditions may appear, the controlled variable set-point will need to be changed and then a good transitory

response is required, which is known as a servo-control operation mode. On the other hand, when the set-point remains constant a good load disturbance rejection is needed [103], this is known as a regulatory operation mode.

As a matter of fact, in general, a good disturbance rejection response is not compatible with a good set-point step response and a high performance is often not compatible with a controller which is robust to process model mismatch. Due to this, both operation modes should be considered in the control design. Hence, there are tuning rules devised for regulatory operation mode [51] or for servo operation mode [111, 113]. Moreover, there are some tuning rules that offers both set of tuning for each operation mode, see [49, 127].

Therefore, the design of a control system must take into account two different trade-off such as the *performance/robustness* and also the *servo/regulatory* control operation mode. Hereafter, we looked for a PI-PID controller tuning rules that faces both trade-offs. Nowadays, new tuning techniques have been developed focused on fulfilling several objectives and requirements, sometimes in conflict among them [33, 50, 125]. Some tuning procedures are based on optimization statements [9, 37, 38, 78, 116].

In this context, as there are different conflicting requirements (trade-off) to handle, it is natural to set up a Multi-Objective Problem (MOP) [57], where the designer has to deal with several requirements, and searches for a solution with desirable trade-off between objectives. Recently, PI-PID controller tuning by means of Multi-Objective Optimization (MOO) have been proposed [33, 81, 97, 98, 112]. In fact, satisfying a set of specifications is often a challenge, because most of the times, such specifications are in conflict.

Consequently, the controller design can be therefore be viewed as the search for the best compromise between all the specifications and thereby in this thesis, a procedure for the design of a control system is presented based on a Multi-Objective Optimization Design (MOOD) procedure [88], which is defined by three main steps: the MOP definition (defining the specifications of the problem) , the MOO process (optimization procedure that lead the designer to a set of solutions) and the Multi-Criteria Decision Making (MCDM) stage (consist in select the best solution according to her/his needs and preferences).

The MOOD procedure has shown to be a valuable tool for PI-PID controller tuning for single input, single output (SISO) see [43, 45]. The design methodology is the same for all the considered process and controllers among this work.

One of the purpose of this thesis was to choose a technique for the MCDM stage, in order to select a fair solution that represents the best compromise between the competitive objectives. This stage of the MOOD procedure is an ongoing research, there is not much information in the literature about it; for this reason in this work the Nash Solution (belongs to the bargaining solution [48]) is proposed as a technique to select one point from the Pareto front approximation.

Furthermore, regarding the desirable characteristics related with the optimization problem such as convergence, diversity and preferences handling; regarding the multi-objective problem, they will be associated with constrained, multi-modal, robust, expensive, many-objectives, dynamic or reliability-based optimization instances. Due to this, a Multi-stage approach between a deterministic and evolutionary optimization techniques is formulated for the MOO process, where the designer will expect solutions with a higher reliability, both in a theoretical and practical sense.

Finally, the aforementioned methodology based on the MOOD procedure will be used for addressing control tuning problems. It will be applied on PI, PID and Fractional-Order-PID (FOPID) controllers; using two different approaches for the MOO process, the Normalized Normal Constraint (NNC) algorithm (deterministic algorithm) and a Multi-stage approach in order to find the Pareto front approximation, that it will leads to devise tuning rules based on the Nash Solution as a MCDM technique.

1.1 Structure of the thesis

This thesis is divided in four parts and the contents are organized as follows:

Part I: Fundamentals

In this first part, the aim is to introduce the generalities of the control system and the background of the Multi-objective optimization.

Chapter 2: presents some important concepts about control system configuration, the different trade-off that the control engineer should take into account during the design process such as robustness/performance and the servo/regulatory operation modes.

Chapter 3: in this chapter, a background on MOO is presented and after a general framework is described in order to incorporate the MOO into any engineering design process.

Part II: Contributions for the Multi-objective optimization design procedure

In the second part of this thesis, the idea is to present two contributions for the MOOD; a Nash Solution as a technique to select a solution from the Pareto front approximation and a Multi-stage approach for searching and improve the convergence in order to find the Pareto set approximation.

Chapter 4: this chapter begins with general aspects about the importance of selecting a point from the Pareto front approximation. Hereafter, introduces the Nash Solution as a MCDM technique to select the most reasonable/preferable solution according to designer preferences for a particular situation.

Chapter 5: it is proposed a Multi-stage approach for the MOO process in order to improve the search capabilities and the convergence properties to find a Pareto front approximation.

Chapter 6: in this chapter two case of studies are presented, to validate the Multi-stage approach proposed on Chapter 5.

Part III: Controller tuning applications

In this third part, the MOOD procedure will be used to deal with controller tuning problems such as PI, PID and FOPID controllers.

Chapter 7: presents the application of MOOD procedure based on the Normalized Normal Constraint (NNC) algorithm for PI controllers in order to devise a set of tuning rules for load disturbance rejection task.

Chapter 8: in this chapter a set of tuning rules for one-degree-of-freedom PID controllers is presented. It addresses two different trade-off: the performance/robustness and the servo/regulatory control mode.

Chapter 9: a Multi-stage approach is used to device the balanced tuning for FOPID controller. The obtained rules take into account at once both the servo and the regulatory modes in an optimal way. Moreover, the user can select the desired level of robustness or keep it between given bounds depending on his/her preferences.

Part IV: Conclusions and perspectives

A final part summarizes the conclusions of the thesis and the proposal for future work.

Chapter 10: finally, the conclusions and main contributions are pointed out, also some future work and research to be conducted are presented.

1.2 Contributions of this research

The thesis has generated the following publications:

Journal papers:

- H. S. Sánchez, G. Reynoso-Meza, R. Vilanova and X. Blasco. *Multi-stage Approach for PI Controller tuning based on multi-objective optimization to improve reliability.* Control Engineering Practice (Submitted -Under review).
- H. S. Sánchez, F. Padula, A. Visioli and R. Vilanova. *Tuning rules for robust FOPID controllers based on multi-objective optimization with FOPDT models.* ISA Transactions (Submitted -Under review).

Conference papers

- H. S. Sánchez and R. Vilanova (2013). *Multiobjective tuning of PI controller using the NNC Method: Simplified problem definition and guidelines for decision making.* Proc. of the 18th IEEE Conference on Emerging Technologies & Factory Automation (ETFA). Cagliary (Italy).

- H. S. Sánchez and R. Vilanova (2013). *Nash-based criteria for selection of Pareto Optimal PI controller*. Proc. of the 17th International Conference of System Theory, Control and Computing (ICSTCC). Sinaia (Romania).

- H. S. Sánchez, G. Reynoso-Meza, R. Vilanova and X. Blasco (2013). Comparación de técnicas de optimización multi-objetivo clásicas y estocásticas para el ajuste de controladores PI. XXXIV Jornadas de Automática. Terrasa (España).

- H. S. Sánchez and R. Vilanova (2014). *Optimality comparison of 2DoF PID implementations*. Proc. of the 18th International Conference of System Theory, Control and Computing (ICSTCC). Sinaia (Romania).

- H. S. Sánchez and R. Vilanova (2014). *Implementación de controladores PID: Equivalencia y optimalidad*. XXXV Jornadas de Automática. Valencia (España).

- H. S. Sánchez, A. Visioli and R. Vilanova (2015). *Nash Tuning for Optimal Balance of the Servo/Regulation Operation in Robust PID Control*. Proc. of the 23th Mediterranean Conference on Control & Automation. Málaga (Spain).

- H. S. Sánchez and R. Vilanova (2015). *Multi-objective optimization for control and process operation*. Proc. of the Doctoral Consortium on Informatics in Control, Automation and Robotics (DICINCO/ICINCO). Colmar, Alsace (France).

- H. S. Sánchez, G. Reynoso-Meza, R. Vilanova and X. Blasco (2015). *Multistage procedure for PI controller design of the Boiler Benchmark problem*. Proc. of the 20th IEEE Conference on Emerging Technologies & Factory Automation (ETFA). Luxembourg.

Journal paper in collaboration:

- G. Reynoso-Meza, H. S. Sánchez, L. dos Santos Coelho and R. Zanetti (2016). *Multidisciplinary optimisation in mechatronic systems: a comparative analysis with multiobjective techniques*. IEEE Latin America Transactions. Vol. 14(1). Pp. 364-370.

Conference paper in collaboration:

- G. Reynoso-Meza, H. S. Sánchez, X. Blasco and R. Vilanova (2014). *Reliability based multiobjective optimization design procedure for PI controller tuning.* Proc. of the 19th World Congress The International Federation of Automatic Control (IFAC). Cape Town (South Africa).

Part I

Fundamentals

Chapter 2

Control System

The purpose of a control system is to obtain a desired response for a given process. We consider the typical feedback control system represented in Figure 2.1, where $P(s)$ is the process, $K(s)$ is the controller, $r(s)$ is the set-point signal, $u(s)$ is the control signal, $d(s)$ is the load disturbance signal, $y(s)$ is the process output and $e(s) := r(s) - y(s)$ is the control error.

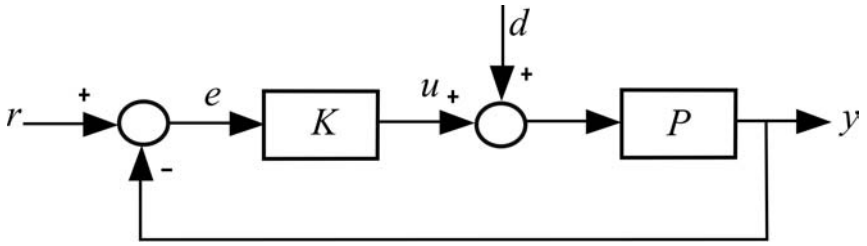


FIGURE 2.1: Feedback control scheme.

It is worth stressing to mention that the control problem consists in finding and selecting the parameters of the controllers, in order to ensure good levels of robustness and performance of the process $P(s)$. To find these optimal parameters for the controller some information about the process will be necessary. According to [123] the First-Order-Plus-Dead-Time (FOPDT) model is frequently used in process control because is simple and describes with sufficient accuracy the dynamics of many industrial process. Therefore, the process dynamics will be described by consider a stable process represented by a FOPDT model of the form:

$$P(s) = \frac{K}{1+Ts} e^{-Ls}, \tag{2.1}$$

where K is the process gain, T is the time constant and L is the dead time. The process dynamics can be fully characterized in terms of the normalized dead-time defined in (2.2), which represents a measure of the difficulty in controlling the process:

$$\tau = \frac{L}{T}. \tag{2.2}$$

Further, the estimation of the parameters of this kind of model can be performed by means of a very simple step-test experiment (or historical data) to be applied to the process. This represents a clear advantage in industry with respect to more advanced identification methods.

The process will be controlled with a one-degree-of-freedom PID controller with a derivative time filter, whose transfer function is given by:

$$K(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d}{N} s} \right) \tag{2.3}$$

where K_p is the proportional gain, T_i is the integral time constant and T_d is the derivative time constant. The derivative time noise filter constant N usually takes values within the range 5-33 [6, 123]. Here, without loss of generality, the value $N = 20$ has been selected [127].

2.1 Servo & Regulatory operation modes

The design of the closed-loop control system must consider the trade-off among the disturbances and set-point changes, referring to the servo and regulatory mode. Consider the closed loop of the system in Figure 2.1, where the process variable of the control system is given by

$$y(s) = \frac{K(s)P(s)}{1 + K(s)P(s)} r(s) + \frac{P(s)}{1 + K(s)P(s)} d(s) \tag{2.4}$$

The process output $y(s)$ depends in general on two input signals, the reference signal $r(s)$ and the disturbance signal $d(s)$. Depending of the input signal, the system can operate in two different modes:

- *servo operation mode*: a good tracking of the signal reference r is the main control task;
- *regulation operation mode*: keeping the process variable at the desired value, in spite of possible disturbances d, is of main concern.

In the first case, that is when disturbances are not taken into account, the output signal can be represented as

$$y_{sp}(s) := \frac{K(s)P(s)}{1 + K(s)P(s)} r(s) \tag{2.5}$$

while in the second case, when the reference signal is not taken into account, we have that the process variable is

$$y_{ld}(s) := \frac{P(s)}{1 + K(s)P(s)} d(s) \tag{2.6}$$

From the point of view of output performance, we can identify a design trade-off by considering the effects of load disturbances and set-point changes on the feedback control system. On the other side, there is also a trade-off between performance and robustness. For example, the SIMC method [106], allows the adjustment of the robustness/performance trade-off by means of a single tuning parameter governing the closed-loop bandwidth, while at the same time guaranteeing a good balance between servo and regulatory control. It is worth stressing this point because in the literature design techniques normally focus on performance in either servo or regulatory mode, see for example [121] and [69] for a historical review.

However, both servo and regulatory requirements must be addressed, for example,in cascade configurations: the inner loop should be tuned based on tracking as it receives the set-points from the master loop. Nevertheless, the inner lop may also need acceptable input disturbance suppression capabilities. In addition there are also cases with both input and output disturbances [106, 110], or when one simply does not know where the disturbance may occur [101]. However, it is obvious that the control system is more complex and it does not solve completely the problem if disturbances occur in the part of the process with the slowest dynamics.

In general, it has to be taken into account that obtaining a fast load disturbance response usually implies increasing the bandwidth of the control system at the expense of a more oscillatory set-point step response [7]. Further, a decrement

of the settling time of the response can usually be obtained at the expense of a decrement of the robustness of the system (and of an increment of the control effort) [35]. It appears therefore that the tuning of the controller is critical if a one-degree-of-freedom PID controller is used. Moreover, there is the situation when the control loop should operate in both servo and regulatory mode, it may not be clear which setting the designer should implement.

When both tasks have to considered, the tuning of the PID controller becomes difficult as the two specifications are conflicting (achieving a high performance in the load disturbance rejection tasks requires in general a high bandwidth of the control system which yields poor stability margins and large overshoots in the set point step responses [123]).

2.2 Performance & Robustness

The control system design is normally based on the use of linear models, obtained at the system normal operating point, to represent the nonlinear controlled process. Since the beginning the controller tuning only take under consideration the performance [6]. Further, it was noticed that if only the performance was considered, the resulting closed-loop control system probably will have a very low robustness. Further, if the system is designed to have a high robustness [41] without evaluating the performance, the designer will not have any indication of the cost of having such high level of robustness. Owing to this, the robustness and performance trade-off in control system is a well-know issue.

Some choices to evaluate the performance and robustness, the following standard measures will be used along this thesis:

- Integral Square Error (ISE): it will penalize large errors (since the square of a large error will be much bigger) [6].

$$ISE := \int_0^\infty e^2(\tau)d\tau \tag{2.7}$$

- Integral of the Absolute Error (IAE): the most natural way to measure performance is by minimizing the integrated-absolute-error. In general a

low overshoot and a low settling time at the same time [102] in the set-point or load disturbance step response.

$$IAE := \int_0^\infty |e(\tau)| d\tau \tag{2.8}$$

- Total Variation of control action (TV): is a measure of the smoothness of control action [123]. In order to evaluate the manipulated input usage $u(t)$.

$$TV := \sum_{k=1}^{\infty} |u(k+1) - u(k)| \tag{2.9}$$

- Robustness: is the peak of the Maximum sensitivity (M_s), it represents the inverse of the minimum distance of the Nyquist plot from the critical point (-1,0). This index is an indication of the system robustness (relative stability) and typical values range form 1.4 (robust tuning) to 2.0 (aggressive tuning) [6] or fixed to a specific value or constrained to a given range.

$$M_s := \max_{w \in [0,+\infty)} \left| \frac{1}{1 + K(s)P(s)} \right|_{s=jw}. \tag{2.10}$$

2.3 Motivation Example

In order to show how the performance of the system is affected when the controller implemented is not operating according to the tune mode, and example is presented. Consider the following process, taken from [127]

$$P(s) = \frac{e^{-0.5s}}{(s+1)^2} \tag{2.11}$$

Tuning	K_p	T_i	T_d
servo	1.66	1.69	0.51
regulatory	2.41	1.0	0.56

TABLE 2.1: PID controller parameters for P_1

The process has been modeled as FOPDT with $K = 1$, $T = 1.65$ and $L = 0.99$. The optimal ISE tuning for servo and regulatory mode proposed by [127] have been applied, the parameters are shown in Table 2.1. As it can be seen in Figure

2.2 the control system is operating in both, servo and regulation mode. Showing that if the controller was tuned for servo mode it will be concern to the set-point changes and not to disturbances that are affecting the plant input. Moreover, it can be observed that the load-disturbance response of the set-point tuning is closer to the response obtained with the regulation tuning. In general, the load-disturbance response of the set-point tuning is closer to the optimal regulation, which leads the designer to choose the set-point setting.

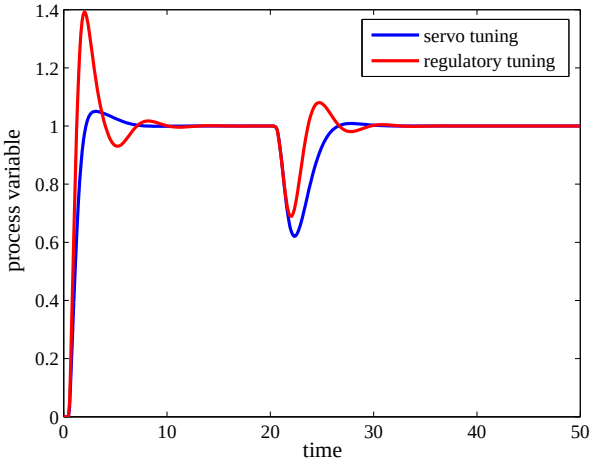


FIGURE 2.2: Process response for servo and regulatory mode for system P_1.

Therefore, the issues mentioned before can be seen as motivating example of the analysis presented in the next chapters, which means that a balance tuning is needed, that considers the trade-off between *servo/regulation* operation modes and the *robustness/performance*. Owing to this, the idea of implementing the MOO can be an alternative to solve this kind of problems, in order to find the optimal parameters for the controller and considering the different trade-off aforementioned. Bearing in mind the previous reasoning, it appears that the control problem has a clear multi-objective nature.

Chapter 3

Multi-objective Optmization

3.1 Generalities

It is worth stressing to mention that there are two different approaches to solve an optimization statement for an Multi-objective problem (MOP) according to [59]; first, the Aggregate Objective Function (AOF) where the designer needs to describe all the trade-off at once and from the beginning of the optimization process, for example, the designer can use a weighting vector to indicate relative importance among the objective. Secondly, the Generate-First Choose-Later ($GFCL$) approach in which the target is to generate a set of Pareto optimal solutions and then the designer will select, a posteriori, the most preferable solution according to his/her preferences [56].

In order to generate such set of desirable solutions in the GFCL approach, the Multi-Objective Optimization (MOO) techniques might be used. Such techniques generate what is called the Pareto front approximation, where all the solutions are Pareto optimal. This means that there is no solution that is better in all objectives, but a set of solutions with different trade-offs among the conflicting objective.

Therefore, a set of optimal solutions is defined as the Pareto set $\boldsymbol{\Theta_P}$ and each solution within this set defines an objective vector. The projection into the objective space is known as Pareto front $\boldsymbol{J_P}$. All the solutions in the Pareto front are said to be non-dominated and Pareto-optimal solutions. This means in the Pareto front, there is not a solution that is better than another one for all the

competitive objectives. To improve one objective will imply to introduce a loss regarding the other ones. It is important to mention that the true Pareto front is unknown, for this reason MOO techniques search for a discrete description of the Pareto set $\boldsymbol{\Theta}_{\boldsymbol{P}}^{*}$ capable of generating a good approximation of the Pareto front $\boldsymbol{J}_{\boldsymbol{P}}^{*}$, see Figure 3.1. In this way, the decision maker can analyze the set and select the most preferable solution. This set of solutions implies that there is flexibility at the decision making stage. The role of the designer is to select the most preferable solution for a particular situation.

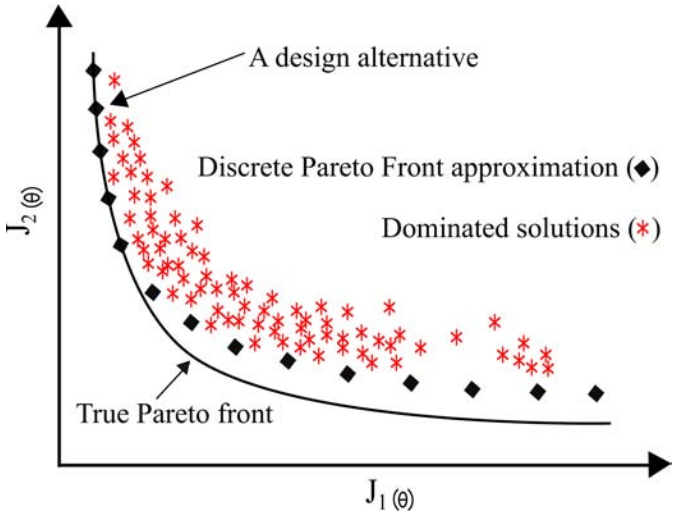


FIGURE 3.1: Pareto front concept for two objectives.

3.2 Multi-objective Optimization Design

In order to incorporate the MOO process into any engineering design, a Multi-objective Optimization Design (MOOD) procedure should be carried out [24]. The MOOD procedure have shown to be a valuable tool for control engineers, see [85, 89, 96, 100]. This procedure allows the designer to be more involved with the design process and to evaluate the performance exchange between the conflicting objectives, the follow step are needed: i) the MOP definition (objectives, decision variables and constraints), ii) the MOO process (search) and iii) the MCDM stage (analysis and selection) and is represented in Figure 3.2. Hence, to obtain the best trade-off from the Pareto front approximation for the controller tuning, the aforementioned steps are needed [88]; this would guarantee the possibility of obtaining a desirable trade-off among the design alternatives or for the trade-off

analysis of the requirements of the controller tuning. Such procedure has shown to be a valuable tool for PI-PID controller tuning for single input, single output (SISO) see [43, 45] where a MOOD have been implemented.

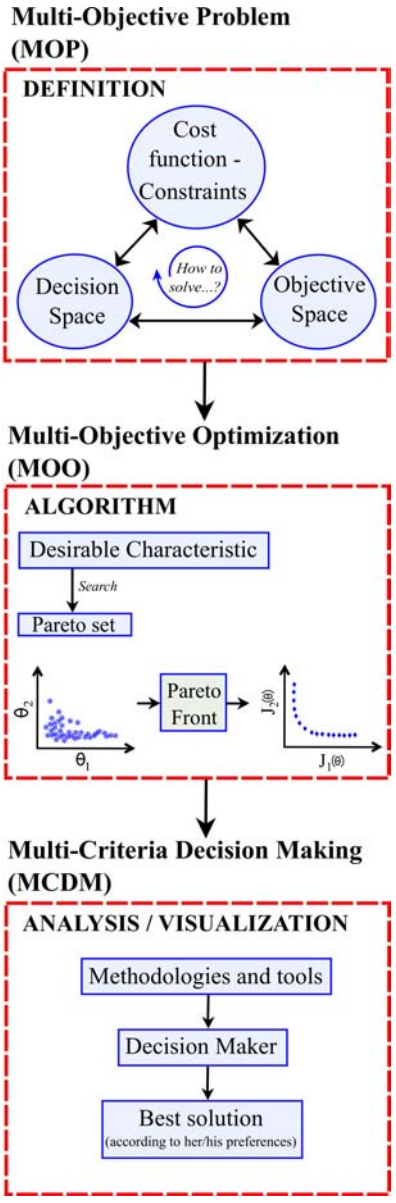


FIGURE 3.2: A Multi-objective Optimization Design (MOOD) procedure.

3.2.1 MOP definition

In this first step, the designer decides *How to solve the problem*?, which are the requirements and constraints. It is important to define the decision space (for the parameters values) it will generate the objective space.

As it has mentioned in [62], a MOP can always be expressed as follows:

$$\min_{\boldsymbol{\theta}\in\Re^n} \boldsymbol{J}(\boldsymbol{\theta}) \quad = \quad [J_1(\boldsymbol{\theta}), \ldots, J_m(\boldsymbol{\theta})] \in \Re^m \tag{3.1}$$

subject to:

$$\boldsymbol{g}(\boldsymbol{\theta}) \leq 0 \tag{3.2}$$

$$\boldsymbol{h}(\boldsymbol{\theta}) = 0 \tag{3.3}$$

$$\theta_{li} \leq \theta_i \leq \theta_{ui},\ i = [1, ..., n] \tag{3.4}$$

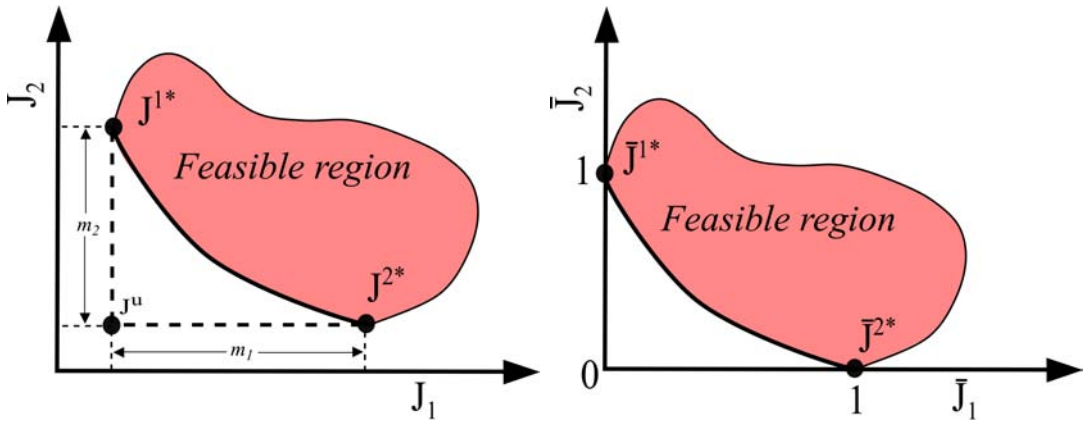
where $\boldsymbol{\theta} = [\theta_1, ..., \theta_n]$ is defined as the decision variables vector, $\boldsymbol{J}(\boldsymbol{\theta})$ is the objective vector, $\boldsymbol{g}(\boldsymbol{\theta})$ and $\boldsymbol{h}(\boldsymbol{\theta})$ are the inequality and equality constraint vectors, respectively, and θ_{li} and θ_{ui} are the lower and upper bounds in the decision space of the θ_i variable.

In this thesis, the formulation of the MOP it will be related with the settings of PI-PID-FOPID controllers in order to satisfy the set of requirements that it will offer the best trade-off among the conflicting objectives.

3.2.2 MOO process

The selection of the optimizer is important to achieve the statement goals in the MOP. The designer should select an optimizer according to the problem at hand and it would look for some desirable characteristics, like convergence, diversity, robust, etc. After describing the MOP, the designer faces another difficult task which is to pass this problem statement to the optimizer because in many cases the point of view of the designer is not the same as that of the optimizer [32].

MOO process has already been proposed for the tuning of PI-PID controllers [43, 46, 68, 88, 95, 97–99, 112, 126]. In [56] reviews some classical multi-objective optimization methods for engineering. However, some experiments conducted in this thesis, incorporate the the Normalized Normal Constraint (NNC) algorithm to generate the set of optimal solutions.



(a) (b)

FIGURE 3.3: (a) General metrics and the non-normalized design space (b) Normalized design metrics.

Normalized Normal Constraint Algorithm (NNC)

The NNC is a deterministic algorithm proposed in [61]. This algorithm incorporates a critical linear mapping of the design objectives. This mapping has the desirable property that the resulting performance of the method is entirely independent of the design objectives scales and in the ability to generate a well distributed set of Pareto points even in numerically demanding situations. Using this algorithm, the optimization problem is separated into several constrained single optimization problems. After a series of optimizations, a set of evenly distributed Pareto solutions results. In order to have an overview of the algorithm, lets consider the MOP described in (3.1). This algorithm consist in seven step process:

1. *Anchor points*: The anchor points are obtained by minimizing the objective functions J_1 and J_2, all anchor points are one unit away from the Utopia point (J^u). Finding the anchor points determine the limits of the Pareto front, see Figure 3.3.

2. *Normalization*: is an important part of the method, without this process, certain important regions of the Pareto frontier would be under-represented in the Pareto set. In Figure 3.3b it shows the normalized form of J, which is $\bar{J}$.

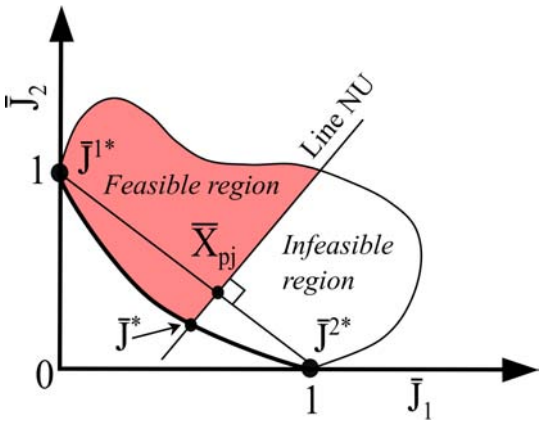


FIGURE 3.4: NNC algorithm for bio-objective problems.

3. *Utopia Line Vector*: is the difference between the normalized anchor points,

$$\bar{Ut} = \bar{J}^{2*} - \bar{J}^{1*} \tag{3.5}$$

4. *Normalized Increments*: the utopia line vector is segmented at equidistant points, denoted by , m is the number of solutions we want to obtain.

$$\delta = \frac{1}{m_1 - 1} \tag{3.6}$$

5. *Generate Utopia Linea Points*: a set of evenly distributed points on the utopia line will be evaluated as,

$$\bar{X}_{pj} = \alpha_{1h}\bar{J}^{1*} + \alpha_{2h}\bar{J}^{2*} \tag{3.7}$$

where, $0 < \alpha_{1j} < 1$.

6. *Pareto points generation*: a set of well-distributed solutions is generated in the normalized space. For each $\bar{X_{pj}}$ generated, a corresponding Pareto solution is obtained.

7. *Pareto Filter*: is in charge of filtering the set of solution obtained in the last step, removing all the not globally Pareto solution.

8. *Smart Filter*: is based on the idea that certain regions of the Pareto frontier can be considered less useful than others. Although no Pareto solution is objectively better than another Pareto solution, the designer may consider some Pareto solutions more desirable than others. This filter essentially gives a smaller set of Pareto points that adequately represents the properties of

compensation for the Pareto front [61]. If a point is not globally Pareto it will be eliminated from the original solution.

Algorithm I: *Normalized Normal Constraint*
1 : Generate the anchor points $J\|_i(x)$ for each objective.
2 : Calculate the Utopian point and NADIR.
3 : Normalized the objectice space.
4 : Generate the utopian hyperplane.
5 : Definition of the normalized increments.
6 : Generate the utopia lines.
7 : **while** *normalized increments* **do**
8 : | Optimize;
9 : **end**
10 : Algorithm concludes.

FIGURE 3.5: NNC Algorithm outline.

Implementing this algorithm[1] the designer ensures to have the significant regions of the corresponding trade-off among the competitive objectives. The NNC algorithm is presented here to solve a bi-objectives problem, see Figure 3.5 but it can be generalized to *n-objectives*. For more details about this algorithm see [59, 61].

3.2.3 MCDM stage

All points within the Pareto front are equally acceptable solutions. Once the Pareto front approximation is provided, the designer needs to choose one of those points as the final solution to the MOP for the implementation phase. Several tools and methodologies are available, in order to facilitate the decision making stage [16, 20, 27, 47, 117], a review with different techniques for decision making analysis can be consulted in [30] and a taxonomy to identify the visualizations is presented in [83].

Somehow, the decision making can be undertaken by using two different approaches: i) by including additional criteria such that at the end only one point from the Pareto front satisfies all of them, and ii) by considering one point that represents a fair compromise between all used criteria. From a controller design

[1]The NNC algorithm is available in Matlab Central at
http://www.mathworks.com/matlabcentral/fileexchange/38976

point of view, the first option can be used to improve the control performance by introducing additional criteria. In other words, as the MOP establishes the search among the Pareto front for a compromise among a set of performance indices, and additional performance (probably of secondary importance) can be introduced. In this way, a new optimization problem will start with the search domain located in the Pareto set in order to find the best solution. The second option does not introduce more information for the decision making and a *fair* point should be selected in order to represent an appropriate trade-off among the different considered cost functions. In the context of finding a PID controller tuning rule, this second option has been preferred because it can be somehow somehow easily automated. It means that a single proposal for the controller design will be the outcome for the MOP. Obviously, the ideal setup would be to reach the utopia point. However, the utopia point is normally unattainable and does not belong to the Pareto front approximation. This is because it is not possible to optimize all individual objective functions independently and simultaneously. Thus, it is only possible to find a solution that is as close as possible to the utopia point. Such solution is called the *compromise solution* (CS) and is Pareto optimal. This approach however, starts from a neither attainable nor feasible solution. Therefore it is not very practical as it does not take into account what can be achieved for each one of the individual objectives functions. Another procedure to select a fair point is to use bargaining games [11]. This solution leads us to a practical procedure for choosing a unique point from the Pareto front, as it will be seen in the next chapter.

Part II

Contributions for the Multi-objective optimization design procedure

Chapter 4

Guidelines for decision making

4.1 Multi-objective Problem motivation

In order to show, the importance of constraining a MOP and choosing a point from the Pareto front, this chapter is devoted to present a representative example.

Consider a closed-loop control system as shown in Figure 2.1, the process is represented by a FOPDT model given by

$$P(s) = \frac{1}{s+1}e^{-0.5s} \tag{4.1}$$

In this case, we will only deal with disturbance attenuation, therefore we can assume without loose of generality that $\beta = 1$ (as usual with the majority of industrial PI controllers). Equation (4.2) shows the transfer function of the selected structure of the PI controller.

$$K(s) = K_p\left(1 + \frac{1}{T_i s}\right) \tag{4.2}$$

4.1.1 Problem Statement

The design of the PI controller is performed in two cases:

1. On the basis of the previous chapters, the MOP (4.3) will be formulated in order to find the parameters of the feedback controller (4.2) required to obtain the desired regulatory control performance.

2. To the previous MOP we will add an equality constraint, to three usual levels of robustness as

   - $M_s = 1.4$: high robustness level.
   - $M_s = 1.6$: medium-high robustness level.
   - $M_s = 1.8$: medium-low robustness level.

Therefore, the MOP is stated as

$$\min_{\boldsymbol{\theta_c}} \boldsymbol{J}(\boldsymbol{\theta_c}) \quad = \quad [J_{IAE}(\boldsymbol{\theta_c}), J_{TV}(\boldsymbol{\theta_c})] \tag{4.3}$$

where

$$\boldsymbol{\theta_c} = [K_p, T_i]$$

The minimization is constrained by using the maximum sensitivity, that is:

$$M_s = \max_{w\in[0,+\infty)} \left| \frac{1}{1 + K(s;\boldsymbol{\theta_c})P(s)} \right|_{s=jw} \tag{4.4}$$

As stated in Section 2.2, M_s can be fixed to a specific value as an equality constraint,

$$\boldsymbol{h}(\boldsymbol{\theta_c}) = h, \tag{4.5}$$

where $\boldsymbol{h}(\boldsymbol{\theta_c}) = M_s$ and $h \in \{1.4, 1.6, 1.8\}$.

4.1.2 Generation of the Pareto front

In order to generate the set of optimal solutions, the NNC algorithm [61] is implemented, see Figure 3.5.

The results obtained for the first case are shown in Figure 4.1, where a Pareto front in the (IAE, TV) space is obtained and the location of the initial guess corresponding to the Ziegler-Nichols (ZN) tuning [41] are displayed. It is worth

stressing to mention, that a good initial guess will help the algorithm to find the optimal area.

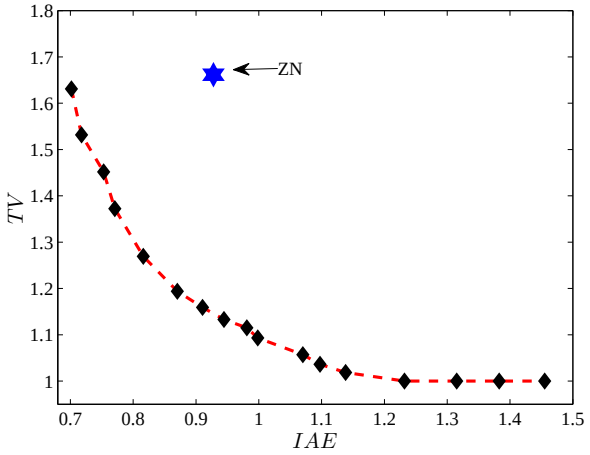


FIGURE 4.1: Pareto front for Ps, where $K = 1$ and $\tau = 0.5$

Furthermore, the results in Figure 4.2 correspond to the second case in which the MOP considered has the value of M_s constrained to each one of the specific values of the set (4.5); what we are doing is constraining the achievable system performance. As it can be noticed the new Pareto sets corresponding to each of the robustness levels are jointly shown with the unconstrained one. Therefore, we get a new Pareto front, considerable smaller that the previous one, constraining ourselves to a really small set of possible solutions.

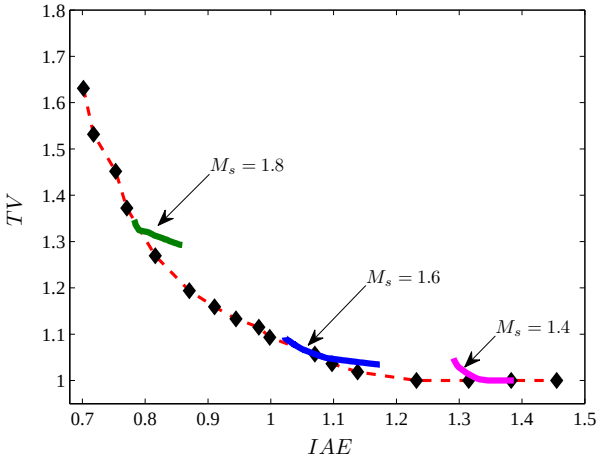


FIGURE 4.2: Comparison of Pareto fronts

With the example at hand, a deep analysis has been performed where it seems that there is a correlation between the value of M_s and the total variation TV.

Some previous results such as the ones in [31] provide the findings that the M_s correlates well with the input usage as given by its TV. Such a correlation is reasonable since a large M_s corresponds to an oscillatory system with large input variations. This is clearly shown in Figure 4.3, therefore for this statement the robustness can be directly associated to the TV performance index and think on that index not just as the input usage but also as a measure of the closed system robustness and also the approximate relation is established $M_s \approx 0.84TV + 0.68$.

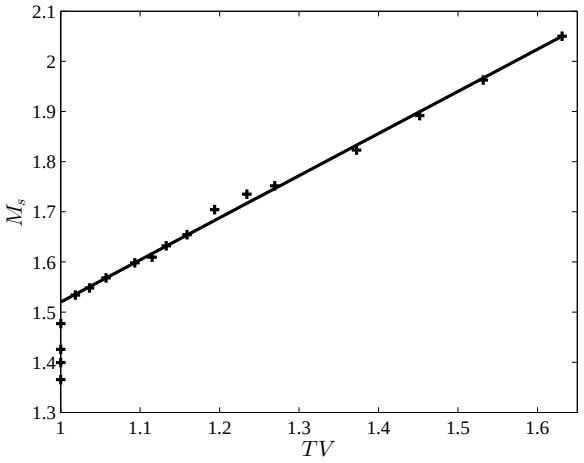


FIGURE 4.3: TV and M_s correlation, $M_s \approx 0.84TV + 0.68$.

This allows the designer to think, in the selection of the appropriate point from the Pareto front will need to take into account the level of robustness we need (lower values for TV). Effectively, this simplifies the setup for the optimization problem and subsequent generation of the Pareto front.

Having in mind, that solving a MOP there is no unique global solution, indeed, this is what is expressed in the Pareto front: there is no way to improve a point without increasing the value of at least one of the objective functions. All the points within the Pareto front are equally acceptable solutions. However, there is the need to choose one of such points as the final solution for the MOP. As it was mentioned in Section 3.2.3, one alternative could be to introduce a selection criteria in order to choose one point, probably the more intuitive option would be the solution that is as close as possible to the utopia point: *Compromise Solution* (CS). The CS is computed by choosing the point on the Pareto front that minimizes the 2-norm from the utopia point. Another option is to use the Nash solution (NS), this alternative belongs to the bargaining games [11]. This solution provides a quite simple and direct approach for selecting one point from the Pareto front.

4.2 Bargaining and *trade-off* solutions selection

In a transaction, when the seller and the buyer value a product differently, a surplus is created. A bargaining solution is then a way in which buyers and sellers agree to divide the surplus. There is an analogous situation regarding a controller design method that is facing two different cost functions for a system. When the controller locates the solution on the disagreement point (D), as shown in Figure 4.4, there is a way for the improvement of both cost functions. We can move within the feasible region towards the Pareto front in order to get lower values for both cost functions. Let θ_1^* and θ_2^* denote the values for the free parameter vector θ that achieve the optimal values for each one of the cost functions f_1 and f_2, respectively. Let these optimal values be $f_1^* = f_1(\theta_1^*)$ and $f_2^* = f_2(\theta_2^*)$. On that basis, the utopia point will have coordinates f_1^* and f_2^* whereas the disagreement point will be located at $(f_1(\theta_2^*), f_2(\theta_1^*))$. As the utopia (U) point is not attainable, we need to analyze the Pareto front in order to obtain a solution. A fair point that represents an appropriate trade-off among the cost functions f_1 and f_2 is defined by the coordinates $(f_1^{Pf}, f_2^{Pf}) = (f_1(\theta_1^{Pf}), f_2(\theta_2^{Pf}))$, where the superindex Pf means Pareto front. On the basis of this formalism, we can identify, in economic terms the benefit of each one of the cost functions (buyer and seller) as the differences $f_1(\theta_2^*) - f_1^{Pf}$ and $f_2(\theta_1^*) - f_2^{Pf}$. The bargaining solution will provide a choice for (f_1^{Pf}, f_2^{Pf}) therefore a benefit for both f_1 and f_2 with respect to the disagreement. It is important to notice that the problem setup is completely opposite to the one that generates the compromise solution (CS) as the closest one to the utopia point.

Formally, a bargaining problem is denoted by a pair $< S; d >$ where $S \in \mathbb{R}^2$, $d \in S$ represents the disagreement point and there exists $s = (s_1, s_2) \in S$ such that $s_i < d_i$. In our case, S is the shaded area shown in Figure 4.4 delimited by the Pareto front and its intersection with the axis corresponding to the coordinates of the disagreement point. In Figure 4.4, different solutions for selecting a point from the Pareto front can bee seen:

1. The disagreement solution (D): it is the solution associated to the disagreement point. Even, if it is not the preferred solution for none of the players, it is a well-defined solution.

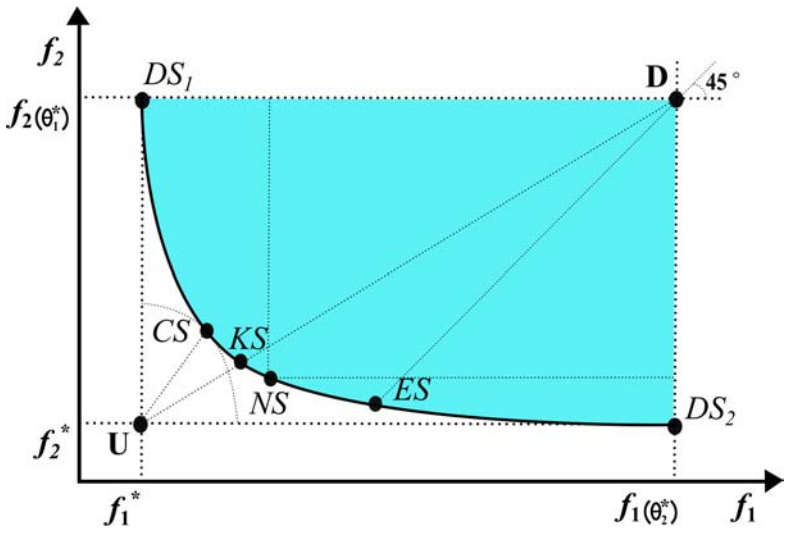


FIGURE 4.4: Location of the bargaining solutions into the Pareto front.

2. The dictatorial solution for player 1 (DS1): it is the point that minimize the cost function for player 1. The same concept can be applied to player 2, yielding the dictatorial solution for player 2 (DS2).

3. The egalitarian solution (ES): it is the greatest feasible point (f_1^{Pf}, f_2^{Pf}) that satisfies $f_1(\theta_2^*) - f_1^{Pf} = f_2(\theta_1^*) - f_2^{Pf}$. This point coincides with the intersection of the 45°diagonal line that passes through the disagreement point with the Pareto front.

4. The Kalai-Smorodinsky solution (KS): it is the point (f_1^{Pf}, f_2^{Pf}) corresponding to the intersection of the Pareto front with the straight line that connects the utopia and the disagreement point.

5. The Nash Solution (NS): it selects the unique solution to the following maximization problem:

$$\begin{aligned} \max_{(f_1^{Pf}, f_2^{Pf})} \quad & (f_1(\theta_2^*) - f_1^{Pf})(f_2(\theta_1^*) - f_2^{Pf}) \\ s.t. \quad & f_1^{Pf} \leq f_1(\theta_2^*) \\ & f_2^{Pf} \leq f_2(\theta_1^*) \end{aligned}$$

In order to illustrate the location of the different solutions that can be selected from the Pareto front, consider Ps. The different solutions that the designer can obtain using the bargaining concept can be seen in Figure 4.5. It is important to highlight that in some cases the NS matches with KS: this happens when both negotiators are in a neutral risk (see, for example, [5]).

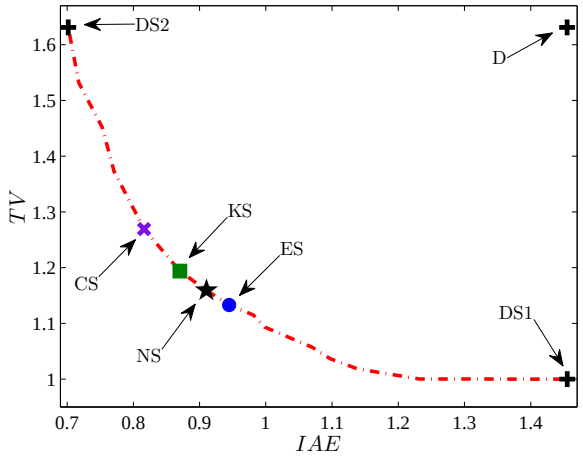


FIGURE 4.5: Pareto front for $P(s)$. Location of the bargaining solutions.

4.3 Nash Solution

In his pioneering work on bargaining games, Nash in [48] established a basic two-person bargaining framework between two rational players, and proposed an axiomatic solution concept which is characterized by a set of predefined axioms and does not rely on the detailed bargaining process of players. Nash proposed four axioms that should be satisfied by a reasonable bargaining solution:

- Pareto efficiency: none of the players can be made better off without making at least one player worse off.
- Symmetry: if the players are indistinguishable, the solution should not discriminate between them. The solution should be the same if the cost function axis are swapped.
- Independence of affine transformations: an affine transformation of the cost functions and of the disagreement point should not alter the outcome of the bargaining process.
- Independence of irrelevant alternatives: if the solution (f_1^{Pf}, f_2^{Pf}) chosen from a feasible set A is an element of a subset $B \in A$, then (f_1^{Pf}, f_2^{Pf}) must be chosen from B.

Nash proved that, under mild technical conditions, there is a unique bargaining solution called *Nash bargaining solution* satisfying the four previous axioms. Indeed,

by considering the different options for selecting a point from the Pareto front, the NS is the only solution that satisfies these four axioms [48]. In fact, the Nash solution is simultaneously utilitarian (Pareto efficient) and egalitarian (fair). Also from a MOO point of view, by maximizing the product $(f_1(\theta_2^*)-f_1^{Pf})(f_2(\theta_1^*)-f_2^{Pf})$, we are maximizing the area of the rectangle that represents the set of solutions dominated by the NS. Actually, the NS provides the Pareto front solution that dominates the larger number of solutions, therefore being absolutely better (that is, with respect to both cost functions at the same time) than any one of the solutions of such rectangle. These are the reasons why the NS represents an appropriate choice for the (semi)-automatic selection of the fair point from the Pareto front.

4.3.1 Comparison Examples

In this section, the previous ideas will be used to locate well known tuning rules into the Pareto front and tho show how the NS selection performs compared to existing tuning rules.

Consider the fourth-order controlled processes proposed as benchmark in [8] and given by the transfer function:

$$P_\alpha(s) = \frac{1}{\prod_{n=0}^{3}(\alpha^n s + 1)}, \tag{4.6}$$

with $\alpha \in \{0.1, 1.0\}$.

Using the three-point identification procedure *123c* [1] FOPDT models were obtained, whose parameters are listed in Table 4.1.

TABLE 4.1: Example - $P_\alpha(s)$ FOPDT models

α	K_p	T	L	τ_o
0.10	1	1.003	0.112	0.112
1.0	1	2.343	1.860	0.794

For comparison purposes the following PI tuning methods that in some extend considered the control system robustness into the design procedure were selected: the *Model-Reference Robust Tuning* (MoReRT) [2] that uses a model matching

approach for smooth time responses and, at the same time, ensures a pre-specified, level of robustness $M_s = \{1.4, 1.6, 1.8, 2.0\}$; the *Kappa-Tau* (K-T) [6] that uses an empirical closed-loop dominant pole design of 2DoF PI controllers for a batch of controlled processes and provides tuning relations for robustness levels of $M_s = 2.0$ and $M_s = 1.4$; the *Simple Internal Model Control* (SIMC) [105] that is an IMC-based tuning for 1DoF PI controllers to obtain a good trade-off between speed of response, disturbance rejection, robustness ($M_s \approx 1.59$), and control effort requirements; the *Approximated MIGO* (AMIGO) [41] which is based on the loop shaping MIGO method [9] that maximizes the controller integral gain for the minimization of the integrated error to a step load disturbance, subject to a robustness constraint ($M_s = 1.4$) for PI controllers, in particular the revised version of the AMIGO method in [7] will be used.

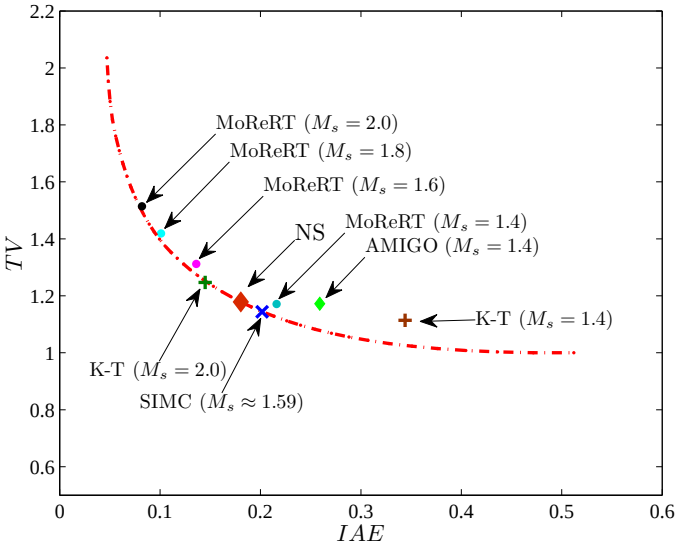


FIGURE 4.6: Pareto front for the process model corresponding to $\alpha = 0.1$.

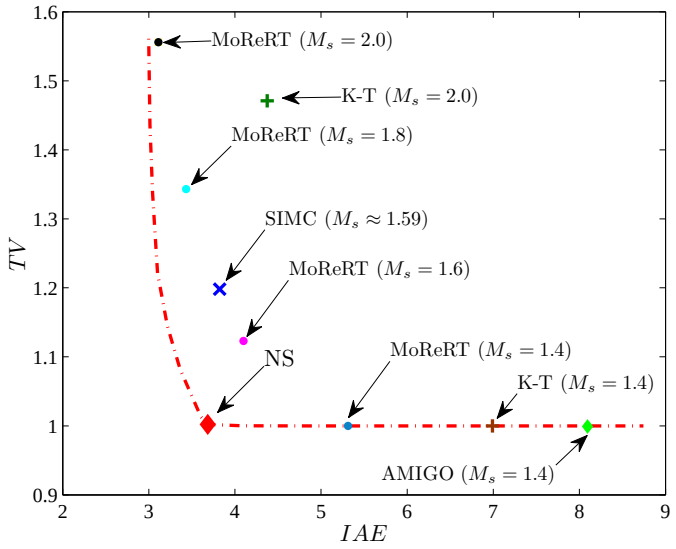


FIGURE 4.7: Pareto front for the process model corresponding to $\alpha = 1.0$.

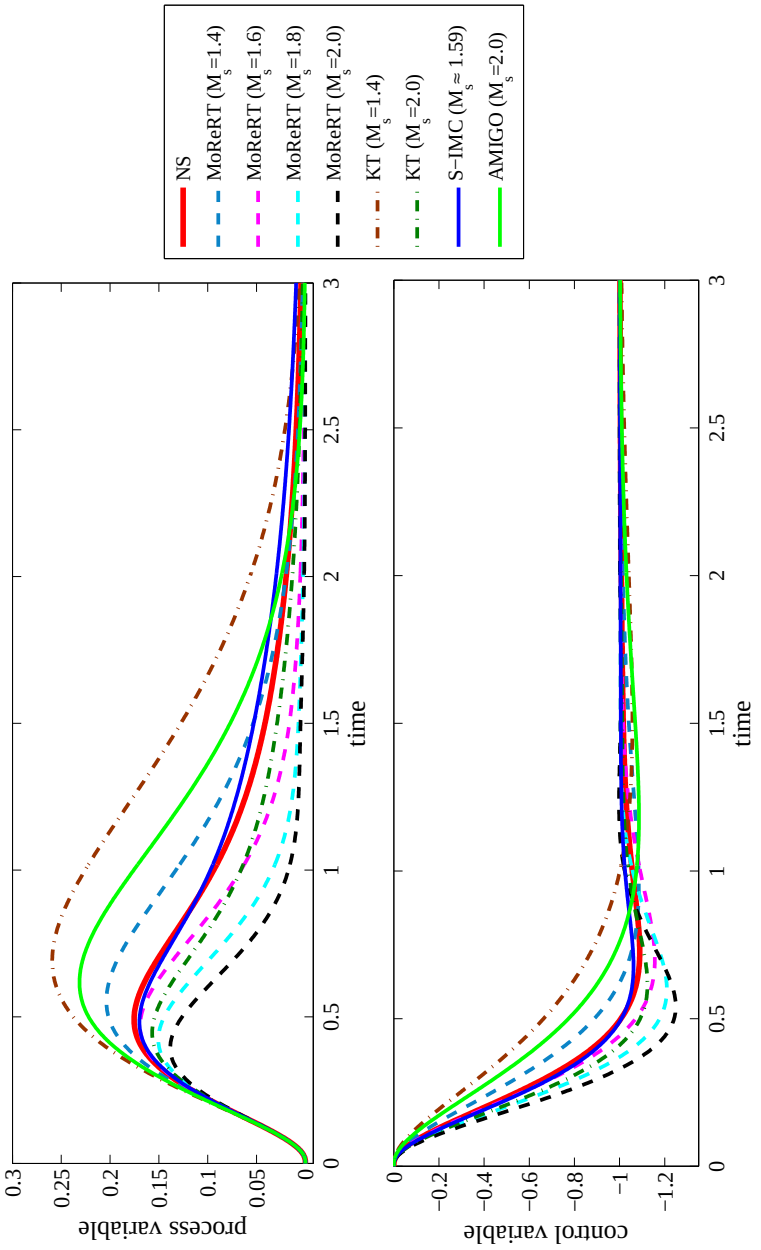


FIGURE 4.8: Load disturbance step responses for $P_{\alpha=0.1}(s)$.

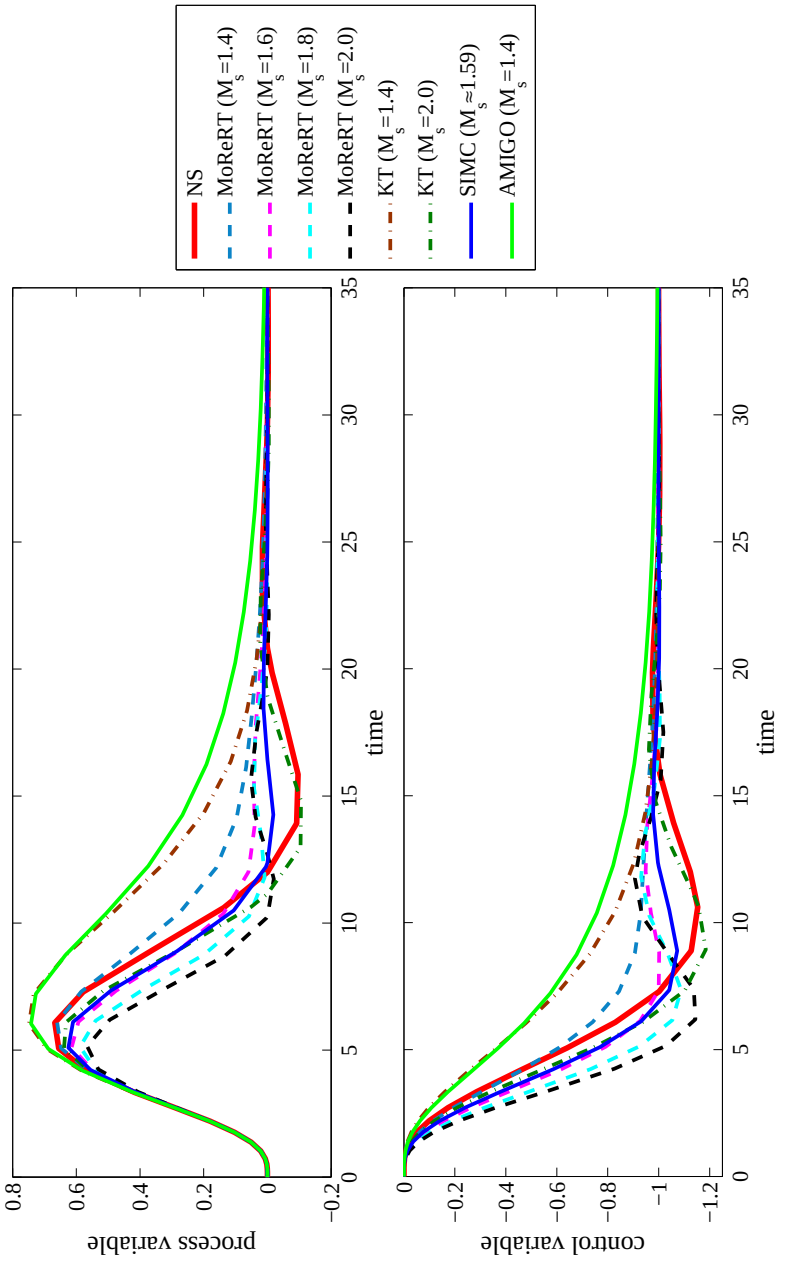


FIGURE 4.9: Load disturbance step responses for $P_{\alpha=1.0}(s)$.

The Pareto fronts corresponding to the process models corresponding to $\alpha = 0.1$ and $\alpha = 1.0$, are shown in Figures 4.6 and 4.7, respectively. These examples provide a graphical comparison of the performance/robustness trade-off among the tuning rules aforementioned and the Nash Solution (NS). The achieved time responses when facing a step load disturbance are shown in Figures 4.8 and 4.9. It can be stated that the controller tuning suggested by the NS choice improves the time responses in comparison with the other tuning rules.

Note that the *MoReRT*, *SIMC* and *K-T* ($M_s = 2.0$) tuning rules are almost located on the Pareto front, for the process model corresponding to $\alpha = 0.1$. However, for the second example $\alpha = 1.0$, it can be seen that some of the *MoReRT* and *K-T* ($M_s = 1.4$) tuning rules are located on the Pareto front. Nevertheless, as it can be observed the *NS* is located in the Pareto front and very close to *MoReRT* ($M_s = 1.6$) tuning rule, which means that the robustness level is $M_s \approx 1.6$ (remember we can assimilate TV to M_s).

4.4 Summary

In this chapter the multi-objective optimization for tuning a PI controller using the NNC method was implemented in order to give guidelines for decision making. The first analysis, was conducted to see the trade-off among the performance indexes as well as the fact of considering the robustness as a constraint in the formulation of the MOP.

However, it has even seen that control input usage has a high correlation with the robustness (measured in terms of M_s). Furthermore, the Pareto front approximation provides a graphical interpretation of the performance/robustness trade-off. It is seen that some of the existing tuning rules can effectively be improved both in terms of control input usage and performance (they are dominated solutions).

Therefore, for decision making, we propose the Nash Solution (NS) [11]; which lies on the Pareto front and is computed as the intersection between the surface and the diagonal line that passes through two opposite vertexes of the smallest cube that inscribes the surface. The NS provides an automatic selection and a direct approach for the choice of one point from the Pareto front approximation, this will generate a possibility for tuning a controller that it can generates better system outputs than existing tuning methods.

Chapter 5

Multi-stage Approach

In this chapter, a Multi-stage Approach for MOO process is presented in order to offer the designer the desired characteristics (as convergence, constrained, robust optimization, etc.) that could be related to the Pareto Set (solutions) required by the decision maker. On the other hand, we propose a MOOD procedure focused on reliability-based optimization instances, to evaluate quantitatively the performance degradation of the controller, due to unexpected or unmodeled dynamics.

5.1 Introduction

The increasing complexity of control systems and processes makes necessary using different measures to evaluate a controller's performance in order to be considered in the design process. One of the fundamental tools to face complexity of today's systems and processes is by approximating a model. Such model will be used (for example) by the control engineer, in order to 1) select the most convenient control structure and 2) tune such controller selection.

Being plant-model mismatch one of the main responsible for performance degradation, it becomes important not only to consider a robust design process, but also a reliability design process, see for example [107]. Reliability consist in evaluate, the performance degradation quantitatively given the expected model uncertainties. As the performance is usually stated in terms of the nominal model, the designer must introduce some measure of performance degradation when the model deviates

from its nominal one. This is the idea of reliability and constitutes and alternative way of looking at the robust performance specification: minimizing the deviation, in statistical terms, from the performance achieved for the nominal model when it is evaluated on a potential alternative plant.

On the basis of the presented scenarios, the main goal of this chapter is to present a Multi-stage approach for MOO with the purpose of dealing with reliability specifications. This reliability specification is introduced as a way of minimizing nominal performance degradation due to model uncertainty in controller design. The main advantage of the Multi-stage approach proposed here, compared to the single approaches, is the improvement of the convergence properties.

5.2 Reliability based design optimization

One of the fundamental tools to face complexity of today's systems and processes is by approximating a model. It is usual to face in this modeling stage with the simplicity *vs.* accuracy dilemma. As a consequence, discrepancies between the process and the model are expected. Additionally, the designer can face other uncertainties. According to [15] there are two types of uncertainties:

- Random or Aleatory Uncertainties: are of physical nature, where the variables within the system do have an inherent change.
- Epistemic Uncertainties: reflects the differences between the model used to describe the system and the real system.

Recently, ongoing research is being performed to introduce suitable techniques for reliability based design optimization (RBDO). These methods can be classified according to three major categories according to [52]: simulation methods, numerical integration and analytical methods. In this work, we will implement the first method, that are well known in the literature using sampling and estimation, being the most used the Monte Carlo Simulation (MSC) method [93]. This method is used quite often because its simplicity and robustness.

RBDO instances concern is to guarantee a given performance, considering the possibility (in the case of controller tuning) of unmodelled dynamics (epistemic

uncertainties). In this work we will implement the MCS method to evaluate the degradation of the performance when uncertainties on the system exist. In this case, several simulations will be carried out, increasing the computational burden of the cost function itself. This method computes an index by simulating the model and the random behavior of the system. Hereafter, comes the fact that running several simulations could affect the performance and their exploration capabilities and hence slow down the overall convergence of the algorithm. For this reason, the idea of using the Multi-stage approach for solving optimization problems including reliability will bring confidence, therefore an effective and robust design alternative for the system. As the generation of the Pareto front approximations for reliability problems involves an increase of the computational burdens, this convergence improvement is highly valued.

5.3 MOOD procedure using the Multi-Stage Approach

In what follows, we present a MOOD procedure, including the Multi-stage approach on the MOO process, to handle reliability-based optimization statements. In order to give an example, a PI controller is used.

5.3.1 MOP Definition including Reliability

From the previous concepts in Chapter 4, the most basic MOP statement for PI controller tuning including performance, robustness and reliability could be represented as:

$$\begin{aligned} \min \boldsymbol{J}(\boldsymbol{\theta_c}) &= [J_1(\boldsymbol{\theta_c}), J_2(\boldsymbol{\theta_c}), J_3(\boldsymbol{\theta_c})] \\ J_1(\boldsymbol{\theta_c}) &= Performance(\boldsymbol{\theta_c})\Big|_{P(s)} \\ J_2(\boldsymbol{\theta_c}) &= Robustness(\boldsymbol{\theta_c})\Big|_{P(s)} \\ J_3(\boldsymbol{\theta_c}) &= \sigma\left(J_1(\boldsymbol{\theta_c})\Big|_{P'(s)}\right) \end{aligned} \tag{5.1}$$

where,

$$\boldsymbol{\theta_c} = [K_p, T_i] \tag{5.2}$$

are the parameters of the PI controller, $P(s)$ is the nominal model used as base case. According to the control loop of Figure 2.1 common choices in controller tuning for the goals objectives are: the IAE, TV (we associate the robustness level with the goal of achieving low input usage). Some previous results such as the ones in [31] provide the findings that the Ms-value correlates well with the input usage as given by its TV).

Furthermore, a new goal is introduced:

- Reliability based objective $\left[J_3(\boldsymbol{\theta_c}) := \sigma\left(J_1(\boldsymbol{\theta_c})\Big|_{P'(s)}\right)\right]$: the nominal controller design will be stated in terms of a nominal process model $P(s)$. In order to take into account plant model mismatch because of uncertainty, a family $\mathcal{F}$, of models can also be considered such that the nominal performance is maintained within such family. Stated in other words, what it is desired is the achieved nominal performance to degrade as less as possible as we face one of the possible plant family models $P'(s) \in \mathcal{F}$. The goal will be to minimize the deviation from the achieved nominal performance, given by the value of $J_1(\boldsymbol{\theta_c})$. Here σ sands for the standard deviation.This deviation will be estimated by using the Montecarlo sampling approach. It provides not only the performance values also evaluated. This objectives brings something meaningful about performance degradation, something practical

Hence, the optimization procedure considers to include the Robust Performance due to the presence of uncertainties on the parameters or during the design stage. The purpose of including $J_3(\boldsymbol{\theta_c})$ is to improve the result with respect to the optimal criteria, the algorithm must find a way to reduce the deviation in order to have a robust design [91].

The idea of incorporating reliability as a design objective relies on the fact that it introduces an additional perspective to the controller tuning problem. That is, besides the information that a robustness measure gives to the control engineer, the reliability measure gives information about expected performance degradation. As noticed in [107], a stochastic evaluation of a controller proposal is useful when discrepancies between process and model are expected. That is, it is possible to

measure quantitatively the expected degradation on performance. Therefore, this provides (for example) a measure on "failure risk" in fulfilling control requirements.

5.3.2 Multi-stage Approach

Given the MOP definition (5.1), some desirable characteristics for the optimizer will be required. Hybridization techniques have been generally accepted as a strategy promoting diversity throughout the search for the global optimum; such techniques seek to combine the advantages of different algorithms whilst, at the same time, trying to minimize any disadvantage they could have if they work separately [115]. This kind of techniques are quite popular methodologies that the designer implements in order to improve the convergence rate and accuracy of optimization algorithms. The hybridization technique is used for an unified purpose, where all the algorithms are implemented to solve the same problem and can be grouped in two categories: collaborative and integrative hybrids. In this work a collaborative hybrids is implemented, which involves the combination of two or more algorithms. Moreover, according to the way that the algorithms run they can be classified in three structures: multi-stage, sequential and parallel. Interested readers may refer to [115] for more details about the taxonomy of the hybrid algorithms.

In this chapter a Multi-stage approach is presented. This approach involves two stages; first, one algorithm will perform the local search and the second algorithm acts as a global searcher. The framework that represents the Multi-stage approach is illustrated in Figure 5.1.

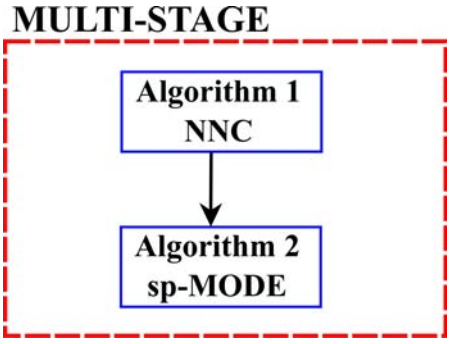


FIGURE 5.1: Multi-stage approach structure.

On the basis of these two algorithms the Multi-stage algorithm is to be defined; the NNC algorithm that will be searching for the convergence area (local search) and

the sp-MODE algorithm it is in charge of the global search. The NNC algorithm will find the first Pareto front approximation; the sp-MODE algorithm will use, as initial population, the locally Pareto-optimal approximation calculated by the NNC algorithm in the 3-objective optimization instance. This will reduce the numerical burden associated to the Montecarlo approach and would improve its exploitation capabilities (local and dominated solution are avoided from the Pareto front approximation).

Therefore, in order to generate the Pareto front approximation using the Multi-stage approach, two algorithms[1] are merged:

1. Deterministic Optimization Approach: the Normalized Normal Constraint (NNC) algorithm proposed in [61] and already presented in Chapter 3. The main advantage of this algorithm rely on their local convergence capabilities as well as their robustness to provide a good approximation of the Pareto front. However, they are highly sensitive to the initial conditions required to run the optimization and some issue regarding computational burden. Even if the objectives have been stated to guarantee convexity properties the constraints incorporated to achieve a good spread over the Pareto front approximation may modify the objective space.

2. Evolutionary Optimization Approach: the Multiobjective Differential Evolution Algorithm with Spherical Pruning (sp-MODE) algorithm is based on Differential Evolution [25], which uses a spherical grid in order to keep the diversity in the approximated Pareto front, see Figure 5.2b. The basic idea of the spherical pruning is to analyze the proposed solutions in the current Pareto front approximation by using normalized spherical coordinates from a reference solution. With such approach, it is possible to attain a good distribution of the Pareto front [87]. The algorithm selects one solution for each spherical sector, according to a given norm or measure. Such algorithm has been used with success for controller tuning purposes [84]. Evolutionary algorithms have shown interesting properties to handle highly constrained and non-linear objective functions due to their flexibility and adaptability [88, 114]; however two potential drawbacks are also known: (1) given their

[1]Both algorithms are available in Matlab Central at
http://www.mathworks.com/matlabcentral/fileexchange/38976 and
http://www.mathworks.com/matlabcentral/fileexchange/39215

stochastic nature, their convergence can not be guaranteed and (2) the tuning of their own parameters could be a time consuming task and, if selected inappropriately, their performance could be deteriorated.

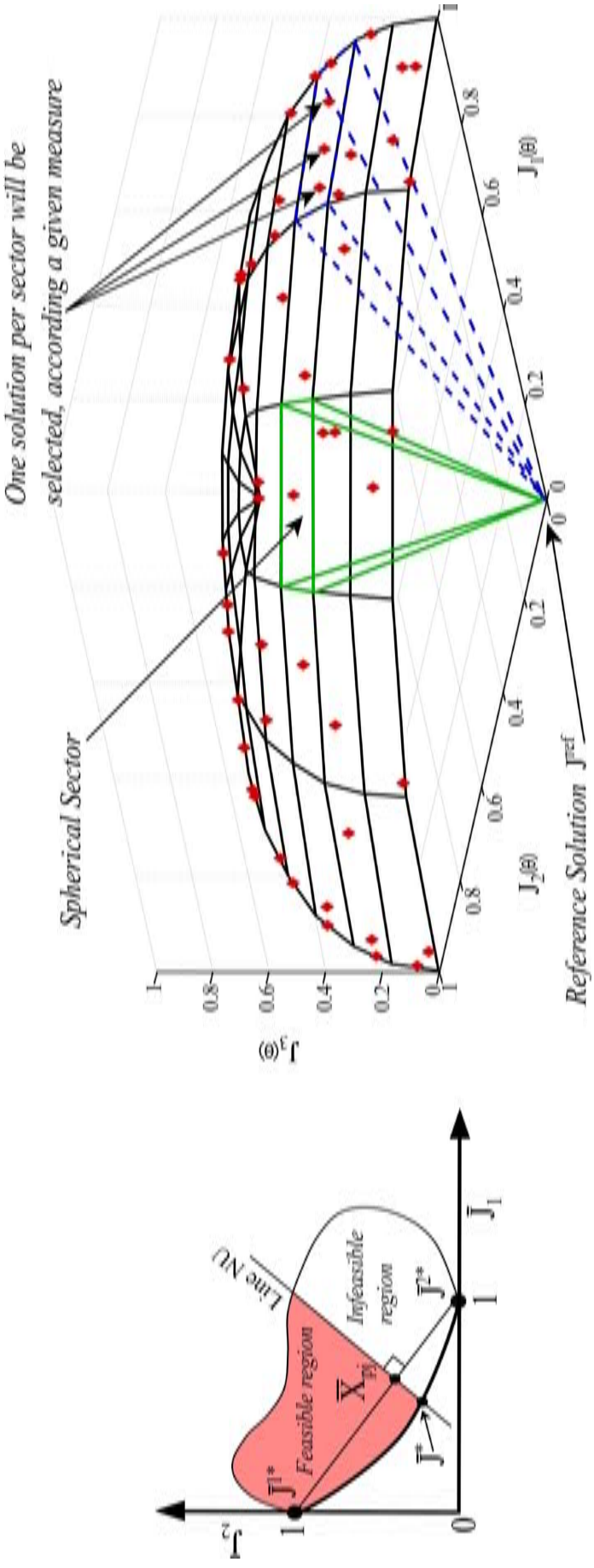


FIGURE 5.2: Algorithms used for the accomplishment of the Multi-stage Approach. In the left side: the NNC algorithm in the bi-objective case. In the right side: sp-MODE algorithm.

Algorithm II: *Multi-stage approach*
1 : Determine the initial solution for the NNC algorithm.
2 : Compute the anchor points for objectives $J_1(\theta_c)$ and $J_2(\theta_c)$.
3 : Approximate the sub-Pareto front $J'(\theta_c) = [J_1(\theta_c), J_2(\theta_c)]$ with NNC algorithm and the computed anchor points.
4 : Use the Pareto set approximation $\Theta_p'^*$, computed previously as initial population for sp-MODE algorithm.
5 : Approximate the Pareto front $J(\theta_c) = [J_1(\theta_c), J_2(\theta_c)]$.
6 : Perform decision making step for controller selection.

FIGURE 5.3: Multi-stage approach outline.

The multi-stage approach (Algorithm II, see Figure 5.3) allows both algorithms to complement each other in despite of their drawbacks and improve the results of the overall optimization in terms of convergence and accuracy, when a MOP described by Equation (5.1) is being solved.

5.3.3 MCDM stage

In order to select a solution from the Pareto front approximated, two design alternative are chosen from the Pareto front approximation:

- (i) Nash Solution - (NS): provides the Pareto front solution that dominates the larger number of solutions from the Pareto front (being absolutely better than any one of the solutions). This option has been introduced in Chapter 4.

- (ii) Level Diagrams - (LD): as a second desing alternative we are using level diagrams[2]. They have been used in [17, 86] as they represent a trade-off as commented on [82] among the properties for visualization stated by [53]: simplicity, persistence and completeness. LD consists of representing each objective and design parameter on separate diagrams. This approach is based on the classification of Pareto front points according to their proximity to ideal points measured with a specific norm of normalized objectives

[2]Tool available for Matlab at http://www.mathworks.es/matlabcentral/fileexchange/24042

(several norms can be used); in each diagram (objective and design parameter), solutions are synchronized in the vertical axis, which represents the aforementioned norm.

The first option leads to a (semi)-automatic selection, while the latter leads to an active seeking. Next, to show the applicability of this procedure, two different instances will be presented in the following Chapters.

5.4 Summary

A Multi-stage approach for controller tuning based on multi-objective optimization to improve reliability was presented. Such approach has considered the inclusion of reliability as a design objective into the MOP statement. In order to deal with Robust Performance by minimizing nominal performance degradation when a family of potential plant models is considered. This degradation is quantified in a stochastic way by means of MCS method.

This approach will improve its exploitation capabilities of the algorithms and it will reduce the numerical burden associated with the generation of the Pareto front approximations for reliability problems. It is worth stressing, that if a solution gives the desirable trade-off, then it is possible to optimize such solution, looking for a locally Pareto optimal solution which dominated the preferable solution. In this way, the preferability (from the point of view of the trade-off among conflicting objectives) and convergence are fulfilled.

Chapter 6

Two case studies

Chapter 5 presented the Multi-stage approach involving reliability based optimization for controller tuning. This chapter presents two case studies to validate such approach: the first one is the benchmark described in [67], which proposes a Boiler control problem for controller tuning and as second case, the approach will be evaluated on a laboratory experiment build up on a non-linear Peltier Cell. The obtained results validate the usefulness of the Multi-stage approach as a tool for solving multi-objective optimization problems with reliability constraints for controller tuning applications.

6.1 The Boiler Control Benchmark (IFAC-2012)

The first process under consideration is the *Control Benchmark 2012* presented in [67]. It proposes a Boiler control problem [29, 66] based on the work of [79]. This work improves the model provided in [13] by adding a non-linear combustion equation with a first order lag to model the oxygen excess in the stack and the stoichiometries *air-to-fuel* ratio for complete combustion. The aim of this first case study is twofold. First, to show how the designed PI controller faces the control problem with plant-model mismatches. Secondly, to validate the hypothesis that the proposed multi-stage approach leads to better convergence properties for the MOP describe in Equation (5.1) than each one of the single approaches alone.

6.1.1 Process description

The control of boilers has been an important problem for a long time. It is well known that the correct control of this plant can improve the performance with its corresponding economical savings and safety improvement. But it is not a simple task since non-minimum phase characteristics as well as not self-regulating behaviour is commonly found.

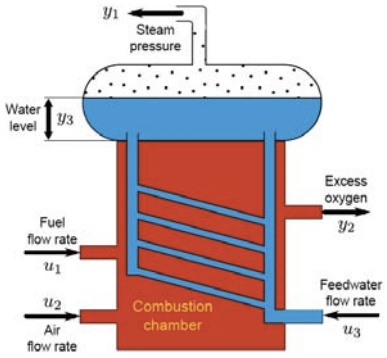


FIGURE 6.1: Boiler plant layout.

In the boiler area, many models exist ranging from complex knowledge based models to experimental models derived from special plant tests. A boiler benchmark was proposed in [67] on the basis of the control oriented model proposed by [79]. This model predicts the process response in terms of measurable outputs (drum pressure, drum water level, and excess oxygen in fuel gas) to the major manipulated inputs (air/fuel flow rates, feedwater flow rate) as well as the effect of disturbances (changed steam demand, sensor noise), model uncertainty (e.g., fuel calorific value variations, heat transfer coefficient variations, distributed dynamics of the steam generation), and constraints (actuator constraints, unidirectional flow rates, drum flooding). A diagram can be found in Figure 6.1. The heat from the combustion in the chamber is transferred to the water, producing the water vapor. The residual gases from the combustion are vented to the atmosphere.

Therefore, as a case study, the boiler benchmark is considered to be the plant to control. This benchmark is used to evaluate and compare different kinds of tuning/control techniques [36, 58, 90, 104, 124]. This benchmark [1] considers two versions of the problem. In this work, the reduced SISO version is implemented to evaluate our proposal. The SISO version, see Figure 6.2 is well suited for PI/PID

[1]The MATLAB file can be obtained at http://www.dia.uned.es/ fmorilla/benchmark09_10/

controller design, where the fuel flow is manipulated to control the steam pressure, is considered.

The identified nominal model[2] $P(s)$ to be used is:

$$G(s) = \frac{0.4}{1 + 45.7s} e^{-3.42s} \tag{6.1}$$

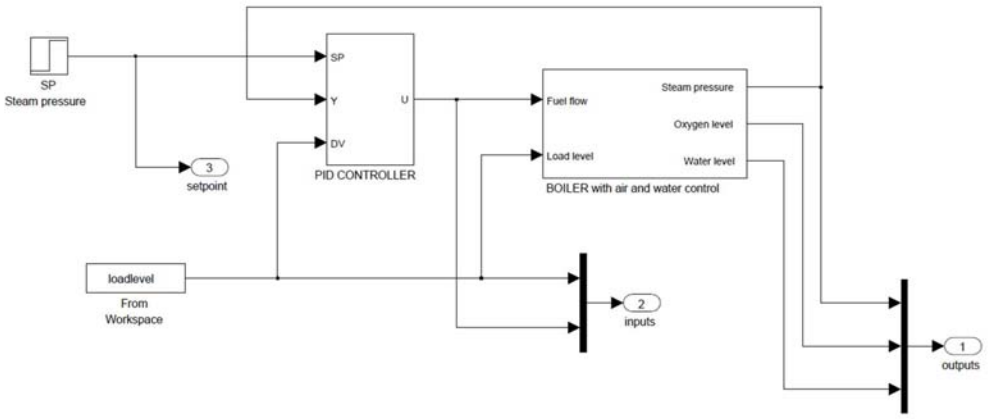


FIGURE 6.2: SISO loop for Boiler benchmark (taken from [67]).

6.1.2 Design problem statement

The designer is interested in measuring the performance on the nominal model, but also measuring the performance degradation due to the expected differences in lags and time constant. Therefore, the MOP with reliability statements as it has been stated in the previous chapter is considered.

For the reliability objective, the behavior of the controller for a potential plant $P'(s)$ that belongs to the model family is considered on the following way. The controller is evaluated in a random uniformly sampled model $P'(s)$ in the intervals $L = 3.42 \pm 20\%$ and $T = 45.7 \pm 20\%$. For this example, 20 models were used. This benchmark also stated a performance index (I_b) based on a reference case controller, $C_{ref}(s)$, see [67]. In this test, it will be stated that the $C_{ref}(s)$ will define the pertinency of the Pareto front, *i.e.* the region of interest for the designer. This is done to avoid sluggish controllers in the Pareto front approximation, since for a

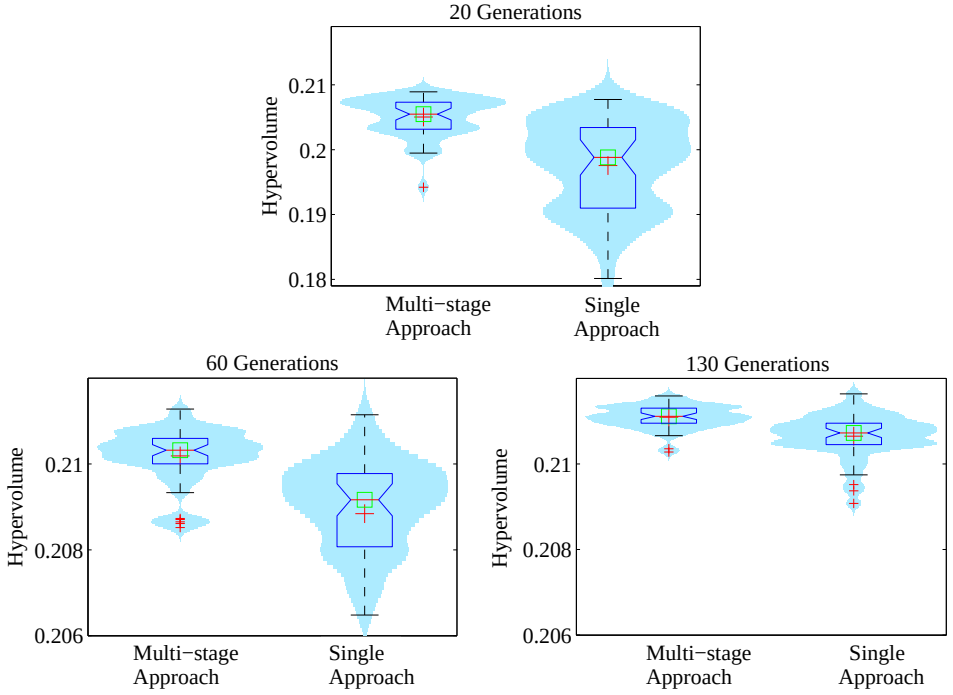
[2]This model was obtained with a step response experiment using the standard identification toolbox from Matlab© with an standard step response

controller, the best way to achieve the minimal $J_2(\boldsymbol{\theta_c})$ is assigning $K_p = 0$. This is the trivial way of achieving the minimal input usage $TV = 0$ (no response of the controller to the measured error).

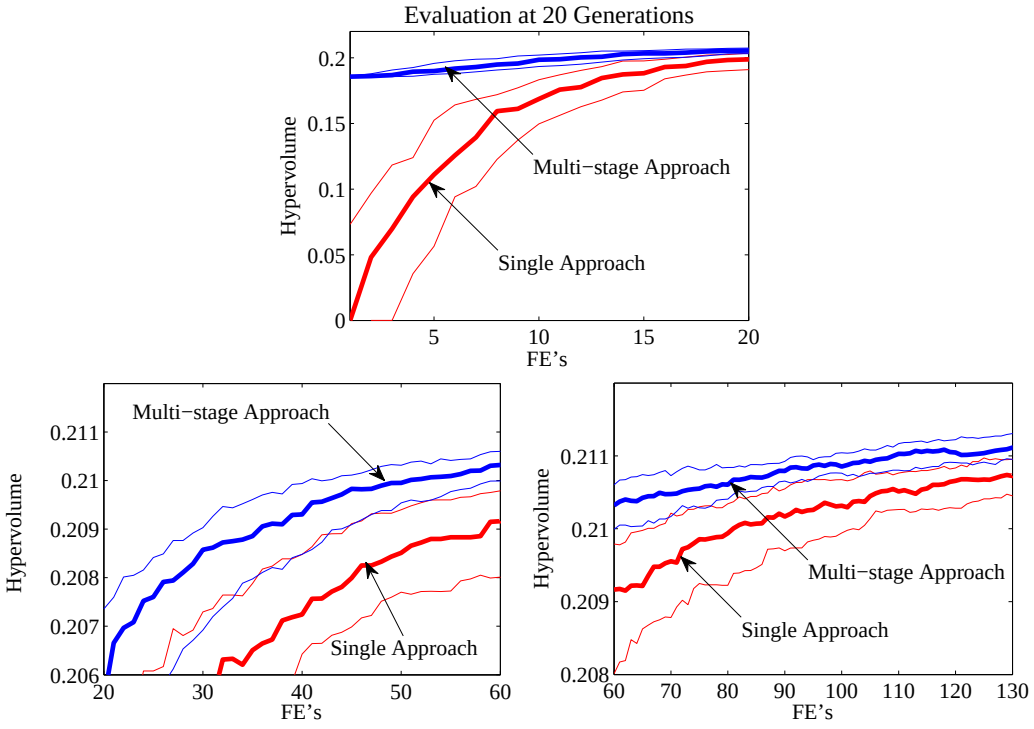
6.1.3 Results and discussions

The first step in the application of the multi-stage approach is the solution of the MOP but without considering the reliability statement (therefore solving a 2-objective problem). This way, a first approximation to the Pareto set will be generated. This first approximation is faster to be obtained and is based on the application of the NNC algorithm. In this case, we will use as Initial Solution (IS), the $C_{ref}(s)$ already provided in the Benchmark definition, therefore, to obtain the Pareto front approximation by the NNC algorithm we will use as $IS = [2.5, 50]$. Once we have the first approximation of the Pareto set generated, the second approach (sp-MODE) for 3-objective MOP will be implemented, using the results from the NNC solution (2-objective problem) as the initial population.

In order to test the usefulness of the Multi-stage approach in the optimization process, the normalized hypervolume indicator proposed by [129] will be used. This measure allows to compare the performance of the Muti-stage approach versus the Single approach (sp-MODE) by computing and tracking the hypervolume through the algorithm execution. With this information it is possible to deploy the convergence plot of the optimization process, in the same way that for single objective optimization the minimum value achieved is tracked. In order to calculate the hypervolume for the first approach (NNC) with 3-objective, 2842 function evaluation were employed and achieved an hypervolume of 0.1856.



(a) Box-plots and violins plots as hypervolume indicator.



(b) Convergence path of hypervolume indicator.

FIGURE 6.3: Hypervolume indicator implemented on the Boiler Control Benchmark. Comparison among the Multi-stage and Single approach (sp-MODE). For such purpose, a total of 20, 60 and 130 generations and 51 independent runs are carried out.

Hereafter, to evaluate the hypervolume for the Single approach (sp-MODE) and the multi-stage approach it will be used the number of the functions evaluations as budget, implemented for the first approach to accomplish the optimization. For such purpose, 51 independent runs are carried out.

Numerical results are shown in Table 1, where the corresponding values of the minimum, median, maximum, mean and variance of the attained hypervolumes are provided. The Statistical significance was validated with the Wilcoxon test at 95% [28]. As it can be noticed, the Multi-stage approach achieved higher values of the median which means that it provides larger dominated regions (better performance).

TABLE 6.1: Hypervolume indicator achieved for the Multi-stage and Single approach (sp-MODE). A total of 20, 60 and 130 generations (20 function evaluations per generation).

	20 generations		60 generations		130 generations	
	Multi-stage	Single	Multi-stage	Single	Multi-stage	Single
minimum	**0.1942**	0.1801	**0.2085**	0.2030	**0.2103**	0.2091
median	**0.2055**	0.1988	**0.2103**	0.2092	**0.2111**	0.2107
maximum	**0.2089**	0.2077	**0.2113**	0.2111	**0.2116**	0.2116
mean	**0.2055**	0.1975	**0.2112**	0.2088	**0.2111**	0.2106
variance	**0.09e-4**	0.47e-4	**0.04e-5**	0.20e-5	**0.043-6**	0.25e-6

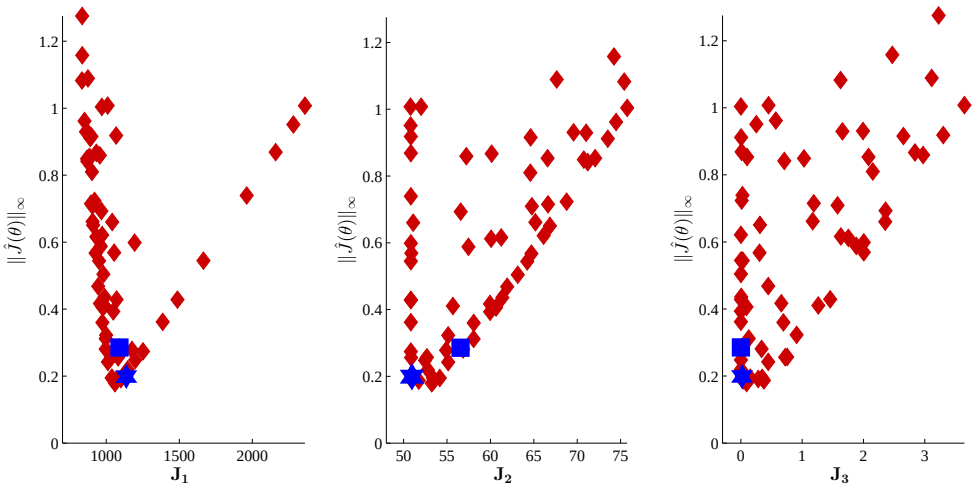
Additionally, in Figure 6.3a the distribution plots of hypervolumes achieved (Table 1) are represented graphically using the box-plots and violin-plots, where a total of 20, 60 and 130 generations (20 function evaluations per generation). The upper and lower ends of the box, represents the upper and lower quartiles; while, the thick line between second and third quartile is the median. The dashed lines summarize the spread and shape of the distribution. Further, the converge path of hypervolume is despicted in Figure 6.3b.

The resulting Pareto front and Pareto set approximations generated by the Multi-stage approach are shown in Figures 6.4a,b. Figure 6.4a shows the projection of the Pareto front along each one of the three objective functions. The values for the overall objective function $\|\boldsymbol{J}(\boldsymbol{\theta_c})\|$ with

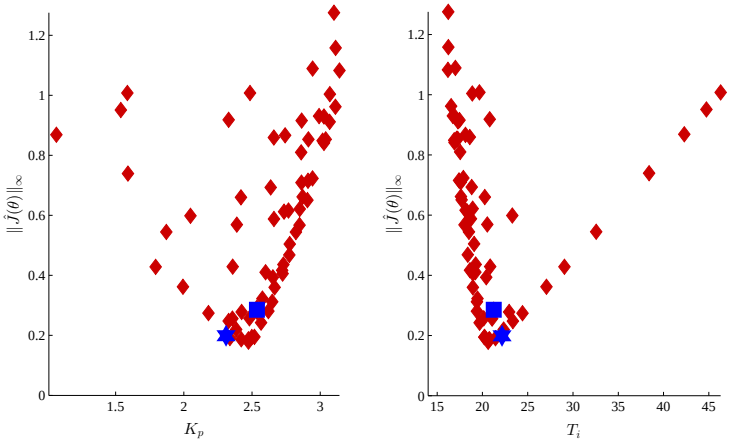
$$\boldsymbol{J}(\boldsymbol{\theta_c}) = [J_1(\boldsymbol{\theta_c}), J_2(\boldsymbol{\theta_c}), J_3(\boldsymbol{\theta_c})] \tag{6.2}$$

are shown along the x-axis values for each one of the $\boldsymbol{J}(\boldsymbol{\theta_c})$. Two solutions were selected from the Pareto front: Nash-Solution $\boldsymbol{\theta_c} = [2.31, 22.16]$ and a solution selected by analyzing the Pareto front using Level Diagrams: $\boldsymbol{\theta_c} = [2.54, 21.25]$ (NS and LD respectively). These solutions are also displayed in the Pareto set approximation of Figure 6.4b. Here the values of $\|\boldsymbol{J}(\boldsymbol{\theta_c})\|$ are projected with respect each one of the parameters to be optimized (K_p and T_i).

Therefore, it is seen that the Multi-stage approach can effectively generate the three-dimensional Pareto front approximation with better convergence properties than the evolutionary algorithm applied as a Single approach (sp-MODE) and the NNC algorithm by itself.



(a) Pareto Front approximations.



(b) Pareto Set approximations.

FIGURE 6.4: Pareto front and Pareto set approximation for the Boiler Control Benchmark. The controllers selected with the Nash Solution (blue star) and Level Diagrams (blue square) arerepresented.

6.1.4 Experimental setup

In order to implement both design alternatives (NS and LD) in the Boiler benchmark (that is, the original non-linear model proposed in the benchmark), the performance index defined by the benchmark (I_b) is calculated to evaluate the performance for a controller C_s. It is an aggregate objective function which combines the mean value of five individual indexes (IAE, ITAE[3] and TV[4]) using a

[3]ITAE: Integral of the time-weighted absolute error value.
[4]TV: Total variation of the control action. Also known as IADU.

weighting factor ($w = 0.25$). This indexes are referred to a C_{ref}. More details can be consulted in [67].

In order to compare with the original benchmark two different tests are proposed:

- Test type 1: performance when the system had to attend a time variant load level.
- Test type 2: performance when the system had to attend a sudden change in the steam pressure set-point.

It is worth noting that the experiments carried out, as they are described in the benchmark, include noise in the signal input, this data is used for the numerical analysis.

In this work, both NS and LD are presented as criteria for generating the solution to a MOP. The differences between the two alternative design come from the way the solution is selected from the Pareto front approximation. Both criteria are optimal (choosing one, depend on the preferences of the designer).

- NS: this criteria gives the same importance to the conflicting objectives and it is automatically calculated from the Pareto Front, .
- LD: if the designer wants to do an analysis of the Pareto front or have preferences of one or other objective, the LD allows he/her this possibility.

Figure 6.5 shows a comparative of the controllers selected through the NS and LD for the Test type 1. The time responses of both design alternatives are better than the response offered by the reference controller according to the index stated for the benchmark *competition*. Furthermore, it can be noticed that the steam pressure shows minor deviations from its set-point as a consequence of having increased activity in the fuel flow. However the water level shows similar deviations and the oxygen level remains controlled by the fuel/ air ratio, affected only by the noise. For the Test type 2 see Figure 6.6 where it can be noticed the variations in the steam pressure response has almost disappeared.

As it can be noticed both design alternatives bring an effective controller that fulfills all the requirements, with a better performance than the reference controller. For the Test type 1 and 2, the reference controller achieves a $I_b = 1.00$ and for

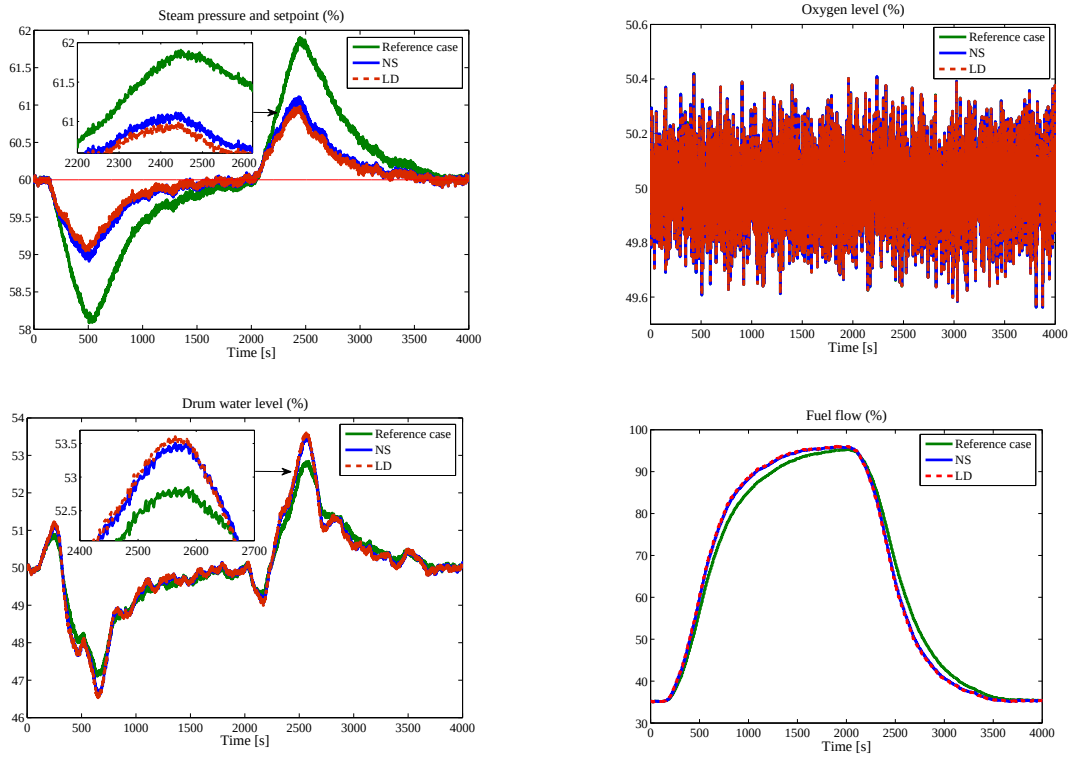


FIGURE 6.5: Test type 1: Performance of the PI controller selected with the NS $[K_p,\ Ti]$=[2.31, 22.16] ($I_b = 0.86$), LD $[K_p,\ Ti]$=[2.54, 21.25] ($I_b = 0.85$) and its comparison with the reference case $[K_p,\ Ti]$=[2.5, 50] ($I_b = 1.00$) in the benchmark setup.

TABLE 6.2: Performance indexes for the benchmark setup of the selected design alternatives.

Controller	I_b (Test 1)	I_b (Test 2)
C_{ref}	1.00	1.00
NS	0.86	0.83
LD	0.85	0.82

the two Pareto Optimal selected controllers these values are slightly above 0.8 (see Table 6.2). Furthermore, the simulation response of the design alternatives are despited in Figures 6.5 and 6.6.

Therefore, the proposed Multi-stage Approach brings a suitable framework that includes the reliability based on MOOD procedure. The approach is effective not only regarding the obtained PI controllers that do have a better performance than the C_{ref} (is less than the unit) but also achieves better values of the hypervolume than the Single approach (sp-MODE) and offers a better convergence properties for this kind of instances.

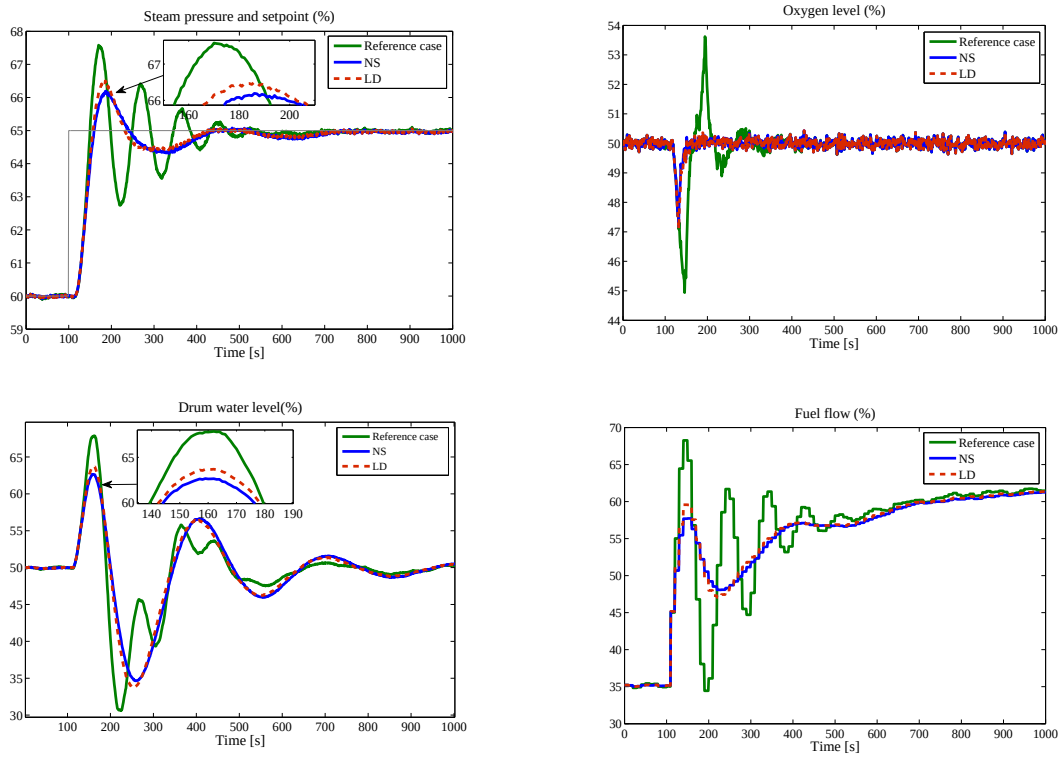


FIGURE 6.6: Test type 2: Performance of the PI controller selected with the NS $[K_p,\ Ti]$=[2.31, 22.16] ($I_b = 0.83$), LD $[K_p,\ Ti]$=[2.54, 21.25] ($I_b = 0.82$) and its comparison with the reference case $[K_p,\ Ti]$=[2.5, 50] ($I_b = 1.00$) in the benchmark setup.

6.2 Application to a Peltier Cell temperature control

The second case of study, is a Peltier Cell (thermoelectric process). The aim of this test is to validate in a *real* process the Multi-stage approach. The laboratory setup is shown in Figure 6.7.

6.2.1 Process description

The device under consideration is based on the Peltier effect, it is a heat pump where the manipulated variable is the current [%] whilst the controlled variable is the temperature [$^{\circ}C$] of the cold-face [55]. This kind of processes has non-linear dynamics as it will be shown. This effect was discovered by the physicist Jean Charles Athanase Peltier, in 1834.

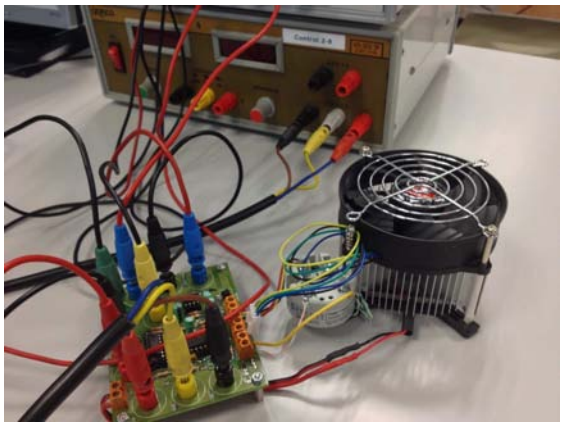


FIGURE 6.7: SISO loop for Boiler benchmark (taken from [67]).

6.2.2 Design problem statement

The designer is interesting in tuning a controller to mainly work around $T_e = -5°C$, but capable of achieving a good behavior in the range $T_e = [-7.5°C, 7.5°C]$. Therefore, successive step reference changes have been made in the T_e range in order to identify FOPDT models. The nominal model is described by,

$$P(s) = \frac{0.64}{2.74s + 1} e^{-0.2s} \tag{6.3}$$

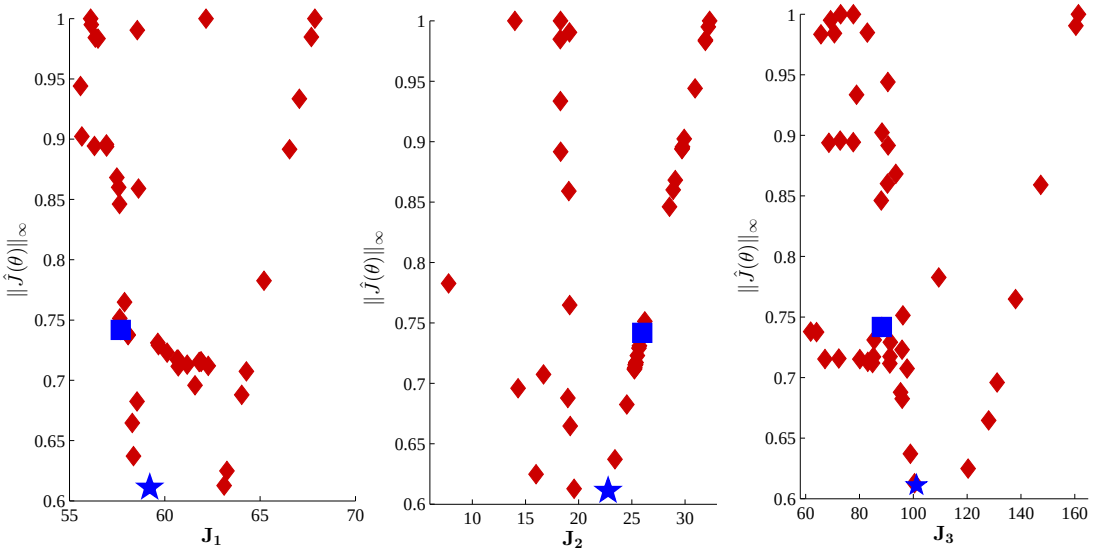
In this test, the MOOD procedure for PI controller design described before is implemented. The reliability objective the controller will be evaluated in a random uniformly sampled model $P'(s)$ in the intervals $K = 0.64 \pm 50\%$ and $T = 2.74 \pm 30\%$. For this example, 51 models were used.

6.2.3 Results and discussions

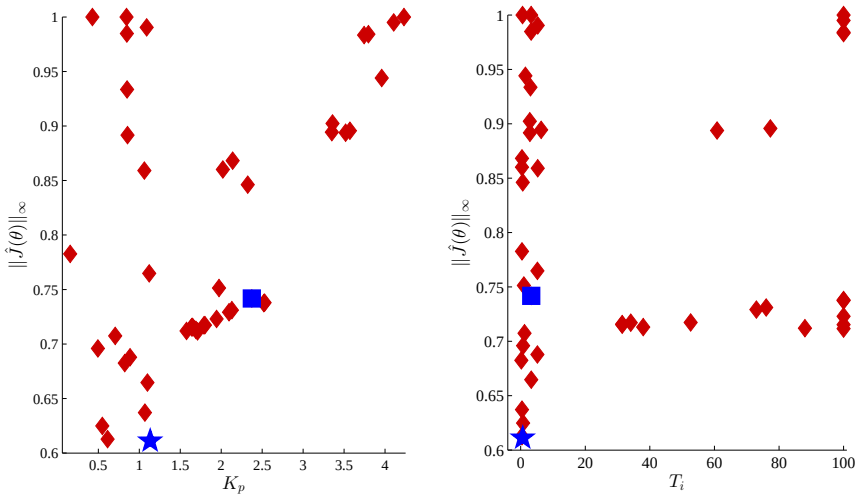
As commented before, the NNC algorithm is focused on objectives $J_1(\boldsymbol{\theta_c})$ and $J_2(\boldsymbol{\theta_c})$. The preliminary bi-objective Pareto front corresponding to these two objectives (therefore without taking into account the reliability objective) is approximated with the NNC algorithm (11,000 function evaluations approximately). The initial solution $IS = [0.19, 2.74]$ employed in the optimization process was calculated with the S-IMC tuning rule proposed by [106].

In the second step, the sp-MODE algorithm will use the Pareto front approximation provided by the NNC algorithm as an initial population. In order to avoid

controllers with high degradation on the third objective, the maximum value $\overline{J_3}$ for $J_3(\theta_c)$ has been bounded to $\overline{J_3} = J_3(0.19, 2.74)$. As with the previous example, and as commented when describing the MCDM stage, two final solutions are considered, the one that corresponds to the Nash solution (NS) and to the selection by using Level diagrams (LD).



(a) Pareto Front approximations.



(b) Pareto Set approximations.

FIGURE 6.8: Pareto front and Pareto set approximation for the Peltier Cell. The controllers selected with the Nash Solution (blue star) and Level Diagrams (blue square) arerepresented.

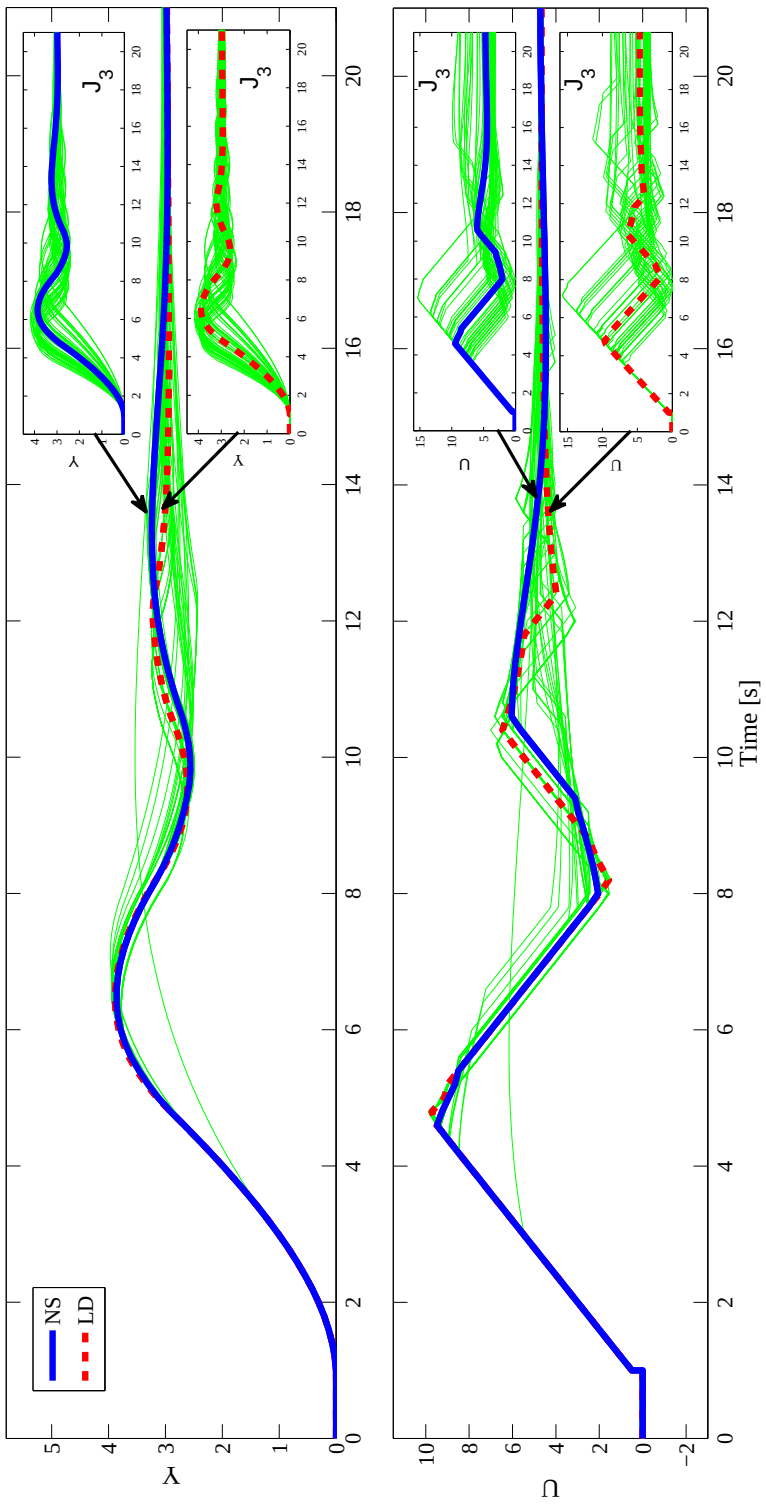


FIGURE 6.9: Temporal Responses. Performance of the selected controllers to face the stochastic set of plants that were use to calculate the 3 objective. Blue line: solution selected with Nash Solution for 3-objective; Red line: solution selected with Level Diagrams.

The obtained approximation as well as the final selected solutions are depicted in Figures 6.8a,b. Here we can see the Pareto set approximation on the parameter space (see Figure 6.8a). As the parameter space is bidimensional two plots are shown here corresponding to the projection of the overall objective function $\|\boldsymbol{J}(\boldsymbol{\theta_c})\|$ with

$$\boldsymbol{J}(\boldsymbol{\theta_c}) = [J_1(\boldsymbol{\theta_c}), J_2(\boldsymbol{\theta_c}), J_3(\boldsymbol{\theta_c})] \tag{6.4}$$

with respect to each one of the parameters (K_c and T_i). Same considerations apply to Figure 6.8b, where the projections of the three-dimensional Pareto front approximation with respect to each one of the objective functions that compose $\|\boldsymbol{J}(\theta_c)\|$ is shown. In both figures,the location of the selected solutions is shown: Nash-Solution $\boldsymbol{\theta_c} = [0.51, 0.41]$ and the solution selected by analyzing the Pareto front using Level Diagrams: $\boldsymbol{\theta_c} = [2.37, 3.26]$ (NS and LD respectively). Performance for both solutions in the time domain with respect to a sample of the plant family models is depicted in Figure 6.9. We can see how in both cases the selected controllers face the stochastic set of plants.

TABLE 6.3: Controller parameters for the Peltier Cell process.

Controller	K_p	T_i	IAE	TV
IS	0.19	2.74	992.4	86.0
NS	0.51	0.41	154.6	263.9
LD	2.37	3.26	120.7	657.4

Both solutions have also been implemented and evaluated in the laboratory setup. The time responses are shown in Figure 6.10 and in Table 6.3 shows the parameters of each alternative. Differences on the performance are due to the additional information used in the MOP statement minding degradation on IAE performance.

6.3 Summary

Overall, we have presented a MOOD procedure using a Multi-stage approach for the MOO process and including reliability-based optimization design for controller

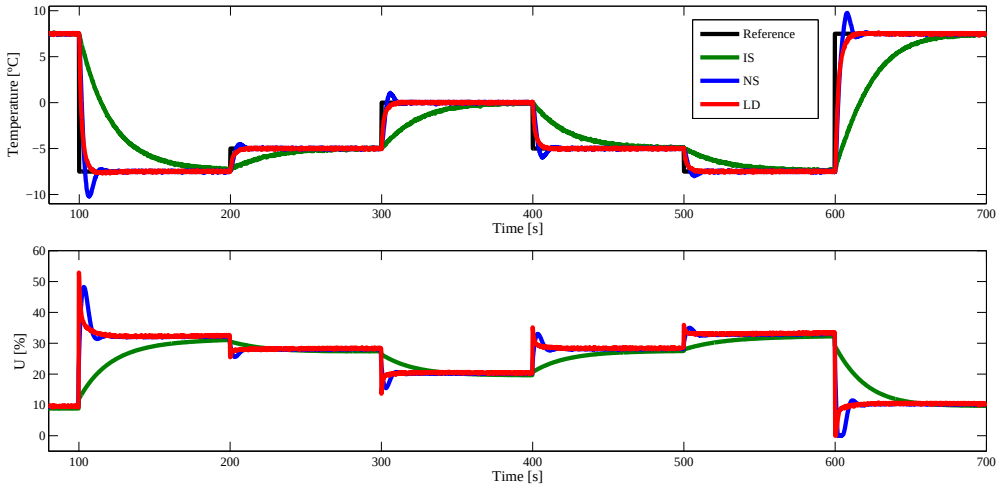


FIGURE 6.10: Performance achieved of the selected design alternatives for the Peltier cell process.

tuning. The procedure implemented here shown to be a useful tool to analyze the trade-off among the design objectives. To validate the approach two different case studies has been considered, the Boiler control problem for controller tuning and as second case, a non-linear Peltier Cell. Both of them based on process models described in terms of FOPDT processes, the formulation of the methodology is completely general. As it can be see it for the result this approach improve the convergence properties and reliable solutions for the Pareto Set approximation. The obtained results validate the proposal for controller tuning.

Part III

Controller tuning applications

Chapter 7

Tuning rules for PI controllers

This chapter addresses PI controller tuning based on multi-objective optimization design procedure in order to obtain an appropriate trade-off selection from the Pareto front. Therefore, the NNC algorithm is used to generate the pareto-optimal solutions and the Nash Solution (NS) technique is implemented to select an unique solution that provides a fair compromise among all the solutions. On the basis of this selection, controller tuning rules are provided for PI controllers and FOPDT models with normalized dead times from $[0.1, 2]$. Simulation examples are given to evaluate the proposed settings.

7.1 Introduction

The PI controllers are easier to understand than other advanced controllers because they have a simple structure. This simplicity is very important to the scientific community that shows continued interest in new tuning methods for finding the optimum parameters [7]. The design of a PI control system starts from a model of the process to be controlled and a set of requirements to be satisfied. These requirements often enter into conflict making the task of finding the appropriate controller parameters not an easy task.

It is on that basis that constrained optimization can enter into play by helping to delimit the trade-off between possible conflicting requirements. Such requirements use to be the conflicting performance and robustness specifications (in the different

forms that they can be established). However, it is well known that design of PI controllers has been based on the derivation of simple tuning rules instead of solving such optimization problems.

On that basis, one possible approach it can be the development of tuning method, by using a multi-objective optimization to obtain evenly distributed Pareto Front approximation that allows the decision maker to decide/choose the must suitable controller for the particular control system. This work presents the tuning rules for PI controller based on the NS as a multi-criteria decision making technique to deal with the performance/robustness trade-off.

7.2 MOOD procedure for PI controller tuning

One goal of this work, is the application of a MOOD framework to obtain the best parameters for the controller tuning, the procedure is described below:

7.2.1 The multi-objective problem definition

Consider the feedback control system represented in Figure 2.1, where the controlled process $P(s)$ and the controller $K(s)$ is described in Equation (2.1) and (4.2), respectively. According to the control loop of Figure 2.1, the design objectives for the controller tuning are: the performance of the closed-loop control system, that it will be evaluated using the IAE given by Equation (2.8), the TV evaluated by Equation (2.9) and the closed-loop control system robustness will be computed using the M_s described in Equation (2.10).

After all, the MOP can be stated as

$$\min_{\boldsymbol{\theta_c}} \boldsymbol{J}(\boldsymbol{\theta_c}) \quad = \quad [J_{IAE}(\boldsymbol{\theta_c}), J_{TV}(\boldsymbol{\theta_c})] \tag{7.1}$$

subject to:

$$1.4 \leq M_s(\boldsymbol{\theta_c}) \leq 2.0 \tag{7.2}$$

where $\boldsymbol{\theta_c} = [K_p, T_i]$ and M_s is the maximum sensitivity of the control system, which is constrained in the interval of $[1.4, 2.0]$, the high and low robustness level, respectively, as it is usually done in process control [6]. It is worth to mention that

the set-point response changes are not likely to occur, due to this, in this chapter we will consider the load disturbance response performance.

7.2.2 Multi-objective Optimization process

Therefore, to generate a set of optimal solutions a NNC algorithm is applied to find a Pareto set approximation where each solution within this set defines an objective vector. The projection into the objective space is known as Pareto front approximation and all the solutions in the Pareto front are said to be non-dominated and Pareto-optimal solutions. The obtained Pareto fronts for the different normalized dead time values are shown in Figure 7.1

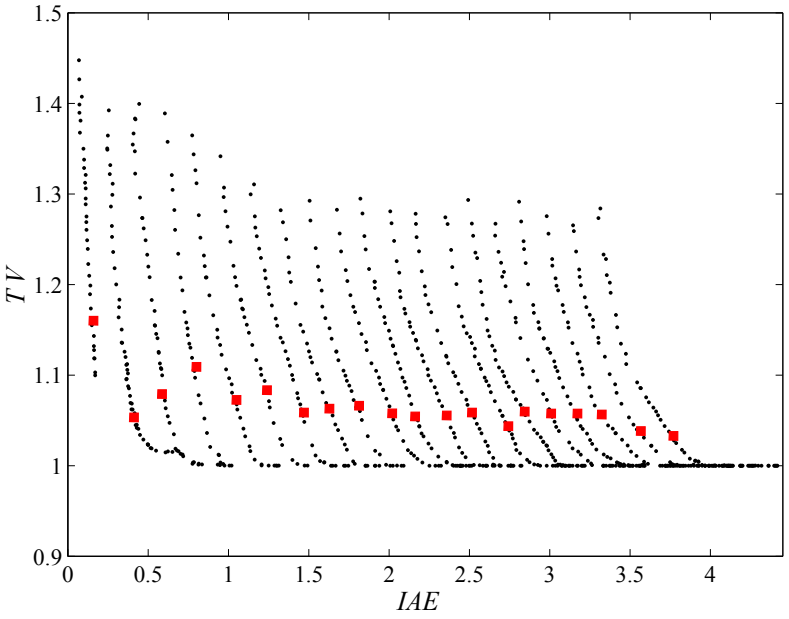


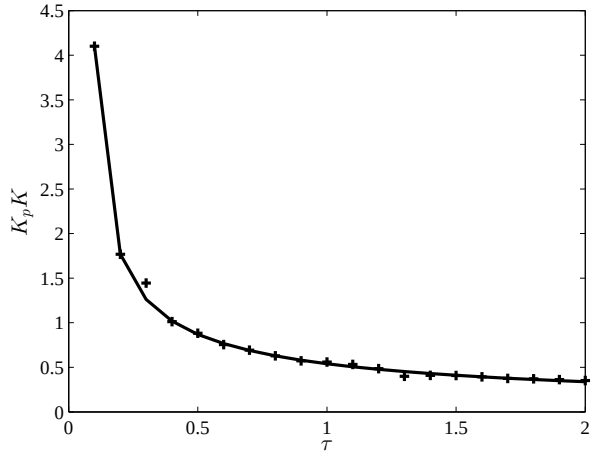
FIGURE 7.1: Pareto Fronts for different normalized dead times and the corresponding Nash Solutions

All the points of the Pareto front are equally acceptable solutions. They just express different trade-off among the competitive objectives.

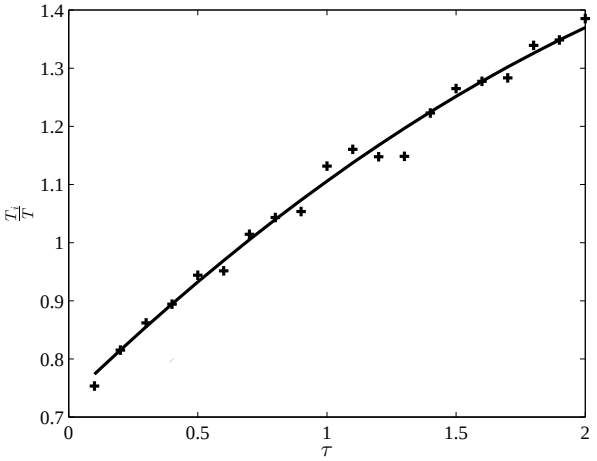
7.2.3 MCDM stage

After the MOO process, a set of solutions is obtained. Then, the final step is to obtain the tuning parameters for the controller, for this we compute the NS for each Pareto front. After the MCDM stage is carried out, a set of NSs is

obtained. Each of them is the optimal solution for a different normalized dead time τ. A set of optimal tuning parameters $[K_p, T_i]$, therefore a specific optimal controller tuning, corresponds to each NS. The whole set of tuning parameters has been calculated by using the least squares fitting technique, as a function of the normalized dead time τ, leading to the results shown in Figure 7.2.



(a)



(b)

FIGURE 7.2: Tuning parameters for the PI controller. *Plus sign*: optimal values of the parameter. *Solid line*: fitting function.

Eventually, the following structure for the controller parameters has been devised:

$$K_p = \frac{1}{K}\left(a\tau^b + b\tau^c\right), \tag{7.3}$$

$$T_i = T\left(a\tau^2 + b\tau + c\right), \tag{7.4}$$

where the values of the coefficients are show in Table 7.1.

TABLE 7.1: Tuning rules coefficients of the PI controllers.

	a	b	c	d
K_p	0.001088	-3.156	0.5391	-0.6733
T_i	-0.05514	0.4295	0.7315	

7.3 Simulation Examples

Consider the fourth-order controlled processes proposed as benchmark in [8] and given by the transfer function:

$$P_\alpha(s) = \frac{1}{\prod_{n=0}^{3}(\alpha^n s + 1)}, \tag{7.5}$$

with $\alpha \in \{0.1, 0.5, 1.0\}$.

Using the three-point identification procedure *123c* [1] FOPDT models were obtained, whose parameters are listed in Table 7.2.

TABLE 7.2: Example - $P_\alpha(s)$ FOPDT models

α	K_p	T	L	τ
0.10	1	1.003	0.112	0.112
0.50	1	1.247	0.691	0.554
1.0	1	2.343	1.860	0.794

For comparison purposes the following PI tuning methods that include a design parameter to deal with the performance/robustness trade-off were selected: MoReRT, Kappa-Tau, S-IMC and AMIGO. This tuning methods were already described in Chapter 4. All of these methods use a FOPDT model approximation for the controlled process.

The tuning rules have been applied to the processes (7.5) and the resulting values of the controller parameters, of the integrated absolute errors, the total variations

and the maximum sensitivity for the different cases are shown in Tables 7.3, 7.4 and 7.5. The processes responses and the control variables for the different cases are plotted in Figure 7.3, 7.4 and 7.5.

Tuning rule	K_p	T_i	β	IAE	TV	Ms
NS	3.46	0.78	1	0.23	1.12	1.44
MoReRT ($M_s = 1.4$)	3.14	0.68	0.60	0.22	1.17	1.40
MoReRT ($M_s = 1.6$)	4.15	0.56	0.54	0.14	1.31	1.60
MoReRT ($M_s = 1.8$)	4.93	0.50	0.52	0.10	1.42	1.80
MoReRT ($M_s = 2.0$)	5.58	0.46	0.51	0.08	1.51	2.0
KT ($M_s = 1.4$)	2.06	0.71	0.89	0.34	1.11	1.24
KT ($M_s = 2.0$)	4.90	0.71	0.47	0.14	1.25	1.71
S-IMC ($M_s \approx 1.59$)	4.46	0.90	1	0.20	1.12	1.60
AMIGO ($M_s = 1.4$)	2.48	0.64	0	0.26	1.17	1.31

TABLE 7.3: Results related to $\alpha = 0.1$ process.

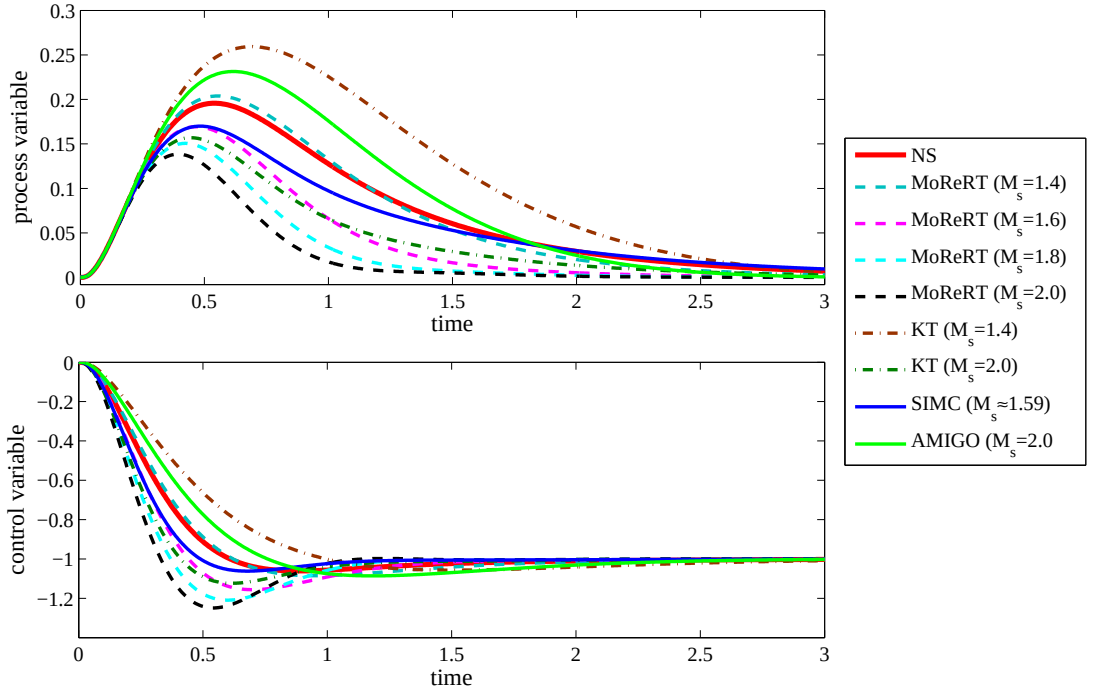


FIGURE 7.3: Temporal Responses for $\alpha = 0.1$ process.

Tuning rule	K_p	T_i	β	IAE	TV	Ms
NS	0.81	1.19	1	1.47	1.14	1.53
MoReRT ($M_s = 1.4$)	0.72	1.44	0.94	2.0	1.00	1.40
MoReRT ($M_s = 1.6$)	0.98	1.46	0.77	1.49	1.12	1.60
MoReRT ($M_s = 1.8$)	1.18	1.44	0.68	1.22	1.30	1.80
MoReRT ($M_s = 2.0$)	1.34	1.41	0.63	1.06	1.46	2.01
KT ($M_s = 1.4$)	0.32	0.81	1	2.60	1.03	1.30
KT ($M_s = 2.0$)	0.67	0.81	0.55	1.56	1.44	1.66
S-IMC ($M_s \approx 1.59$)	0.90	1.25	1	1.38	1.17	1.59
AMIGO ($M_s = 1.4$)	0.37	1.16	0	3.15	1.00	1.21

TABLE 7.4: Results related to $\alpha = 0.5$ process.

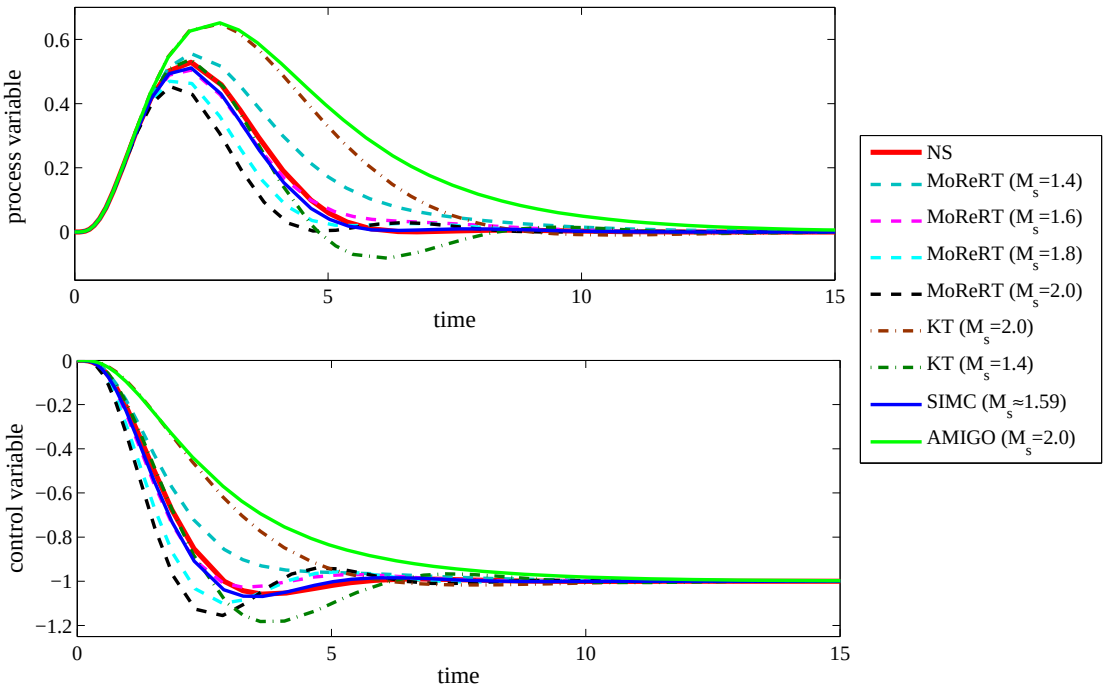


FIGURE 7.4: Temporal Responses for $\alpha = 0.5$ process.

Tuning rule	K_p	T_i	β	IAE	TV	Ms
NS	0.63	2.43	1	3.87	1.16	1.57
MoReRT ($M_s = 1.4$)	0.53	2.83	1.10	5.32	1.00	1.40
MoReRT ($M_s = 1.6$)	0.73	3.01	0.87	4.10	1.11	1.60
MoReRT ($M_s = 1.8$)	0.89	3.05	0.76	3.43	1.34	1.80
MoReRT ($M_s = 2.0$)	1.01	3.05	0.69	3.11	1.54	2.01
KT ($M_s = 1.4$)	0.23	1.60	1	7.00	1.00	1.29
KT ($M_s = 2.0$)	0.49	1.59	0.57	4.38	1.47	1.71
S-IMC ($M_s \approx 1.59$)	0.63	2.34	1	3.82	1.20	1.59
AMIGO ($M_s = 1.4$)	0.28	2.27	0	8.11	1.00	1.23

TABLE 7.5: Results related to $\alpha = 1.0$ process.

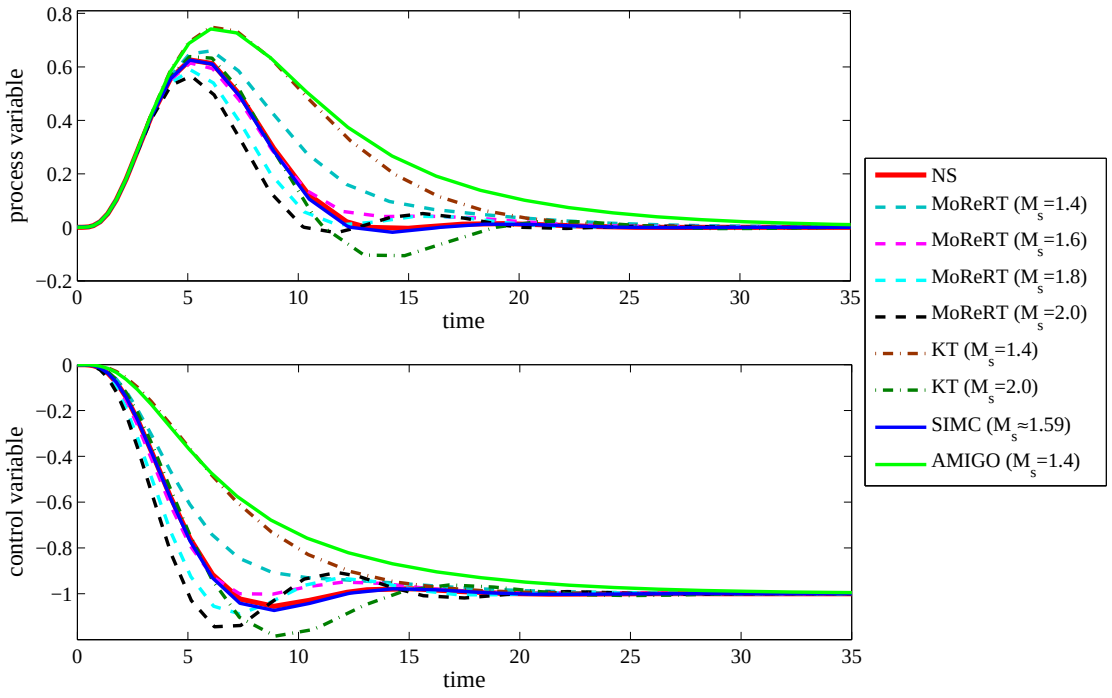


FIGURE 7.5: Temporal Responses for $\alpha = 1.0$ process.

As it can be observed for the processes described before, the tuning method proposed here offers a good balance between the performance and robustness, for the $\alpha = 0.1$ process, the NS offers a better control total variation than the *MoReRT* method, for the $\alpha = 0.5$ and $\alpha = 1.0$ processes the *NS* brings a suitable solution which means a good trade-off between the goal functions. In the case of the *KT*

method neither of them accomplish the robustness level for which they were designed. For example, for the three processes, in the case of *KT* ($M_s = 1.4$) the robustness average was $M_s \approx 1.29$ and for *KT* ($M_s = 2.0$), the robustness average was $M_s \approx 1.69$. Otherwise, the *S-IMC* achieve the desired robustness level $M_s \approx 1.59$, give slightly better performance but the NS offers a better control total variation and a fair robustness level. Finally, the *AMIGO* method accomplished a bit high performance than the NS but the robustness level is not the one it was designed for it (is very far) which give a very slow responses.

7.4 Summary

A Nash based tuning rule for PI controllers have been formulae for a FOPDT models that consider the performance and cost in terms of input usage. The proposed method offer a good balance between the competitive objectives including the robustness level, without constraining to a specific value and avoiding lose much performance. This work has provided an analysis of performance/robustness trade-off by several PI tuning rules. The performance was optimized by using the IAE for disturbance rejection changes. Even some of the analyzed methods do not provide the designed robustness level. The proposed tuning method for PI controllers guarantees a range extending from the low to the high robustness level. The proposed tuning rules, allows the designer to offers a good trade-off between the competitive requirements of the control system. It is worth while to mention, that in this chapter we only consider the regulation mode but in the next chapter both operation modes will be considered.

Chapter 8

Tuning rules for PID controllers

Owing to the fact, that sometimes a one-degree-of-freedom (1DOF) PID has to deal with both set-point changes and load disturbances, it would be desirable to have a tuning method to achieve a good trade-off for this situation. In order to find the best parameters for the controller tuning a multi-objective optimization design procedure is used, to help the designer with the analysis of the competitive objectives. For the MOO process the NNC algorithm is implemented to search for the optimal area and the Nash solution technique is computed as a MCDM technique for a suitably devised optimization problem. Illustrative simulation examples show the effectiveness of the method.

8.1 Introduction

PID controllers have been employed for one hundred years and a lot of experience in their use has been gained, researchers are continuously investigating new design methodologies in order to improve their overall performance, thanks also to the advancement of the computing technology that makes the application of optimization techniques easier.

The research in PID controllers has always been especially focused on the tuning issue, that is, the selection of the PID parameters that are most suitable for a given application. In particular, the development of tuning rules that allow the user to determine the controller gains starting from a simple process model is a topic that has received great attention since the first proposal made by Ziegler

and Nichols [128]. Indeed, many tuning rules have been devised [69]. They are related to different controller structures (the PID can be, for example, in interacting or non interacting form), different process models (for example, a first- or second-order plus dead time transfer function), different control tasks they are required to address (for example, set-point following or load disturbance rejection) and different approaches they are based on (for example, empirical, analytical, or optimal).
In fact, it has to be recognized that the tuning of the controller has often to be performed by taking into account conflicting requirements. From the point of view of output performance, we can identify a design trade-off by considering the effects of load disturbances and set-point changes on the feedback control system. On the other side, there is also a trade-off between performance and robustness.

However, it is clear that having a tuning rule is much more desirable in order to keep the simplicity of the use of PID controllers, which is one of their most appreciated features. For this reason, tuning rules that achieve the minimization of integral performance criteria have been proposed in the past, by assuming a first-order-plus-dead-time (FOPDT) process model [127], which is known to capture well the dynamics of many self-regulating processes. In these works, the set-point following and the load disturbance rejection performance have been considered separately, which makes the choice of the tuning difficult if both tasks have to be addressed in a given application, especially if a one-degree-of-freedom control structure is considered. Tuning rules for weighted servo/regulation control operations, namely, tuning rules that balance the optimal tuning in the two modes, have been presented in [4]. Nevertheless, in these cases the robustness of the control system has not been considered and this might imply a significant performance decrease if the dynamics of the process changes and/or the control effort is too high.

Taking this into account, the purpose of this chapter is to provide a new solution to the tuning problem. First, a MOOD procedure based on the NNC algorithm is applied to different FOPDT processes, yielding a set of Pareto fronts. Secondly, the NS [48] is calculated for each case and tuning rules are determined by fitting the results obtained for the different PID parameters. A comparison with the tuning rules proposed in [4] is also performed.

8.2 MOOD procedure for PID controller

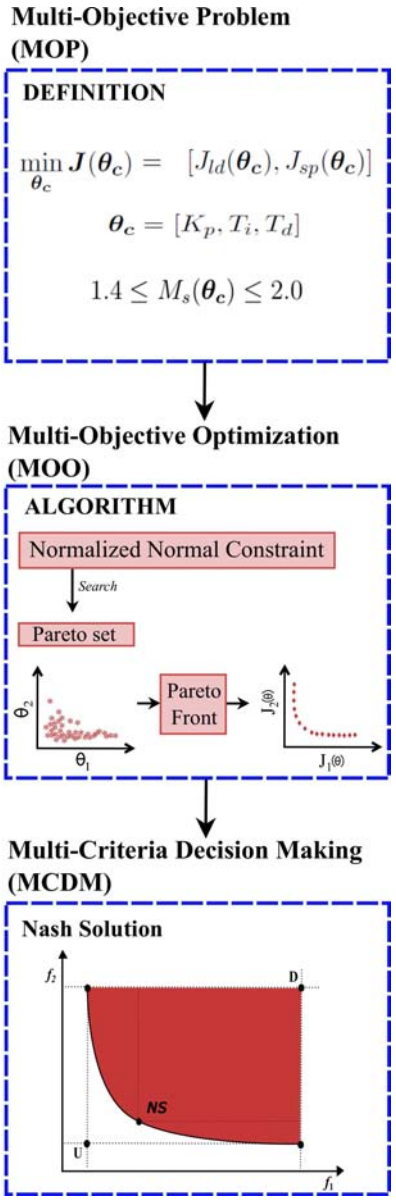


FIGURE 8.1: A Multi-objective Optimization Design (MOOD) procedure.

In Chapter 2 the control problem was formulated. The MOOD procedure implemented in this work is represented in Figure 8.1. First, the MOP definition is described where $\boldsymbol{\theta_c}$ represents the parameters of the PID controller, $J_{ld}(\boldsymbol{\theta_c}) = ISE_{ld}$ is the integrated square error when a load disturbance step response is considered, and $J_{sp}(\boldsymbol{\theta_c}) = ISE_{sp}$ is the integrated square error when a set-point step response is considered. Moreover, the M_s, is constrained in the interval $[1.4, 2.0]$. Secondly, the NNC algorithm is implemented as a MOO techniques to search for a discrete

approximation of the Pareto set capable of generating a good approximation of the Pareto front. Finally, the MCDM stage is carried out in order to select the most preferable solution for the designer, the NS in computed to devise the tuning rules for PID controller.

8.3 Optimal tuning and comparison

Returning to the optimization problem described in Section 8.2, it has been solved in two different ways:

1. Case 1: by unconstraining the M_s in order to compare the proposed method with the intermediate tuning rules presented in [4] where the robustness of the system was not taken into account explicitly.
2. Case 2: by constraining the M_s in a range such as $1.4 \leq M_s(\boldsymbol{\theta_c}) \leq 2.0$., therefore guaranteeing that the resulting control system has an acceptable degree of robustness

The MOO process has been applied, for both cases, to a set of FOPDT processes with different normalized dead time (τ) from 0.1 to 2.0.

For the first case, the optimal parameters proposed in [4] have been used as initial guess for the NNC algorithm. With this tuning rule, the user can select a weighting factor α that determines the importance of the regulation task with respect to the servo task. Therefore, depending on the operation for the control system, we can identify the following cases: $\alpha = 0$ means that only the servo mode is relevant, $\alpha = 0.25$ means that the servo mode is more important than the regulatory mode, $\alpha = 0.50$ means that the two modes are equally important, $\alpha = 0.75$ means that the regulatory mode is more relevant than the servo mode and, finally, $\alpha = 1$ means that only the regulatory mode is relevant. The tuning rules developed in this work have been devised starting from the tuning rules already proposed in [4] for the minimization of the integral square error for the extreme cases, that is, for $\alpha = 0$ and $\alpha = 1$.

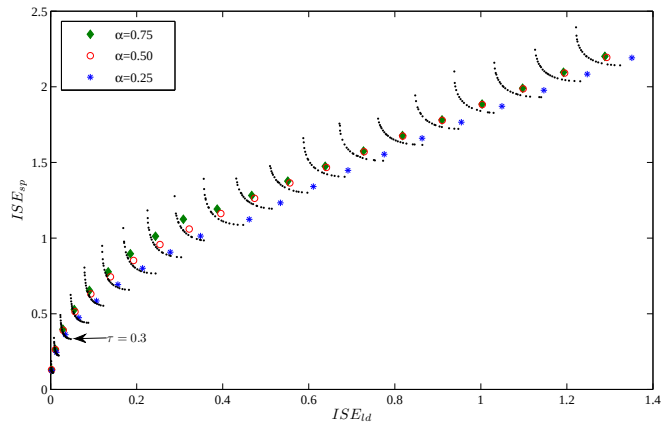


FIGURE 8.2: Pareto fronts for the unconstrained case and performance comparison with the tuning rules proposed in [4] for different normalized dead times.

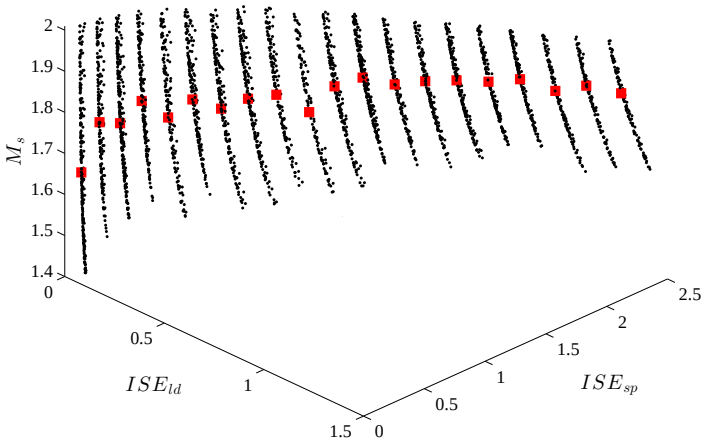


FIGURE 8.3: Pareto fronts for the robust case for different normalized dead times and the corresponding Nash solutions (red square).

The obtained Pareto fronts for the different normalized dead time values are shown in Figure 8.2, where the performance obtained with the intermediate tuning rules proposed in [4] are also shown. As expected, the performance deteriorates when the normalized dead time increases, but it is worth noting that:

- for $\tau < 0.3$, the three weighting factors[1] are located in the Pareto front approximation;
- for $\tau > 0.3$, the three weighting factors are located outside of the Pareto front approximation.

This indicates that these cases are (slightly) suboptimal from the Pareto front point of view (note that, as already mentioned, the intermediate tuning have been developed for the extreme cases and they are the results of fitting strategies). The second case is then developed by applying the MOO process with the robustness constraint $1.4 \leq M_s(\boldsymbol{\theta_c}) \leq 2.0$. The corresponding results are shown in Figure 8.3. A comparison of the two cases can be performed more simply by considering, Figure 8.4 where the results related to $\tau = 1.0$ and $\tau = 1.5$ are highlighted. It appears that achieving a higher robustness is paid in terms of performance or, in other words, just optimizing the performance generally yields a control system with a high maximum sensitivity. The necessity of considering the robustness in the optimal selection of the PID parameters appears clearly in Table 8.1, where it can be noticed that the selection obtained through the constrained Pareto front (NS) proposed in this paper achieves better robustness values and the performance values are not so high in comparison with the intermediate tuning rules from [4] that achieve better performance without taking into account the robustness. Moreover, it is evident that the value of the maximum sensitivity can be critical for those applications where robustness is important.

[1] weighting factors: solutions proposed in [4] for the three weighting factors ($\alpha = 0.25$, $\alpha = 0.50$ and $\alpha = 0.75$).

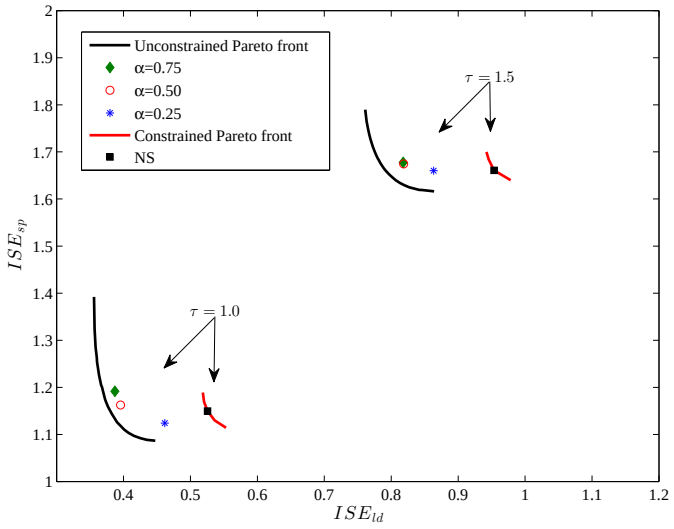


FIGURE 8.4: Comparison between the unconstrained and the constrained (robust case) Pareto fronts.

TABLE 8.1: Results of the comparison between the intermediate tuning rules proposed in [4] and the NS (constrained case).

Tuning rules	τ	ISE_{ld}	ISE_{sp}	M_s
NS	1.0	0.51	1.13	1.84
	1.5	0.96	1.66	1.86
$\alpha = 0.25$	1.0	0.46	1.12	2.86
	1.5	0.86	1.66	2.76
$\alpha = 0.50$	1.0	0.40	1.16	3.02
	1.5	0.82	1.67	2.90
$\alpha = 0.75$	1.0	0.39	1.19	3.34
	1.5	0.82	1.68	2.99

8.4 Tuning rules

After the MCDM stage, a set of NSs is obtained as it is shown in Figure 8.3. Then, the tuning rules have been obtained by a curve fitting procedure that minimizes the least squares errors with respect to the various optimal values. The curve fitting is shown Figure 8.5. As a result, the following tuning rules have been determined:

TABLE 8.2: Parameters for the proposed tuning rules.

	a	b	c	d
K_p	0.233	0.4582	0.7349	-0.9348
T_i	-0.01197	0.5683	0.4343	
T_d	-0.1206	0.5743	-0.01306	

$$K_p = \frac{1}{K}\left(a\tau^b + c\tau^d\right), \tag{8.1}$$

$$T_i = T\left(a\tau^2 + b\tau + c\right), \tag{8.2}$$

$$T_d = T\left(a\tau^2 + b\tau + c\right), \tag{8.3}$$

where the values of the parameters are shown in Table 8.2.

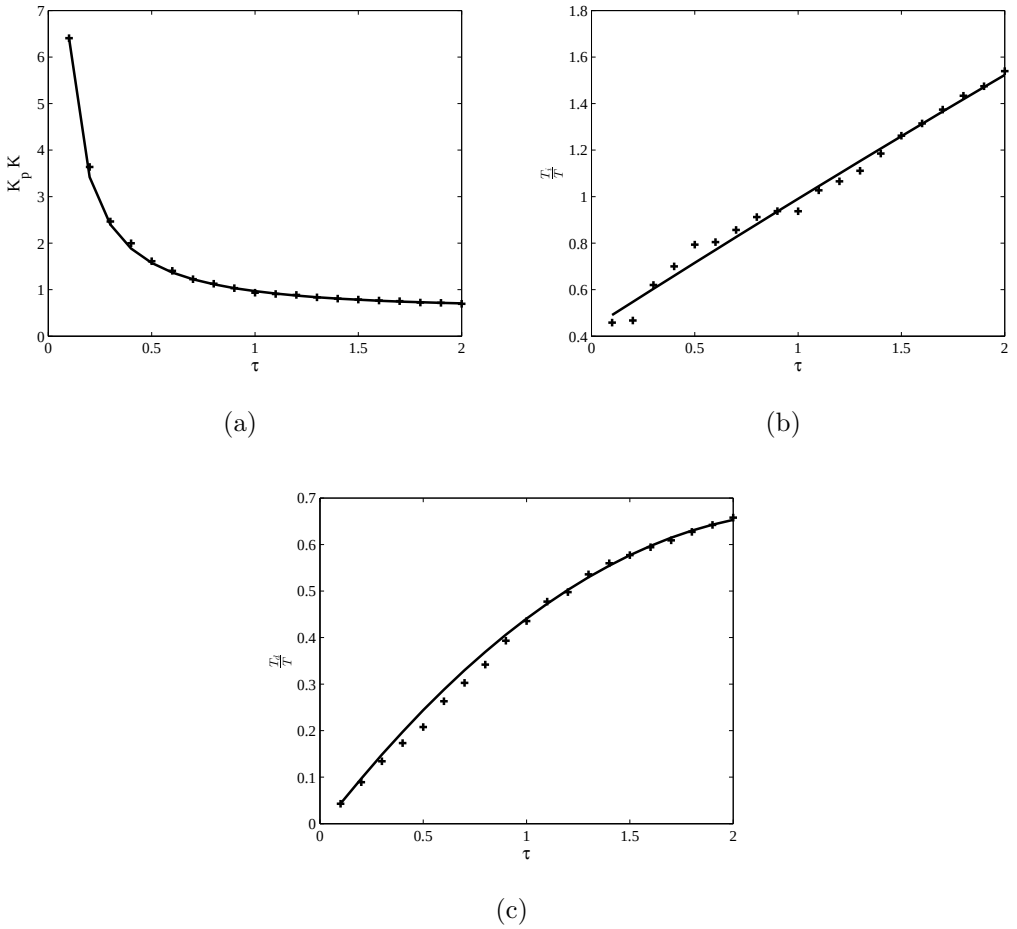


FIGURE 8.5: Tuning parameters for PID controller. Plus sign: optimal values of the parameter. Solid line: fitting function.

A more thorough comparison of the proposed tuning rules with the results obtained by using the intermediate tuning proposed in [4] for the case of $\alpha = 0.5$ is shown in Figure 8.6. It appears that the robustness is increased significantly at the expense of only a slight decrement of the performance in both the set-point and load disturbance step response.

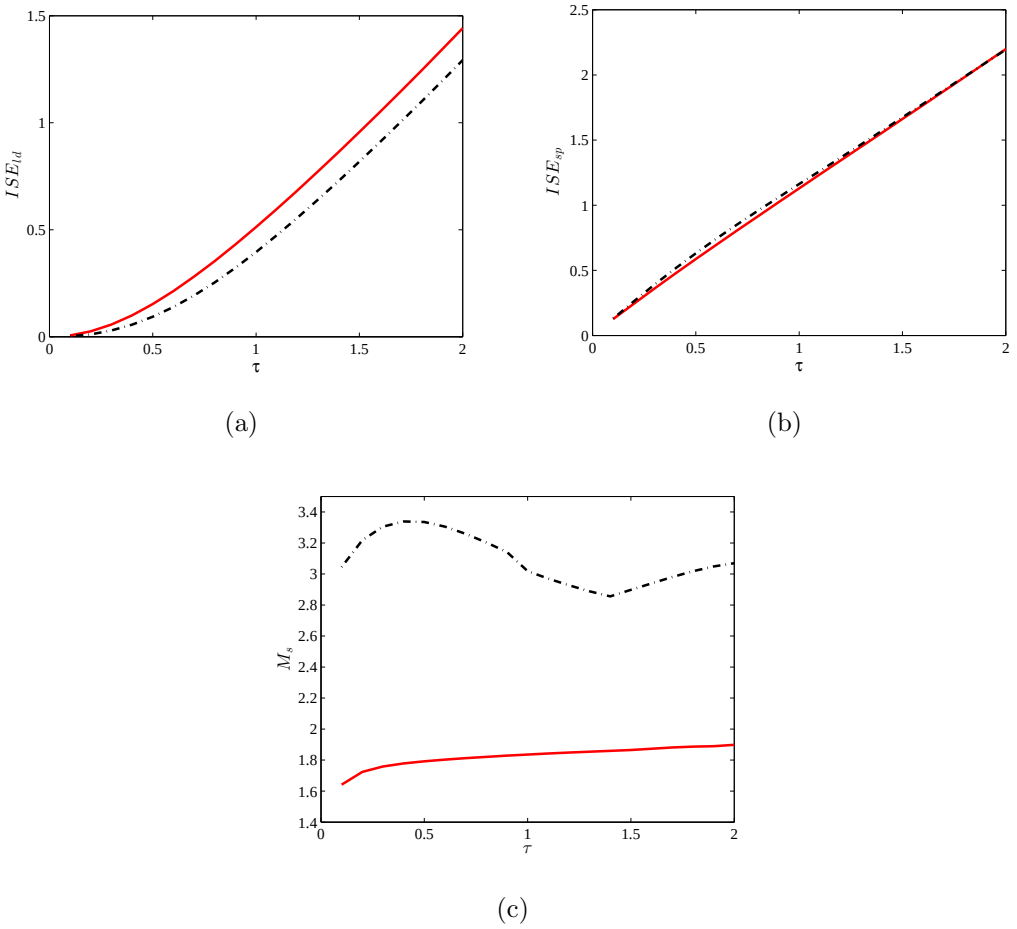


FIGURE 8.6: Comparison between Nash tuning (solid red lines) and intermediate tuning with $\alpha = 0.5$ [4] (dashed black lines) for different normalized dead times.

8.5 Simulation Examples

In this section the tuning rules presented in the previous section are evaluated by considering simulation examples with processes with a different dynamics. A comparison with the tuning rules proposed in [4] is also performed. For each process, the ISE in both the servo and the regulatory tasks, the M_s and the TV

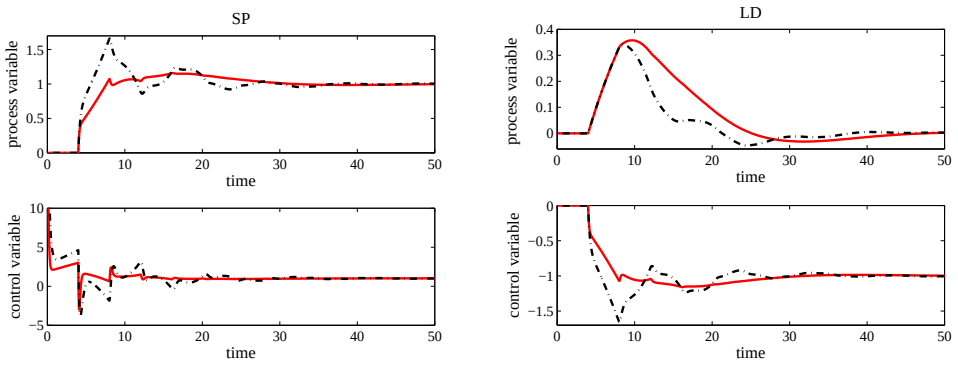


FIGURE 8.7: Set-point (left) and load disturbance (right) step responses for $P_1(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dash-dot line: tuning rules for PID controllers ($\alpha = 0.5$) proposed in [4].

of the control variable for each task are computed in order to provide a more global comparison framework.

8.5.1 FOPDT system

As a first example consider the FOPDT system with $\tau = 0.4$

$$P_1(s) = \frac{1}{10s+1}e^{-4s} \tag{8.4}$$

where, evidently, $K = 1$, $T = 10$, and $L = 4$. The parameters resulting from the application of the tuning rules are shown in Table 8.3. The simulation results related to the both set-point and load disturbance unit step signals are plotted in Figure 8.7. The resulting values of the integrated square error, total variation of the control variable (in order to evaluate the control effort) and maximum sensitivity are shown in Table 8.3.

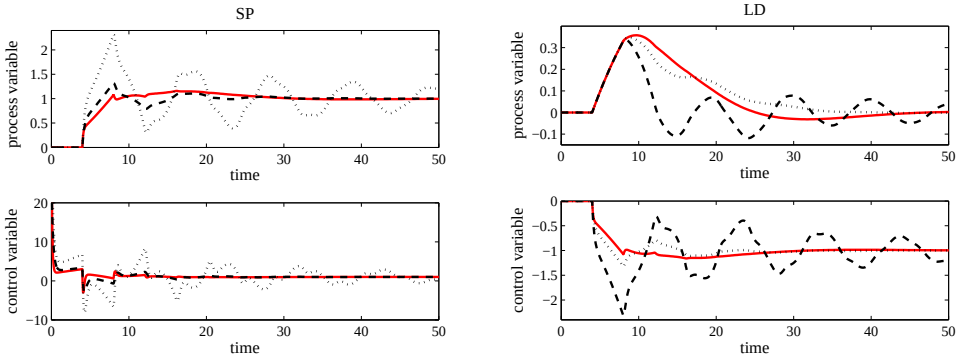


FIGURE 8.8: Set-point (left) and load disturbance (right) step responses for $P_1(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dashed line: tuning rules for servo/regulation PID controllers proposed in [4] applied to the case they have been devised. Dotted line: tuning rules for PID controllers proposed in [4] used for the other control task they have been devised.

Tuning rule	K_p	T_i	T_d	ISE_{sp}	ISE_{ld}	TV_{sp}	TV_{ld}	Ms
NS	1.88	6.60	1.97	4.75	1.00	54.85	1.57	1.77
$\alpha = 0$ (SP)	2.38	9.54	2.17	4.52	0.86	43.97	2.40	2.66
$\alpha = 0.5$	2.74	5.83	2.23	5.13	0.57	61.31	3.43	3.35
$\alpha = 1$ (LD)	3.58	4.50	2.31	11.34	0.53	151.52	10.57	10.98

TABLE 8.3: Results related to $P_1(s)$ ($\tau = 0.4$).

The step responses obtained with the tuning rules proposed in this paper have been compared with those obtained by applying the tuning rules specifically devised in [4] for $\alpha = 0.5$ (see Figure 8.7) which means that the load disturbance and the set-point responses are considered to be equally important (intermediate tuning rules). Further, a comparison with the case $\alpha = 0$ and $\alpha = 1$, which means that only the set-point or load disturbance response, respectively, is relevant, has been also performed. Results are shown in Figure 8.8 (dash-dot line) where, in order to shown that it is worth developing an intermediate tuning rule, it has also been plotted the results obtained by inverting the use of the tuning rules proposed in [4] that is, the tuning rule devised for the set-point following task is applied to the load disturbance task and vice versa (dotted line).

From the obtained results, it can be seen that the proposed robust intermediate tuning rules generate smoother responses with less control effort. This can be also observed in Table 8.3, where the values of TV are generally lower in comparison

with the other tuning rules. Indeed, the performance obtained with the NS tuning is balanced and with a satisfactory robustness. In fact, even if achieving a good level of robustness is paid a little bit in terms of ISE performance, results show the need of considering the maximum sensitivity in a balanced tuning framework also because of the obtained reduced control effort.

8.5.2 High-order process

As a second example, a process with high-order dynamics have been considered:

$$P_2(s) = \frac{1}{(s+1)^4} \tag{8.5}$$

In order to apply the proposed tuning rule, the process has been modelled as a FOPDT with $K = 1$, $T = 2.9$, $L = 1.42$ (thus, the normalized dead time can be determined as $\tau = 0.49$). The results related to both the set-point and load disturbance unit step signals are plotted in Figures 8.9 and 8.10 and the resulting PID parameters as well as performance indexes are summarized in Table 8.4. From the obtained results it can be appreciated that the proposed approach provides also in this case a balanced performance and the sensitivity value is significantly reduced with respect to the other cases where robustness has not been taken into account explicitly.

Tuning rule	K_p	T_i	T_d	ISE_{sp}	ISE_{ld}	TV_{sp}	TV_{ld}	Ms
NS	1.60	2.06	0.69	1.93	0.53	33.87	2.03	1.78
$\alpha = 0$ (SP)	1.99	2.86	0.75	1.60	0.41	22.22	1.75	2.59
$\alpha = 0.5$	2.31	1.91	0.77	1.96	0.33	27.62	2.86	3.31
$\alpha = 1$ (LD)	2.94	1.52	0.81	3.04	0.32	43.88	5.78	9.27

TABLE 8.4: Results related to $P_2(s)$ ($\tau = 0.49$).

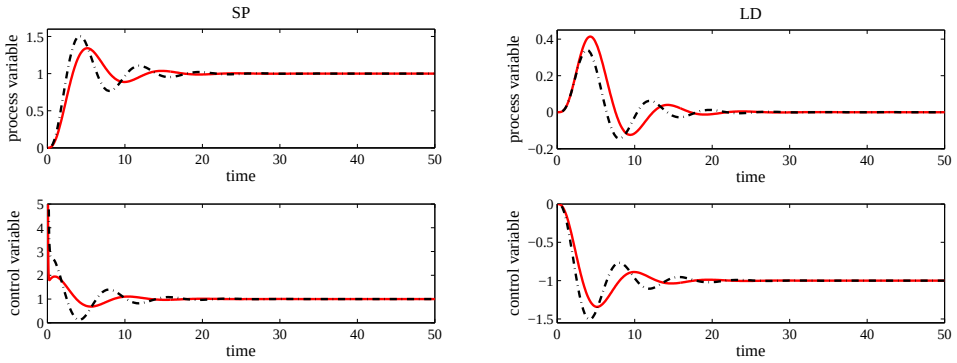


FIGURE 8.9: Set-point (left) and load disturbance (right) step responses for $P_2(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dash-dot line: tuning rules for PID controllers ($\alpha = 0.5$) proposed in [4].

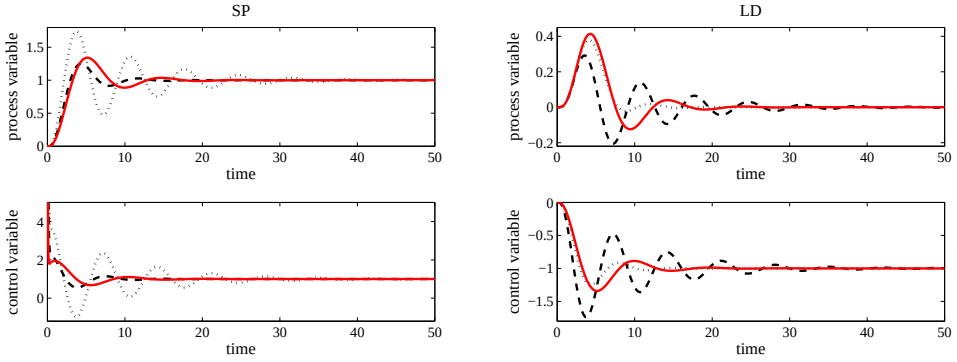


FIGURE 8.10: Set-point (left) and load disturbance (right) step responses for $P_2(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dashed line: tuning rules for servo/regulation PID controllers proposed in [4] applied to the case they have been devised. Dotted line: tuning rules for PID controllers proposed in [4] used for the other control task they have been devised.

As a third example, a process with a much higher number of coincident poles has been considered:

$$P_3(s) = \frac{1}{(s+1)^{20}} \tag{8.6}$$

In order to apply the proposed tuning rules, the process has been modelled as a FOPDT system with $K = 1$, $T = 7.76$, $L = 12.72$ (thus, the normalized dead time can be determined as $\tau = 1.64$). The results are shown in Table 8.5, the responses are plotted in Figure 8.11 and the resulting PID parameters as well as the performance indexes are summarized in Table 8.5. As it can be observed,

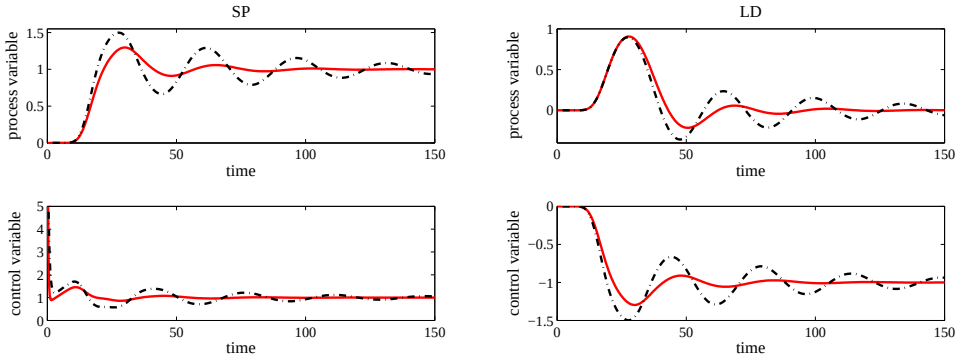


FIGURE 8.11: Set-point (left) and load disturbance (right) step responses for $P_3(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dash-dot line: tuning rules for PID controllers ($\alpha = 0.5$) proposed in [4].

the proposed tuning rule, generally, provides the minimum ISE for the servo and regulatory task. In the responses shown in Figure 8.11, it can be seen that the tuning rule proposed in this work offers an optimal balance in comparison with the extreme cases ($\alpha = 0$ and $\alpha = 1$).

Tuning rule	K_p	T_i	T_d	ISE_{sp}	ISE_{ld}	TV_{sp}	TV_{ld}	Ms
NS	0.76	10.35	4.69	15.85	11.46	16.64	1.95	1.89
$\alpha = 0$ (SP)	0.87	11.31	5.40	17.38	11.62	12.01	3.64	2.59
$\alpha = 0.5$	0.91	10.63	5.63	18.98	12.25	13.45	4.45	2.97
$\alpha = 1$ (LD)	1.06	9.43	6.52	49.11	24.78	26.73	11.97	7.66

TABLE 8.5: Results related to $P_3(s)$ ($\tau = 1.64$).

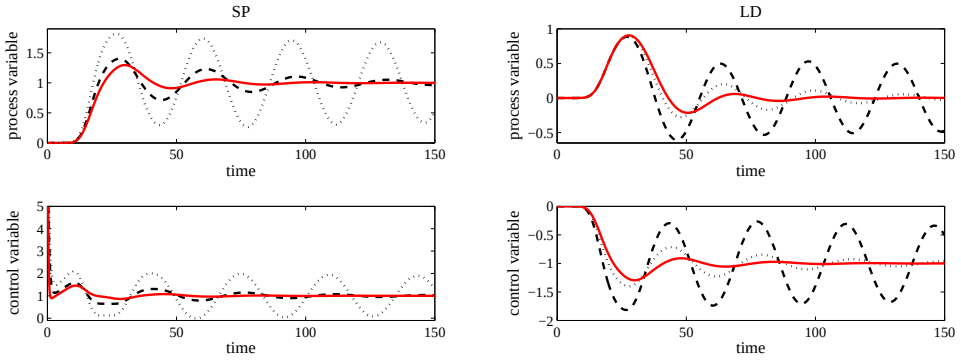


FIGURE 8.12: Set-point (left) and load disturbance (right) step responses for $P_3(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dashed line: tuning rules for servo/regulation PID controllers proposed in [4] applied to the case they have been devised. Dotted line: tuning rules for PID controllers proposed in [4] used for the other control task they have been devised.

8.6 Summary

A set of tuning rules for one-degree-of-freedom PID controllers based on a multi-objective optimization strategy has been presented and quantitatively analyzed. In particular, the tuning allows the minimization of the integrated square error for the set-point following and load disturbance rejection task subject to a constraint on the maximum sensitivity.

This methodology addresses two different trade-off: the performance/robustness and the servo/regulatory control mode. It has been shown that, in this context, the robustness of the system can be a critical issue and therefore it has to be included explicitly in the optimization procedure. The tuning rules developed in this Chapter are based on the Nash solution and it can be easily implemented in standard industrial controllers.

Chapter 9

Tuning rules for FOPID controllers

Chapter 5 presented a Multi-stage approach for controller tuning. It was shown that, this approach look for a set of solutions with some desirable characteristics as exploitation capabilities. This Chapter implement such approach to devise a set of balanced tuning rules for fractional-order proportional-integral-derivative controllers. The control problem is stated as a multi-objective optimization problem where a first-order-plus-dead-time process model has been considered. A set of Pareto optimal solutions is obtained for different normalized dead times and then the optimal balance between the competing objectives is obtained by choosing the Nash solution among the pareto-optimal ones. A curve fitting procedure has then eventually been applied in order to generate suitable tuning rules. Several simulation results show the effectiveness of the proposed approach.

9.1 Introduction

A considerable number of tuning rules for PID controllers has been developed in the last decades. In recent years, fractional control has received a great interest from the control community due to the fact that it provides more flexibility in the design phase than the classical integer order one. This leads to controllers that are

capable to accomplish more demanding control requirements [21, 23, 40, 63, 65, 70–73, 94, 120]. Owing to this, it seems natural to generalize the derivative and the integral orders of the PID controller to any real number leading to the fractional-order PID (FOPID) controller [80, 122].

The use of a FOPID controller implies that a better performance can be achieved, but on the other hand it means that design can be more difficult. However, in order to foster the application of FOPID controllers in industry, the same ease of use as classical PID controllers must be ensured. For this purpose, the stability issue of this kind of controllers has been investigated in [18, 19]. Then, a large number of strategies to tune a FOPID controller has been proposed in the literature to facilitate their implementation [119]. In particular, optimization techniques have been proposed, mainly with the aim of achieving the so-called iso-damping property [12, 64, 118], that is, achieving a flat phase at the gain crossover frequency so that the same overshoot is obtained in the set-point step response despite process gain variations [14]. It has however to be recognized that, as for standard PID controllers, the presence of tuning rules can represent a key factor for the success of FOPID controllers. Taking this into account, rules that consider the optimization of the load disturbance response with a robustness constraint have been proposed [22, 39]. Similarly, the minimization of the integrated absolute error with a constraints on the maximum sensitivity has been pursued in [74], where both the set-point following and the load disturbance rejection tasks have been considered separately.

It turns out that a FOPID controller tuning rule that addresses both the set-point following task (servo mode) and the load disturbance task (regulatory mode) at the same time is still missing. Indeed, for the design of a control system, it is important to take into account the trade-off between these specifications. Moreover, the other important trade-off between performance and robustness has to be considered. Nevertheless, it is well known that these are in general competing objectives that have been already investigated for standard, integer order, PID controllers (see [4], where however the robustness issue is disregarded).

In this context, as there are different conflicting requirements to handle, it is natural to set up a MOP [57]. In fact, in general, a good disturbance rejection response is not compatible with a good set-point step response and a high performance is often not compatible with a controller which is robust to process model mismatch. Eventually, tuning techniques can be obtained based on the so called

Pareto front approximation, where all the solutions are Pareto-optimal and offer different trade-offs between the objectives, see for example [34, 42, 60, 77].

In order to find a controller with an optimal balance between the posed objectives a MOOD procedure is implemented. For this purpose a new approach for the MOO has been implemented in order to improve the convergence capabilities. We consider a servo/regulatory optimally balanced tuning, where the MOO procedure is performed by optimizing the set-point following and the load disturbance rejection performance, measured in terms of the IAE and considering FOPDT processes with different normalized dead times. In order to achieve a reasonable performance/robustness trade-off, the M_s has been used as a measure of the system robustness. Regarding the robustness constraint, a double approach has been pursued: in the first case the value of M_s has been constrained in a reasonable range while in the second case the value of M_s has been constrained to a specific value that the user can selected depending on the application. Then, in order to select the best compromise between the different objectives, the NS has been determined [48] as a MCDM technique for each normalized dead time. Finally, tuning rules have been determined by using a least squares fitting technique with the obtained optimal results with respect to the normalized dead time. It is worth to highlight that the obtained tuning rules follow the same structure for all the situations considered. Simulations results show the effectiveness and the robustness of the proposed tuning rules and the advantage of a unique tuning strategy capable to address both the servo and the regulatory tasks.

9.2 FOPID controller structure

We consider the typical feedback control system represented in Figure 2.1, where the process dynamics can be fully characterized in terms of the normalized dead-time defined in (9.1), which represents a measure of the difficulty in controlling the process:

$$\tau = \frac{L}{T}, \qquad \tau \epsilon \, [0.1, 4]. \tag{9.1}$$

In this work, the process will be controlled with a fractional-order PID controller, which is defined as a generalization of the standard ISA form for the PID controller,

whose transfer function is given by [76]:

$$K(s) = K_p \left(1 + \frac{1}{T_i s^{\lambda}} + \frac{T_d s^{\mu}}{\frac{T_d}{N^{\mu}} s^{\mu} + 1} \right), \tag{9.2}$$

where K_p is the proportional gain, T_i is the integral time constant, T_d is the derivative time constant, N is the derivative filter parameter, λ and μ are the non-integer order of the integral and of the derivative actions, respectively (*i.e.*, the tuning parameters).

Remark 9.1. It is worth mentioning that the derivative filter time constant $\frac{T_d}{N^{\mu}}$ is selected in such a way that the phase transition of the filter occurs in $log(N)$ decades after the derivative term $T_d s^{\mu}$ crosses the 0 dB axis, regardless of the derivative order μ. Indeed, selecting the filter time constant as $\frac{T_d}{N^{\mu}}$ prevents the fractional poles migration [54] when the derivative order changes. Here, $N = 10$ has been selected as it is usual in industrial controllers [3, 123].

Finally, it must be pointed out that, in order to implement the fractional-order controller, the Oustaloup continuous integer-order approximation [71] has been used; it consists in using the following approximation based on a recursive distribution of zeros and poles:

$$s^{\nu} \cong k \prod_{n=1}^{\bar{N}} \frac{1 + \frac{s}{\omega_{z,n}}}{1 + \frac{s}{\omega_{p,n}}}, \qquad \nu > 0, \tag{9.3}$$

where $\omega_{z,n}$ and $\omega_{p,n}$ are, respectively, the frequencies at which the zeros and the poles occurs, $\nu \in \mathbb{R}$ is the fractional order, $\bar{N}$ is the number of poles an zeros used for the approximation and k is the gain. The approximation is only valid in a frequency interval $[\omega_l, \omega_h]$, where ω_l and ω_h are, respectively, the lower and the higher limit. Finally, the gain is adjusted so that both sides of (9.3) have the same gain in the logarithmic mid point of the interval.

In this work the values of ω_l and ω_h have been selected as $0.001\omega_c$ and $1000\omega_c$, respectively, where ω_c is the gain crossover frequency of the loop transfer function. Furthermore, $\bar{N}$ has been chosen equal to 8. It is worth stressing that the use of these criteria to select the approximation parameters leads to a mismatch between the responses that would be obtained with the ideal fractional controller and the approximated controller that is negligible. Indeed, within the approximation frequency range, the selected number of poles and zeros makes the two controllers virtually undistinguishable. Below the lower approximation limit, because of the

large approximation range centered at ω_c, the presence of a (possibly fractional) integrator flats the closed-loop frequency response, both in the ideal and in the approximated cases, along the 0 dB axis. Finally, above the upper limit, the high frequency roll-off makes again the ideal-approximated closed-loop mismatch negligible.

9.3 MOOD procedure for FOPID controller

As stated in the introduction, we aim at finding an optimal tuning for the proposed FOPID controller taking in to account both the servo and the regulatory modes. Therefore, the MOOD procedure for FOPID controllers is described as follow:

MOP definition for FOPID controller

In the previous sections the control system has been described. Therefore, the MOP statement for FOPID controller tuning could be formulated as:

$$\min_{\boldsymbol{\theta_c}} \boldsymbol{J}(\boldsymbol{\theta_c}) \quad = \quad [\boldsymbol{J_{sp}}(\boldsymbol{\theta_c}), \boldsymbol{J_{ld}}(\boldsymbol{\theta_c})] \tag{9.4}$$

such that

$$g_{ls} \leq \boldsymbol{g}(\boldsymbol{\theta_c}) \leq g_{us}, \tag{9.5}$$

where

$$\boldsymbol{\theta_c} = [K_p, T_i, T_d, \mu, \lambda]$$

are the parameters of the FOPID controller (decision space). The IAE (2.8) is used in order to measure the performance of a given set of tuning parameters. Nevertheless, since we must take into account both the servo and the regulatory mode, we must consider two different IAEs, the one obtained in the set-point step response ($\boldsymbol{J_{sp}}(\boldsymbol{\theta_c}) = IAE_{sp}$) and the other obtained in the load-disturbance step response ($\boldsymbol{J_{ld}}(\boldsymbol{\theta_c}) = IAE_{ld}$). Moreover, aiming just at obtaining the minimal IAE may lead to a poor control performance because the robustness issue is not taken into account. For this reason, in this work we also consider the maximum sensitivity M_s, defined in Equation (2.10).

The minimization is constrained by using the maximum sensitivity M_s, that is:

$$\boldsymbol{g(\theta_c)} = M_s \tag{9.6}$$

$$g_{ls} \leq \boldsymbol{g_{(\theta_c)}} \leq g_{us}, \tag{9.7}$$

where g_{ls}, g_{us}, are the lower ($M_s = 1.4$) and upper ($M_s = 2.0$) constraint limits. As it was mentioned before, the M_s can be constrained to be into a range or to be fixed to a specific value. Therefore, the alternative to the inequality constraint (9.7), is to use an equality constraint

$$\boldsymbol{h(\theta_c)} = h, \tag{9.8}$$

where $\boldsymbol{h(\theta_c)} = M_s$ and $h \in \{1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0\}$.

MOO process

In this work, a multi-stage approach is proposed, where two algorithms are merged in a sequential manner for controller tuning, when a MOP is being solved:

- The Normalized Normal Constraint (NCC) algorithm: it is a deterministic algorithm defined in Chapter 3.
- The Multi-objective Differential Evolution Algorithm with Spherical Pruning (sp-MODE): it is an evolutionary optimization approach described in Chapter 5.

For more details about this approach, interested readers can refer to Chapter 5.

MCDM stage

Therefore, for decision making, we used the Nash Solution (NS) described in Chapter 4. The NS lies on the Pareto front surface and is computed as the intersection between the surface and the diagonal line that passes through two opposite vertexes of the smallest cube that inscribes the surface.

9.4 Optimal tuning

In order to find a set of optimal tuning rules for the FOPID controller, the MOOD procedure stated in the previous section has been applied to a set of FOPDT process models with different values of the normalized dead time τ ranging from 0.1 to 4.0. Note that the considered range of normalized dead times spans from lag dominant processes to truly delay dominant ones, being the upper limit $\tau = 4.0$ whereas most of the existing tuning rules cover just till $\tau = 2.0$.
The gain of each process has been set equal to one and the time constant has been normalized to one without loss of generality since the gain is a pure scale factor and the process dynamics is completely parameterized by using the normalized dead time τ.
Unit step signals have been employed for the set-point following and load disturbance rejection tasks in order to compute the corresponding IAEs.
The optimization problem (9.4) has been solved by using two different types of constraints:

1. the maximum sensitivity M_s has to be in a range (hereafter addressed as *M_s-range* case):
$$1.4 \leq M_s \leq 2.0; \tag{9.9}$$

2. the maximum sensitivity M_s has to be equal to a given value (hereafter addressed as *M_s-value case*):
$$M_s = \{1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0\}\,. \tag{9.10}$$

The following framework has been used in order to solve the MOO problem:

- the servo and the regulation operation modes have been considered in the same optimization problem;
- as initial guess for the NNC algorithm, the optimal parameters proposed in [74] were used;
- the sp-MODE algorithm is used with:
  - the scaling factor (F), which is a real and constant factor that controls the amplification of the differential variations that allows the mutant vector. In this context, F=0.5 is usually a good initial choice (see [109]);

– for each target vector and its mutant vector, a trial child vector is calculated. In order to increase the diversity of the disturbed parameter vector the Crossover rate (Cr) is introduced. A good choice for it is Cr=0.9, according to the guidelines proposed in [26, 88, 109].

The outcome of the MOO procedure is a set of Pareto fronts, for different normalized dead times and, in the M_s-*value* case, for different maximum sensitivities. It is worth stressing that the Pareto fronts obtained for the M_s-*value* case are two dimensional curves where the x and y axes are, respectively, IAE_{ld} and IAE_{sp}. Conversely, for the M_s-*range* case, the obtained Pareto fronts are three dimensional surfaces where the x and y axes remain the same, while the z axis is M_s.
Once a suitable set of Pareto fronts has been obtained, the NS for each of them is computed. An optimal tuning for the FOPID controller, *i.e.*, a set of optimal parameter K_p, T_i, T_d, λ and μ , corresponds to each NS.
It is worth noting that, when considering the IAE minimization as the objective function, the obtained optimal integrator is always of integer order (that is $\lambda = 1$), as pointed out in [74–76]. In fact, the optimizer always tries to decrease the integrator order λ. However, for $\lambda < 1$ the IAE is unbounded. Indeed, by using the final value theorem it is immediate to check that the integral error for the set point unit step response is

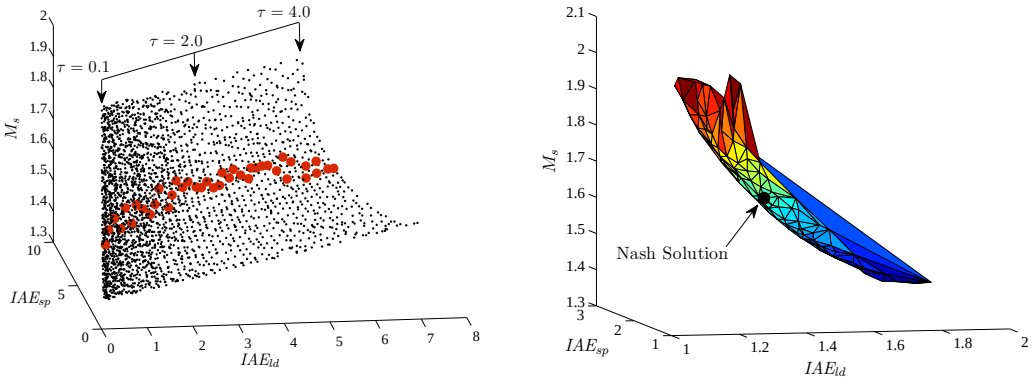
$$\lim_{s\to 0} \frac{s}{s^2} \frac{1}{1+P(s)K(s)} = \frac{1}{s} \frac{s^\lambda \left(1+\frac{T_d}{N^\mu}s^\mu\right)(Ts+1)}{\left(1+\frac{T_d}{N^\mu}s^\mu\right)(Ts+1)+(T_i s^\lambda+1)(T_d s^\mu+1)e^{-Ls}}, \tag{9.11}$$

while the integral error for the unit load disturbance step response is

$$\lim_{s\to 0} \frac{s}{s^2} \frac{P(s)}{1+P(s)K(s)} = \frac{1}{s} \frac{s^\lambda \left(1+\frac{T_d}{N^\mu}s^\mu\right)}{\left(1+\frac{T_d}{N^\mu}s^\mu\right)(Ts+1)+(T_i s^\lambda+1)(T_d s^\mu+1)e^{-Ls}}. \tag{9.12}$$

Evidently, both of them are unbounded when $\lambda < 1$, but a bounded integral error is a necessary condition for a bounded IAE. Hence, the solution $\lambda = 1$ is chosen. This allows the selection of the optimal integrator order analytically. Moreover, another advantage of these results is that the dimensionality of the optimization problem is reduced since λ is fixed in advance.

Eventually, all the optimal parameters are computed as functions of the normalized dead time and, for the M_s-*value* case, also as functions of the desired maximum



(a) (b)

FIGURE 9.1: (a) Pareto fronts for the M_s-*range* case for different normalized dead times and the corresponding NSs and (b) the Pareto front for $\tau = 0.7$ and the corresponding NS.

sensitivity. The fitting function, suitably scaled with respect to the process dc-gain and with respect to process time constant will constitute the set of tuning rules.

9.4.1 M_s-*range* case

As a first case we consider the M_s-*range* case introduced above. We consider the MOP where the M_s is constrained in a range between 1.4 and 2.0. The results obtained for the different normalized dead times are shown in Figure 9.1, where a set of Pareto fronts in the $(IAE_{sp}, IAE_{ld}, M_s)$ space is obtained and the NSs are displayed. It is worth stressing that Pareto fronts are actually surfaces even if this can be hardly appreciated from the figure. Moreover, in order to clarify this issue, the Pareto front for the case $\tau = 0.7$ is shown as an illustrative example.

Tuning rules

After the MCDM stage is carried out, a set of NSs is obtained. Each one of them is the optimal solution for a different normalized dead time τ. A set of optimal tuning parameters $[K_p, T_i, T_d, \lambda, \mu]$, therefore a specific optimal controller tuning, corresponds to each NS.

The whole set of tuning parameters has been calculated by using the least squares fitting technique, as a function of the normalized dead time τ, leading to the results shown in Figure 9.2. Eventually, the following structure for the controller

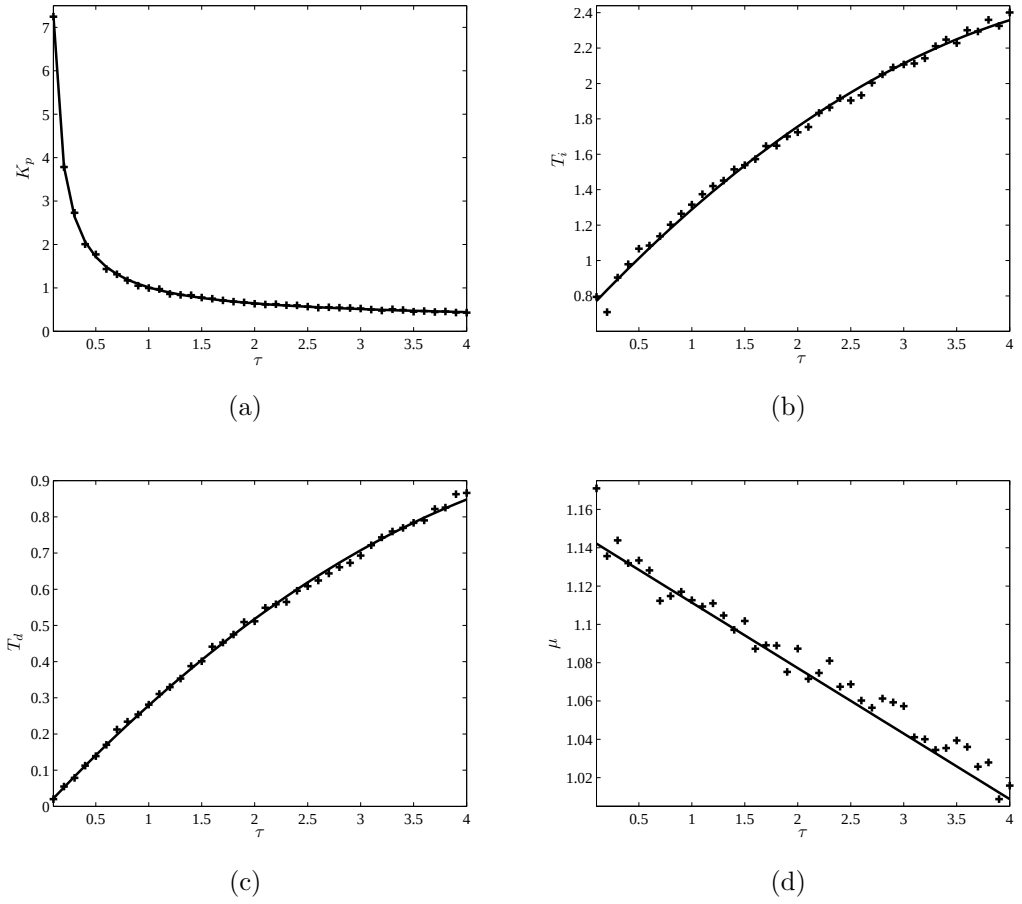


FIGURE 9.2: Tuning parameters for the FOPID controller for the M_s-*range* case. *Plus sign*: optimal values of the parameter. *Solid line*: fitting function.

parameters has been devised:

$$K_p = \frac{1}{K}\left(a\tau^b + c\tau^d\right), \tag{9.13}$$

$$T_i = T^\lambda\left(a\tau^2 + b\tau + c\right), \tag{9.14}$$

$$T_d = T^\mu\left(a\tau^2 + b\tau + c\right), \tag{9.15}$$

$$\mu = a\tau + c, \tag{9.16}$$

$$\lambda = 1, \tag{9.17}$$

where the values of the coefficients are show in Table 9.1. It is worth noting that the fitting functions have been scaled with respect to the process dc-gain K and with respect to the process time constant T. In this way the obtained tuning rules have the nice property of being both time scale and gain invariant. In order

TABLE 9.1: Tuning rules coefficients of the FOPID controllers for the M_s-*range* case.

	a	b	c	d
K_p	0.5951	-1.0417	0.4161	-0.2331
T_i	-0.0558	0.6356	0.7086	
T_d	-0.0246	0.3127	-0.0091	
μ	-0.0342	1.1456		

to achieve this result, the scaling factor of T_i and T_d depends, respectively, on λ and μ. This prevents, again, the fractional pole/zero migration when the orders change.

9.4.2 M_s-*value* case

As a second case we consider the M_s-*value* approach, in which the MOP considered has the value of M_s constrained to each one of the specific values of the set (9.10). Following the same approach used in the previous section, the MOOD procedure has been repeated for different normalized dead times and for different value of M_s. For each Pareto front obtained, the NS is eventually computed.

Tuning rules

Once again, in order to obtain a set of tuning rules for each considered M_s value, each tuning parameter has been calculated by using the least squares fitting technique, as a function of the normalized dead time τ. Hence, the number of fitting is equal to the number of considered levels of M_s, *i.e.*, seven fitting have been performed for each parameter. The structure of the obtained tuning rules for the controller parameters is reported in (9.13)-(9.17), where the values of the coefficients are in Tables 9.2-9.8, depending on the desired level of robustness. It is worth noting that the formulas are the same as the M_s-*range* case due to the fact that the parameters have the same trend. The plots of the fitting curves are not shown for the sake of brevity.

TABLE 9.2: Tuning rules coefficients for the FOPID controllers for the M_s-*value* case with $M_s = 1.4$.

	a	b	c	d
K_p	0.4937	-1.0286	0.2082	-0.1928
T_i	-0.0190	0.3847	0.7913	
T_d	-0.0137	0.3188	-0.0117	
μ	-0.0691	1.1682		

TABLE 9.3: Tuning rules coefficients for the FOPID controllers for the M_s-*value* case with $M_s = 1.5$.

	a	b	c	d
K_p	0.2321	-0.0826	0.5987	-1.0242
T_i	-0.0448	0.5446	0.6857	
T_d	-0.0165	0.3128	-0.0102	
μ	-0.0467	1.1556		

TABLE 9.4: Tuning rules coefficients for the FOPID controllers for the M_s-*value* case with $M_s = 1.6$.

	a	b	c	d
K_p	0.3812	-0.2357	0.5658	-1.0813
T_i	-0.1105	0.8771	0.5574	
T_d	-0.0177	0.3043	-0.0085	
μ	-0.0449	1.1555		

TABLE 9.5: Tuning rules coefficients for the FOPID controllers for the M_s-*value* case with $M_s = 1.7$.

	a	b	c	d
K_p	0.7256	-1.0221	0.3064	-0.0624
T_i	-0.0900	0.8472	0.5113	
T_d	-0.0273	0.3198	-0.0075	
μ	-0.0343	1.1403		

TABLE 9.6: Tuning rules coefficients for the FOPID controllers for the M_s-*value* case with $M_s = 1.8$.

	a	b	c	d
K_p	0.6788	-1.0718	0.4356	-0.1779
T_i	-0.0625	0.7417	0.5621	
T_d	-0.0249	0.3187	-0.0087	
μ	-0.0343	1.1378		

TABLE 9.7: Tuning rules coefficients for the FOPID controllers for the M_s-*value* case with $M_s = 1.9$.

	a	b	c	d
K_p	0.6818	-0.3413	0.4936	-1.1998
T_i	-0.0767	0.8260	0.4788	
T_d	-0.0272	0.3320	-0.0111	
μ	-0.0464	1.1544		

TABLE 9.8: Tuning rules coefficients for the FOPID controllers for the M_s-*value* case with $M_s = 2.0$

	a	b	c	d
K_p	0.4646	-0.1509	0.7615	-1.0902
T_i	-0.0571	0.7219	0.5585	
T_d	-0.0301	0.3464	-0.0133	
μ	-0.0440	1.1430		

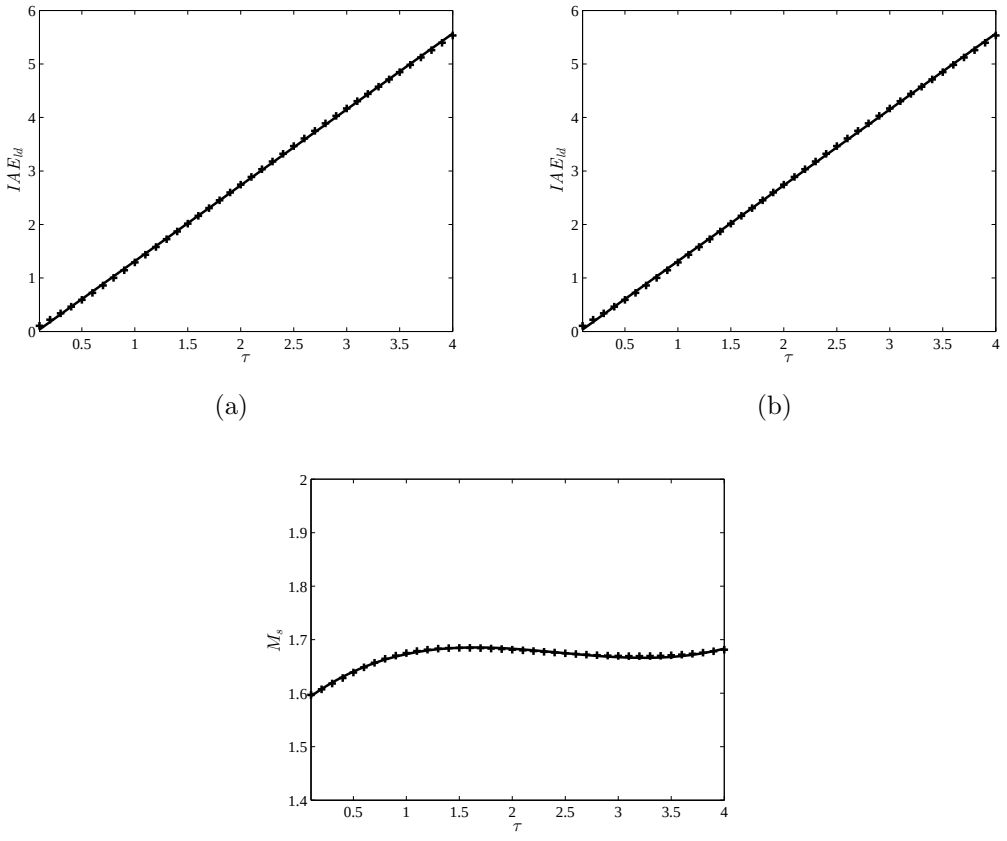


FIGURE 9.3: Performance and robustness assessment index function for the range case. *Solid line*: fitting function (9.18), (9.19) and (9.20). *Plus sign*: optimal value of IAE_{ld}, IAE_{sp} and M_s for the M_s-*range* case.

9.5 Performance assessment

The performance assessment of the proposed rules is addressed in this section in order to provide, in a single and direct way, the expected outcome for the selected controller. The main purpose of the performance assessment is to allow the user to know in advance the values of the IAE cost function that will be obtained (both for set-point following and the load disturbance rejection) when applying the proposed tuning rules. Thus, the user can evaluate the effectiveness of the proposed rules against other ones and decide if they are suitable for a given application without the need of a simulation. Further, the user can calculate the IAE obtained with an existing controller and he/she can decide if it is worth substituting it, as the performance (and the robustness) that can be achieved with the proposed tuning rules can be determined a priori by knowing the FOPDT process model.
When using M_s-*range* rules, the obtained maximum sensitivity is not constrained to a specific value, but in a range from 1.4 to 2.0. Hence, in addition, for the M_s-*range* case, the robustness assessment for the M_s is also presented so that the user can calculate in advance the maximum sensitivity that will be obtained.
In order to obtain the performance assessment rules, the IAE has been calculated for all the considered normalized dead times and maximum sensitivities. The obtained IAE values in the M_s-*range* and in the M_s-*value* case has been calculated by using a fitting technique with suitable functions. Also, the obtained M_s in the case of M_s-*range* has been fitting as a function of the normalized dead time. The obtained results can be expressed as

$$IAE_{ld} = KT\left(a_1\tau + a_2\right), \tag{9.18}$$

$$IAE_{sp} = KT\left(b_1\tau + b_2\right), \tag{9.19}$$

$$M_s = c_1\tau^3 + c_2\tau^2 + c_3\tau + c_4, \tag{9.20}$$

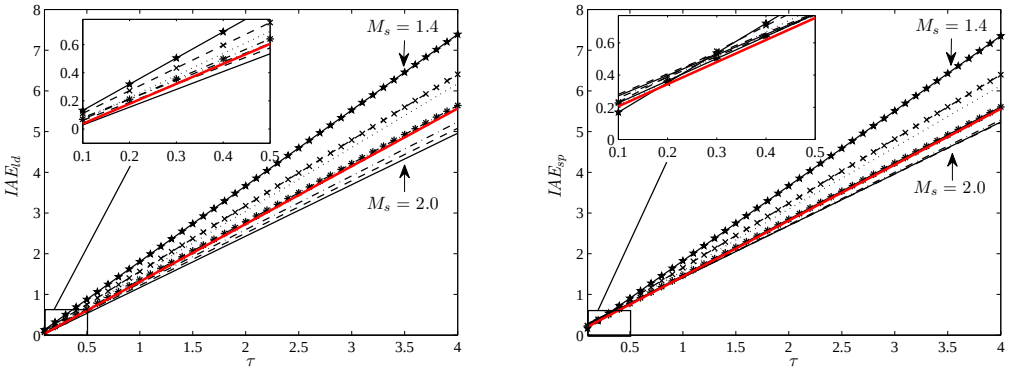
where the values of the coefficients to be employed are shown in Tables 9.9 and 9.10.
The optimal values of IAE_{ld}, IAE_{sp} for the M_s-*range* case, along with the M_s values and the corresponding fitting functions are reported, for the different values of τ, in Figure 9.3. Finally, in Figure 9.4 the same results are illustrated for the M_s-*value* case where, for the sake of comparison, also the M_s-*range* results are plotted. By looking at the results in Figure 9.4, it can be noticed that, after a certain limit that is outlined by the results for the $M_s - range$ case, there is no point in increasing

TABLE 9.9: Performance assessment function parameters for IAE in load disturbance rejection and set-point following task.

	a_1	a_2	b_1	b_2
M_s-*range*	1.4184	-0.1046	1.3731	0.0687
$M_s = 1.4$	1.8603	-0.0537	1.8413	-0.0153
$M_s = 1.5$	1.6148	-0.0508	1.5808	0.0742
$M_s = 1.6$	1.5670	-0.0902	1.5169	0.0613
$M_s = 1.7$	1.4290	-0.0740	1.3793	0.0888
$M_s = 1.8$	1.3217	-0.0521	1.2900	0.1286
$M_s = 1.9$	1.2872	-0.0673	1.2675	0.1541
$M_s = 2.0$	1.2644	-0.0978	1.2734	0.1433

TABLE 9.10: Robustness assessment function parameters for M_s-*range* in load disturbance rejection and set-point following task.

	c_1	c_2	c_3	c_4
M_s-*range*	0.0096	-0.0705	0.1543	1.5797



(a) (b)

FIGURE 9.4: Performance assessment index function for the M_s-*value* case. *Red line*: Performance assessment index for M_s-*range* case.

the M_s, because the performance improvement is not significant compared to the loss of robustness. This behavior is even more evident for low time delays and, for the set-point tracking task, a performance loss is eventually obtained by increasing too much the M_s. Finally, another effect of this behavior can be appreciated in Figure 9.3, where the optimal M_s indeed exhibits a decrement for low dead times.

9.6 Simulation results

In order to verify the effectiveness of the proposed tuning rules, different illustrative simulation results are presented in this section. Moreover, for the sake of comparison, other methods proposed in the literature will be also considered. In particular:

- the optimal tuning rules for FOPID controllers proposed in [74]. These tuning rules are denoted as SP or LD (which means that the set-point following or the load disturbance rejection task are optimized, respectively) followed by the target maximum sensitivity (which can be 1.4 or 2.0) and the letter F (which means that the controller is of fractional order). These rules have been obtained by minimizing the IAE with a constraint on the maximum sensitivity, but the set-point following and the load disturbance rejection tasks have been considered *separately*. Hence, these rules represent the anchor points of the Pareto fronts. Thus, it is worth considering them in order to underlying the benefits of a balanced tuning that considers both tasks at once;

- the intermediate tuning rules for integer order PID controller design proposed in [4], denoted by $\alpha = 0.50$, which means that the load disturbance and the set-point responses are considered to be equally important. These rules, in general, lead to high values of M_s. Indeed, the pure IAE optimization is not sufficient to achieve good level of robustness. Hence, they have been considered in the comparison in order to show the need of constraining the value of M_s also in a balanced tuning framework.

Optimal tuning from [74] are considered because they provide the extremum optimal cases but just considering one operational task for the control system. Therefore the balanced tuning benefits should be faced against the eventual poor performance when the loop operates on the task it was not designed for and, also with the optimal one to evaluate the level of optimality achieved by the balanced tuning. On the other hand, the tuning proposed in [4] has been choosen because it follows similar spirit as the ones proposed here but without the MOO framework and without considering robustness. Therefore some advantages of the current proposal are to be advised. In order to thoroughly test the proposed method,

three kinds of processes will be considered: a FOPDT process (with different normalized dead times), a high-order process and a non-minimum-phase process. In all cases, the processes are represented by FOPTD models with different instances for the value of the normalized dead-time and also representing different kind of plant-model mismatch (higher order and non-minimum phase dynamics). For each process and for each tuning, with the exception of the rules proposed in[4], a global performance index (GPI) has been computed by considering the obtained IAE in both the servo and the regulatory tasks, the M_s and the TV of the control variable for each task. Indeed, although it is clear that reducing the M_s (when considering time delay processes) means also reducing the bandwidth, hence the control effort, a direct evaluation of the control effort in terms of TV is interesting.

The proposed global performance index is computed by normalizing each performance index (IAE_{sp}, IAE_{ld}, TV_{sp}, TV_{ld} and Ms) against its highest value, obtained by ranging over the considered set of tuning rules. Then, the normalized indexes are summed up and the result is eventually divided by the number of considered performance indexes (5 in this case). In this way, the global performance index is always between 0 and 1, where 1 is the worst possible tuning and 0 is the utopia point.
Roughly speaking, the GPI is a compact index that synthesizes the radar plot information. Hence, exactly as for the radar plot, it does not have an absolute meaning; rather, it can be used to evaluate a given approach against a selected benchmark tuning rules. In this context, the GPI and the radar plot are only provided for the proposed rules and the ones of [74], aiming at evaluating the NS against the Pareto front anchor points. On the contrary, the rules proposed in [4] have such a different behavior that it can be appreciated directly from the tables with results and from plot of the responses. Moreover, they would flat down all the differences between the proposed balance tuning and the anchor points (recall that the GPI is relative index) and for this reason they have not been included.

9.6.1 FOPDT processes

The following FOPDT processes are considered, where $K = 1$ and $T = 1$:

$$P_4(s) = \frac{1}{s+1}e^{-0.67s}, \qquad \tau = 0.67, \tag{9.21}$$

$$P_5(s) = \frac{1}{s+1}e^{-2.5s}, \qquad \tau = 2.5. \tag{9.22}$$

First, the tuning rules for the M_s-*range* and M_s-*value* have been applied to the process (9.21) and the resulting values of the controller parameters, of the integrated absolute errors, the total variations and the maximum sensitivity for the different cases are shown in Table 9.11. The process responses and the control variables for the different cases and for both the set-point following and load disturbance rejection tasks are plotted in Figure 9.5. In particular, the M_s-*range* case is considered with the M_s-*value* cases for $M_s = 1.4$ and $M_s = 2.0$. In these latter two cases the step responses obtained with the tuning rules proposed in this paper (solid line) are compared with those obtained by applying correctly the tuning rules specifically devised in [74] for a single task (dashed line) and with those that are obtained by inverting the use of them, that is, the tuning rule devised for the load disturbance is applied to the set-point following task and vice versa (dotted line). In this way, the balancing of the tuning between the two tasks can be clearly seen. Moreover, the performance obtained with the considered tuning rules is compared in a synthetic way by means of the radar diagram in Figure 9.6 and the corresponding global performance indices are shown in Table 9.12. Finally, a deeper analysis has been carried out in order to show the effect of the selected M_s on the process response; the results are shown in Figure 9.7, where the physical meaning of the M_s choice appears.

Tuning rule	K_p	T_i	T_d	μ	λ	IAE_{sp}	IAE_{ld}	TV_{sp}	TV_{ld}	Ms
M_s-*range*	1.36	1.11	0.19	1.12	1	1.02	0.82	30.26	1.40	1.66
M_s=1.4	0.97	1.04	0.20	1.12	1	1.26	1.11	18.94	1.17	1.40
M_s=2.0	1.67	1.02	0.21	1.11	1	1.04	0.65	43.31	2.03	2.00
SP 1.4 F	0.83	0.98	0.22	1.2	1	1.26	1.20	41.20	1.14	1.42
SP 2.0 F	1.26	1.03	0.27	1.2	1	0.92	0.83	78.80	2.06	2.15
LD 1.4 F	0.61	0.54	0.33	1.2	1	1.38	1.12	29.71	1.38	1.44
LD 2.0 F	0.91	0.52	0.38	1.1	1	1.15	0.70	24.50	1.65	1.95
$\alpha = 0.50$	1.78	0.80	0.35	1	1	1.24	0.58	41.07	3.43	3.26

TABLE 9.11: Results related to $P_4(s)$ ($\tau = 0.67$).

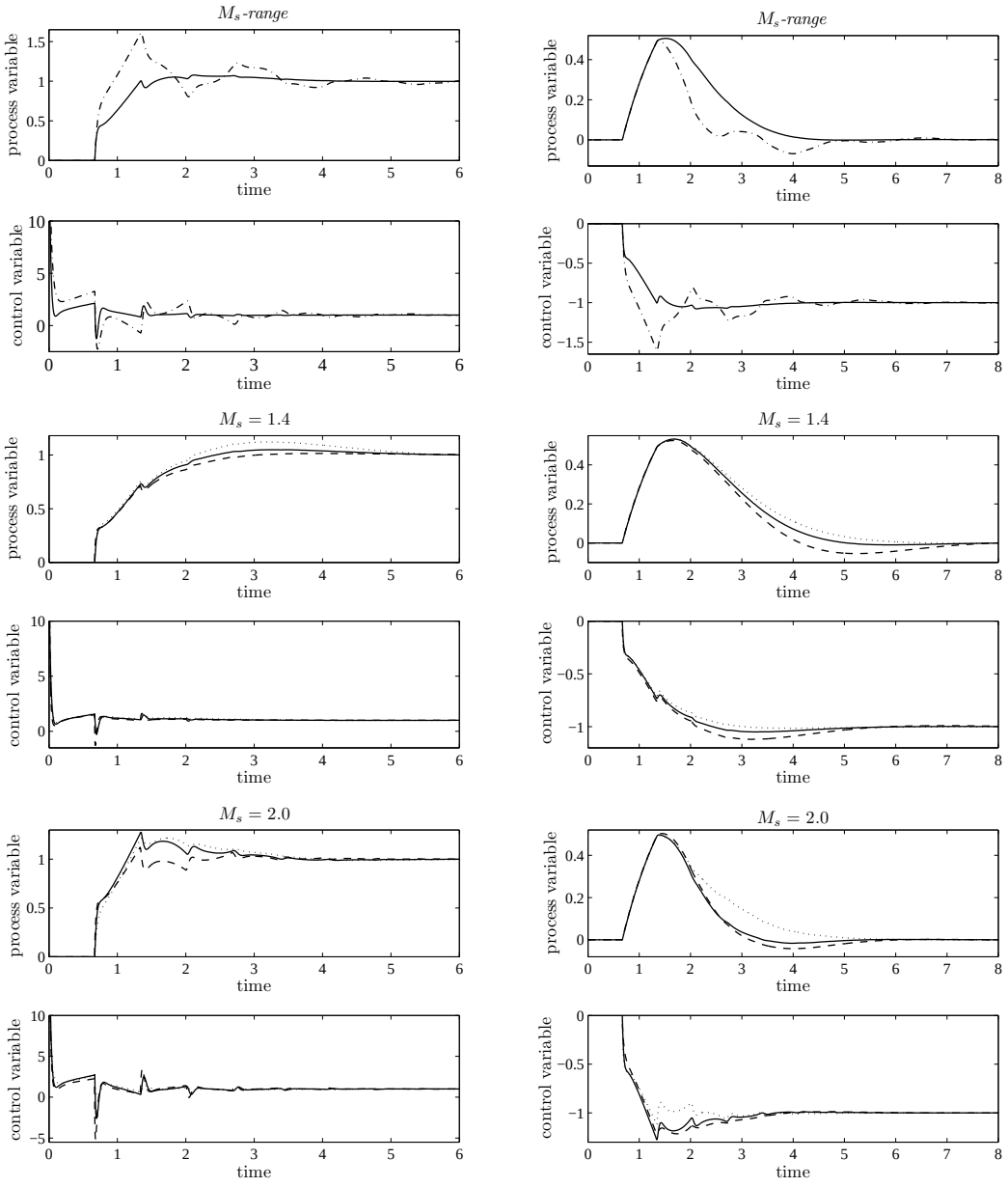


FIGURE 9.5: Set-point and load disturbance step responses for $P_4(s)$. Solid line: proposed tuning rules for FOPID controllers. Dash-dot line: tuning rules for PID controllers proposed in [4]. Dashed line: tuning rules for FOPID controllers proposed in [74]. Dotted line: tuning rules for FOPID controllers proposed in [74] used for the other control task they have been devised.

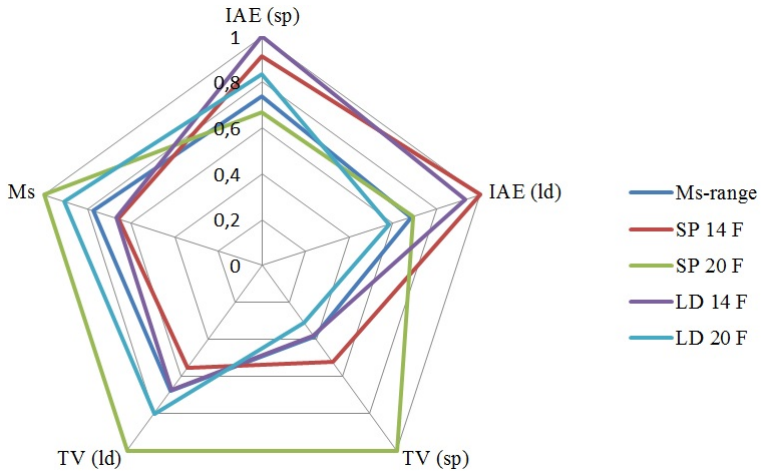


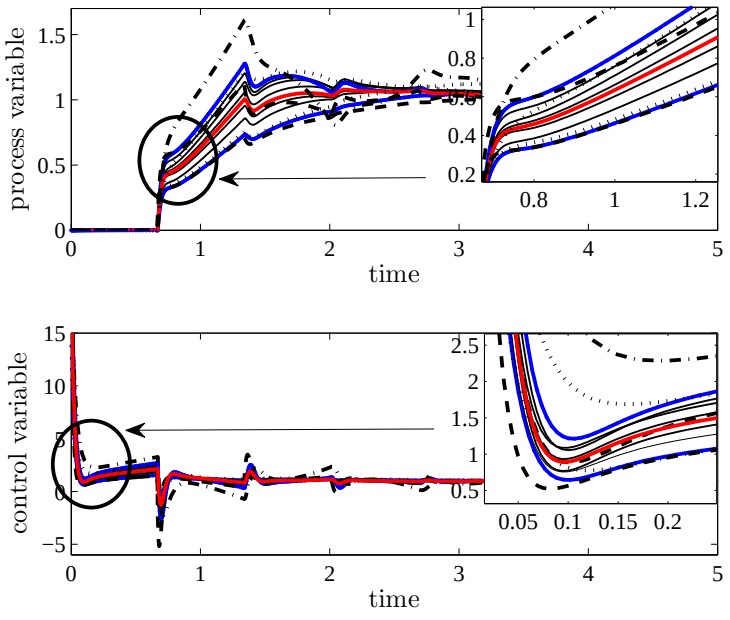
FIGURE 9.6: Radar diagram for $P_4(s)$

Tuning rule	GPI
M_s-*range*	0.65
SP 1.4 F	0.73
SP 2.0 F	0.87
LD 1.4 F	0.73
LD 2.0 F	0.69

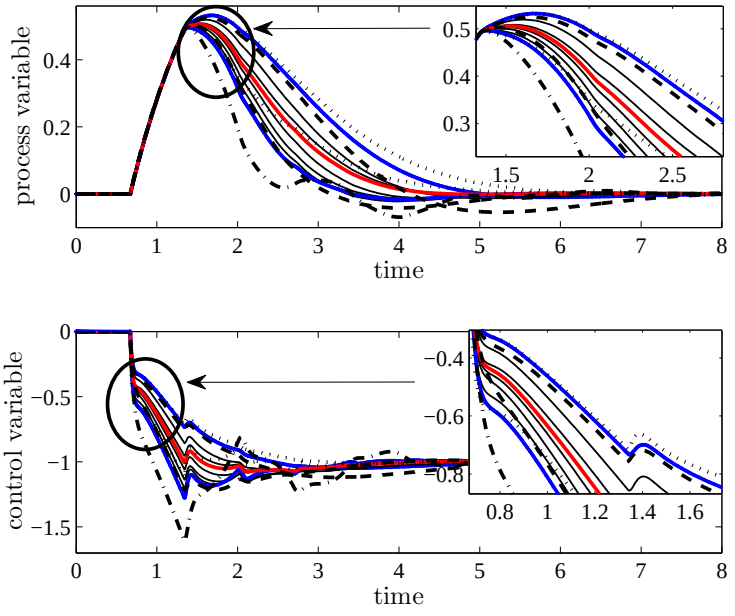
TABLE 9.12: Global performance index for each tuning of $P_4(s)$

Results related to the other FOPDT process (9.22) are shown in Table 9.13. The process responses and the control variables for both the set-point following and load disturbance rejection tasks are plotted in Figure 9.8. Further, the performance obtained with the considered tuning rules is compared by means of the radar diagram in Figure 9.9 and the corresponding global performance indices are shown in Table 9.14.

A deeper analysis has been then carried out again in order to show the effect of the selected M_s on the process response; the results are shown in Figure 9.10, where the physical meaning of the M_s choice appears.



(a)



(b)

FIGURE 9.7: Step responses for $P_4(s)$. Comparison between the tuning rules for M_s-*range* and M_s-*value* case and the tuning rules proposed in [74]. (a) Set-point following task. (b) Load disturbance rejection task. Solid red line: M_s-*range* tuning. Solid blue lines: boundaries of the M_s-*value* tuning. Dashed line: optimal tuning rules proposed in [74]. Dotted line: response obtained using the tuning proposed in [74] for opposite operation mode. Dash-dot line: intermediate tuning rules proposed in [4].

Tuning rule	K_p	T_i	T_d	μ	λ	IAE_{sp}	IAE_{ld}	TV_{sp}	TV_{ld}	Ms
M_s-*range*	0.57	1.95	0.62	1.06	1	3.46	3.44	11.17	1.42	1.68
M_s=1.4	0.37	1.63	0.70	1	1	4.51	4.50	5.21	1.07	1.41
M_s=2.0	0.69	2.01	0.66	1.03	1	3.32	2.99	14.03	1.94	2.00
SP 1.4 F	0.21	0.96	1.07	1.2	1	4.81	4.80	9.71	1.09	1.38
SP 2.0 F	0.35	1.03	1.31	1.2	1	3.40	3.27	19.33	1.94	2.08
LD 1.4 F	0.26	1.08	0.93	1.2	1	4.44	4.42	12.26	1.13	1.44
LD 2.0 F	0.53	1.40	0.90	1.2	1	3.40	2.96	32.59	2.57	2.34
$\alpha = 0.50$	not applicable									

TABLE 9.13: Results related to $P_5(s)$ ($\tau = 2.5$).

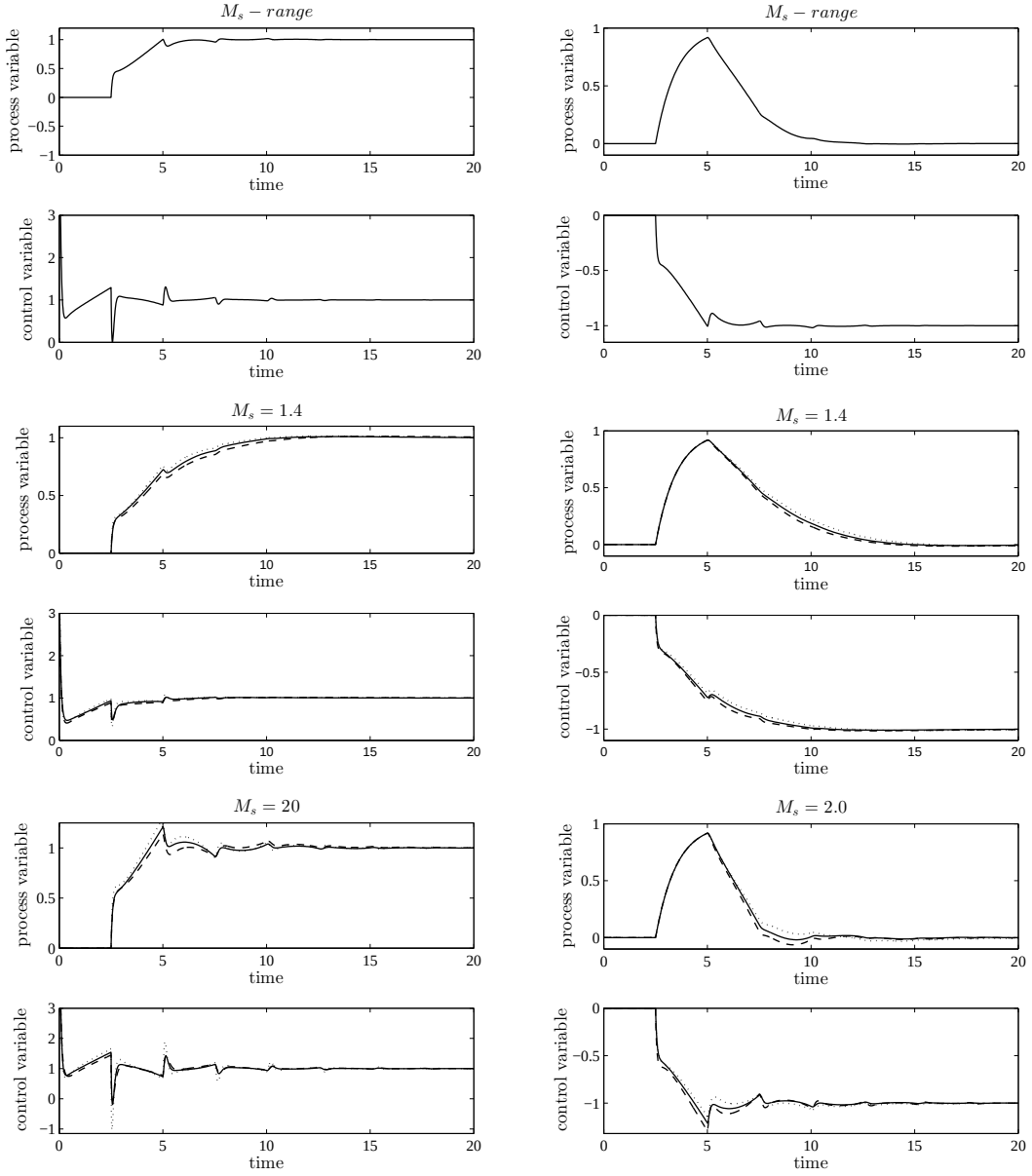


FIGURE 9.8: Set-point and load disturbance step responses for $P_5(s)$. Solid line: proposed tuning rules for FOPID controllers. Dash-dot line: tuning rules for PID controllers proposed in [4]. Dashed line: tuning rules for FOPID controllers proposed in [74]. Dotted line: tuning rules for FOPID controllers proposed in [74] used for the other control task they have been devised.

Tuning rule	GPI
M_s-*range*	0.61
SP 1.4 F	0.66
SP 2.0 F	0.72
LD 1.4 F	0.66
LD 2.0 F	0.86

TABLE 9.14: Performance index for each tuning of $P_5(s)$

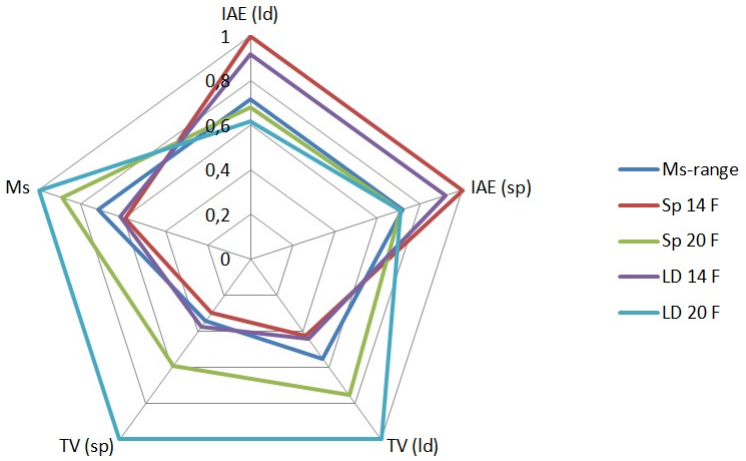
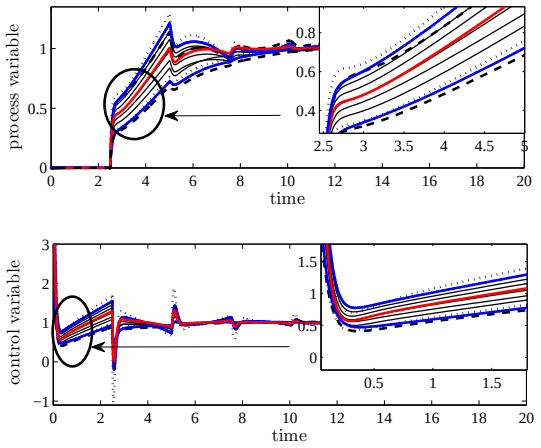
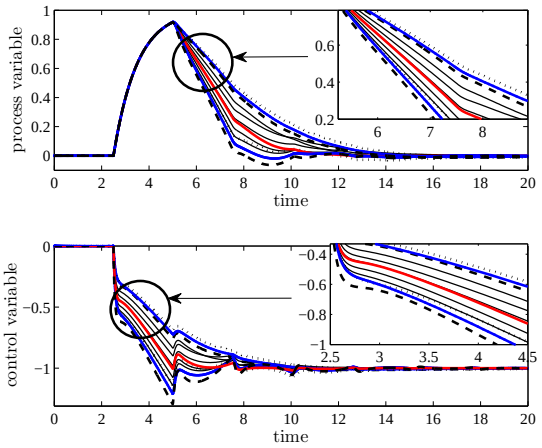


FIGURE 9.9: Radar diagram for $P_5(s)$



(a)



(b)

FIGURE 9.10: Step responses for $P_5(s)$. Comparison between the tuning rules for M_s-*range* and M_s-*value* case and the tuning rules proposed in [74]. (a) Set-point following task. (b) Load disturbance rejection task. Solid red line: M_s-*range* tuning. Solid blue lines: boundaries of the M_s-*value* tuning. Dashed line: optimal tuning rules proposed in [74]. Dotted line: response obtained using the tuning proposed in [74] for opposite operation mode.

From the obtained results it can be appreciated that the proposed approach is effective and that the obtained results are very close and sometimes better in both the objectives compared to the optimal ones achieved in [74]. Nevertheless, from the comparison with the tuning rules specifically devised in [74] for a single task (dashed line) and with those that are obtained by inverting the use of them, the

Tuning rules	K_p	T_i	T_d	μ	λ	IAE_{sp}	IAE_{ld}	TV_{sp}	TV_{ld}	Ms
M_s-*range*	0.73	4.87	1.46	1.09	1	7.42	6.98	9.81	1.16	1.68
M_s=1.4	0.49	4.17	1.53	1.06	1	8.91	8.79	6.18	1.04	1.42
M_s=2.0	0.88	4.83	1.55	1.07	1	7.29	6.36	11.31	1.38	1.98
SP 1.4 F	0.34	2.96	2.43	1.2	1	9.25	9.20	17.10	1.02	1.40
SP 2.0 F	0.54	3.15	2.97	1.2	1	6.88	6.48	26.78	1.14	2.00
LD 1.4 F	0.33	2.63	2.53	1.2	1	8.95	8.81	16.56	1.06	1.44
LD 2.0 F	0.59	3.07	2.62	1.2	1	7.28	6.40	29.50	1.35	2.00
$\alpha = 0.50$	0.91	4.16	2.20	1	1	6.91	5.90	9.80	1.49	2.92

TABLE 9.15: Results related to the high-order process $P_6(s)$.

balancing of the tuning between the two tasks can be clearly seen.

9.6.2 High-order process

As another example, in order to verify the robustness of the proposed rules, the following high-order process has been considered:

$$P_6(s) = \frac{1}{(s+1)^8}. \tag{9.23}$$

In order to apply the tuning rules originally devised for FOPID controller, the process has been modelled as a FOPDT process with $K = 1$, $T = 3.06$, $L = 4.95$ and $\tau = 1.62$ (note that the process is therefore dead time dominant). Then, the optimal tuning rules have been applied. The results obtained are shown in Table 9.15 and in Figure 9.11. As for the previous process, the radar diagram and the global performance index for the different considered tuning rules have been determined. They are shown in Figure 9.12 and in Table 9.16, respectively. From the obtained results it can be appreciated that the proposed approach gives satisfactory performance and the sensitivity values are acceptable

9.6.3 Non-minimum-phase process

Finally, a non-minimum-phase process is considered:

$$P_7(s) = \frac{1-s}{(s+1)^3}. \tag{9.24}$$

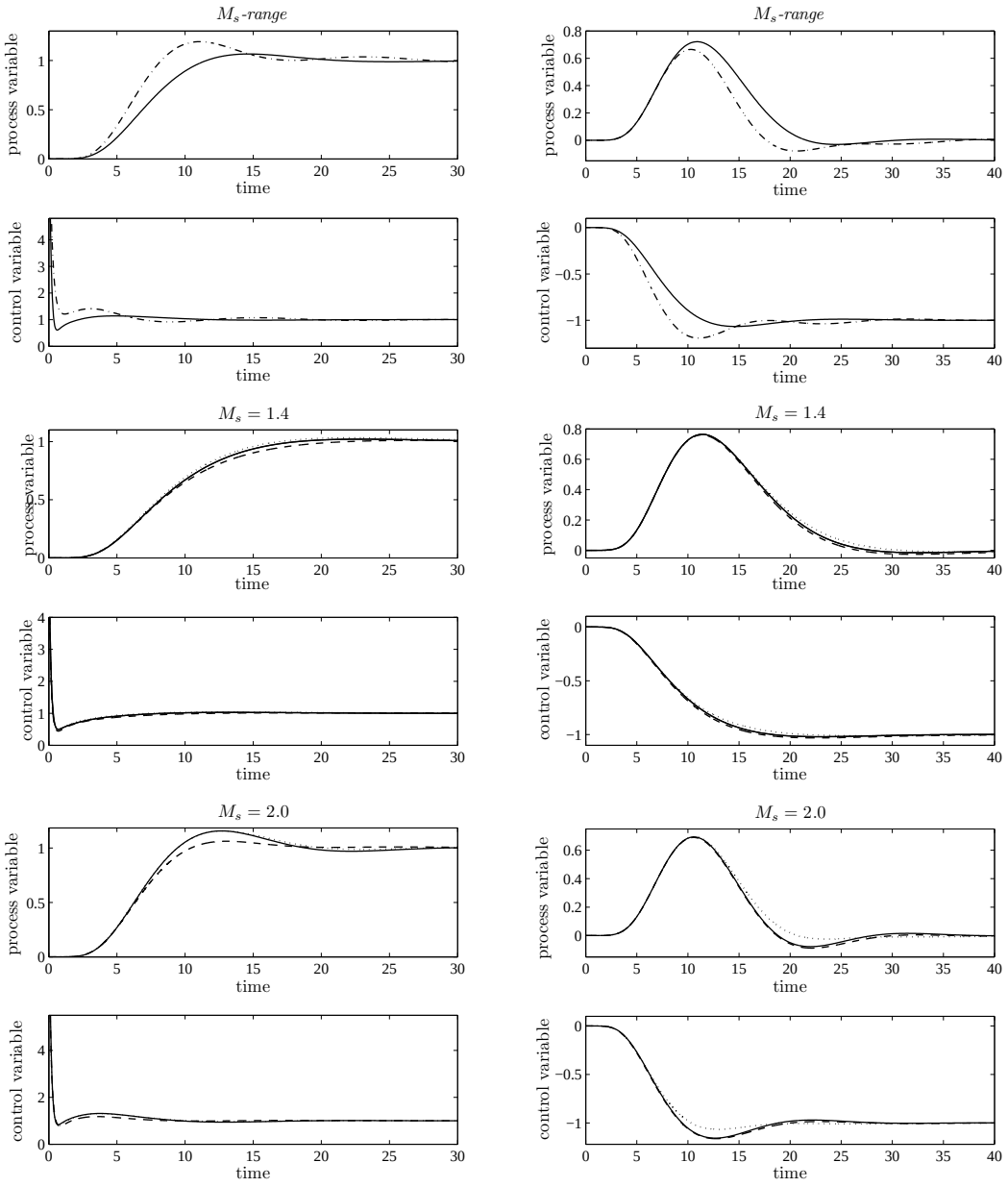


FIGURE 9.11: Set-point and load disturbance step responses for $P_6(s)$. Solid line: proposed tuning rules for FOPID controllers. Dash-dot line: tuning rules for PID controllers proposed in [4]. Dashed line: tuning rules for FOPID controllers proposed in [74]. Dotted line: tuning rules for FOPID controllers proposed in [74] used for the other control task they have been devised.

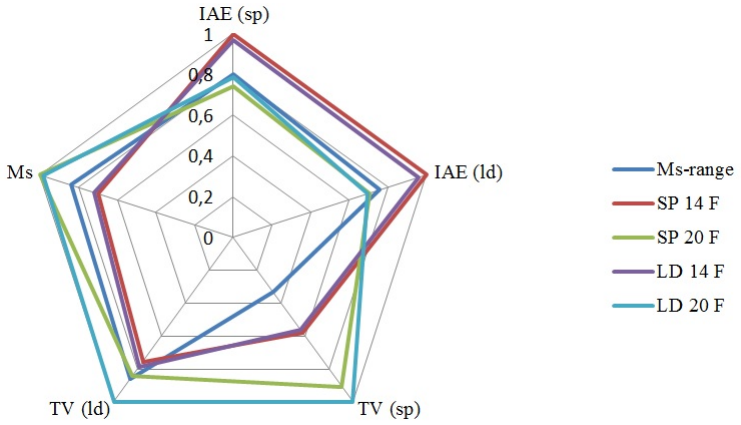


FIGURE 9.12: Radar diagram for the high-order process $P_6(s)$

Tuning rule	GPI
M_s-*range*	0.72
SP 1.4 F	0.81
SP 2.0 F	0.84
LD 1.4 F	0.80
LD 2.0 F	0.89

TABLE 9.16: Performance index for each tuning for process $P_6(s)$.

Tuning rules	K_p	T_i	T_d	μ	λ	IAE_{sp}	IAE_{ld}	TV_{sp}	TV_{ld}	Ms
M_s-*range*	0.78	2.47	0.68	1.10	1	3.34	3.45	9.91	1.39	1.68
M_s=1.4	0.52	2.13	0.72	1.07	1	4.29	4.46	6.42	1.26	1.41
M_s=2.0	0.93	2.43	0.73	1.08	1	3.04	3.00	11.13	1.55	1.97
SP 1.4 F	0.37	1.57	1.01	1.2	1	4.46	4.66	16.11	1.24	1.39
SP 2.0 F	0.59	1.67	1.24	1.2	1	3.15	3.32	25.01	1.47	2.05
LD 1.4 F	0.35	1.33	1.10	1.2	1	4.32	4.45	15.16	1.30	1.43
LD 2.0 F	0.61	1.51	1.10	1.1	1	3.09	2.91	11.86	1.61	2.16
$\alpha = 0.50$	0.97	2.09	1.09	1	1	2.78	2.87	10.60	1.85	2.90

TABLE 9.17: Results related to the non-minimum-phase process $P_7(s)$.

Again, the process has been modeled as a FOPDT process with $K = 1$, $T = 1.62$, $L = 2.39$ and $\tau = 1.48$. Then, the optimal tuning rules have been applied. Results are shown in Table 9.17, the responses are plotted in Figure 9.13, the radar diagram is in Figure 9.14 and the global performance indices are in Table 9.18.

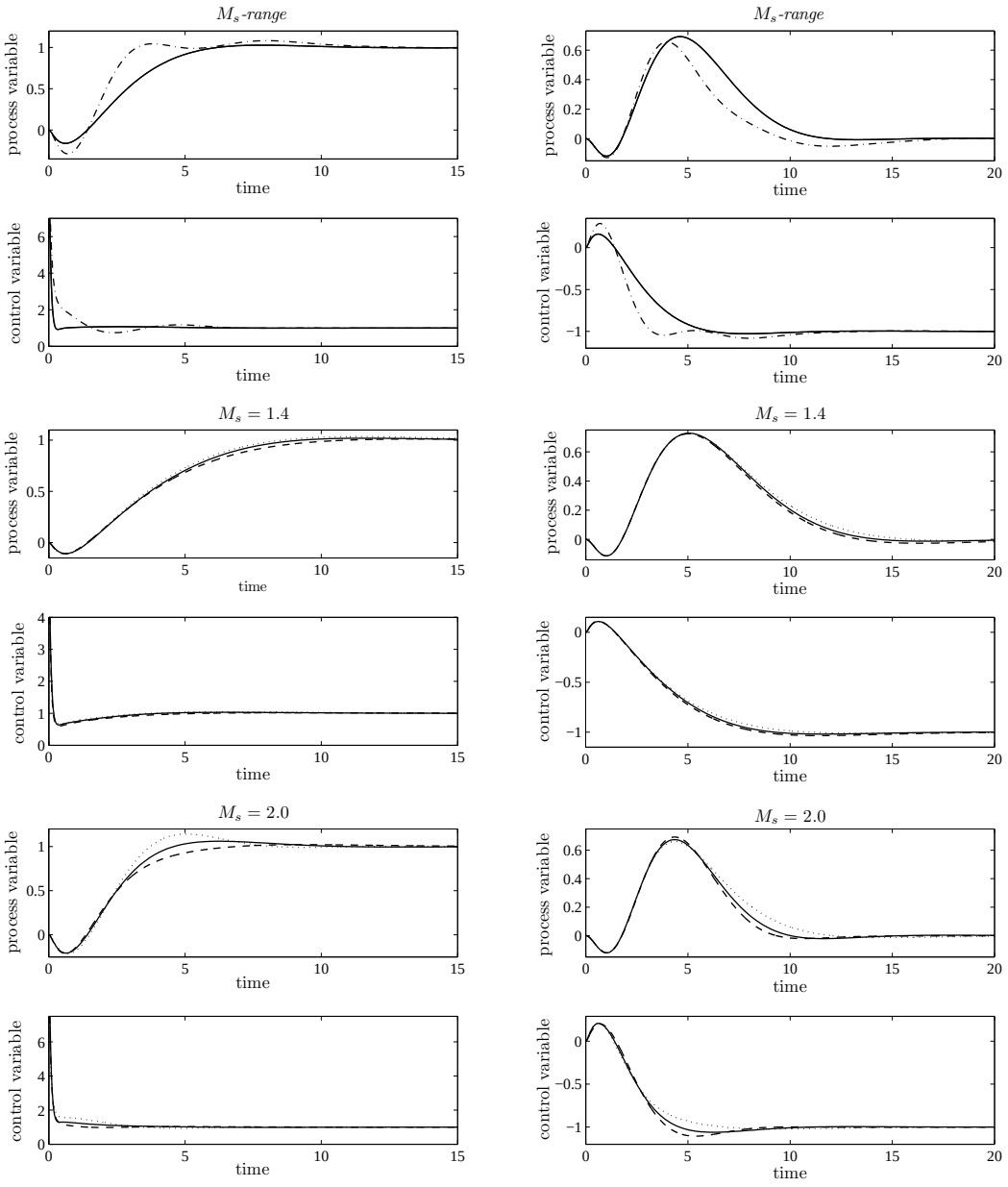


FIGURE 9.13: Set-point and load disturbance step responses for $P_7(s)$. Solid line: proposed tuning rules for FOPID controllers. Dash-dot line: tuning rules for PID controllers proposed in [4]. Dashed line: tuning rules for FOPID controllers proposed in [74]. Dotted line: tuning rules for FOPID controllers proposed in [74] used for the other control task they have been devised.

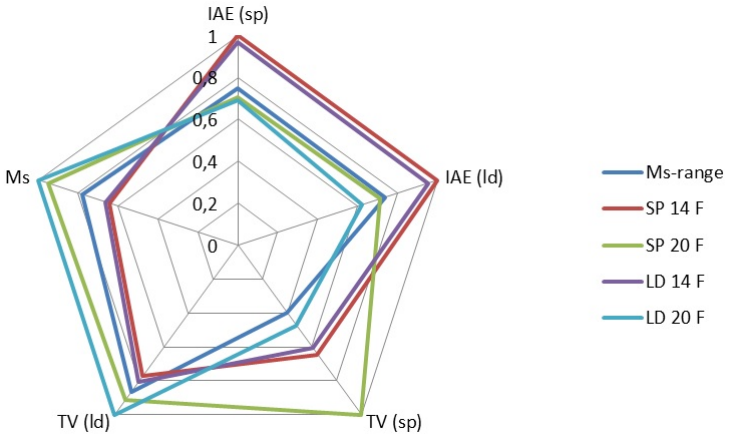


FIGURE 9.14: Radar diagram for the non-minimum-phase process $P_7(s)$

Tuning rule	GPI
M_s-*range*	0.71
SP 1.4 F	0.81
SP 2.0 F	0.86
LD 1.4 F	0.80
LD 2.0 F	0.76

TABLE 9.18: Performance index for each tuning for process $P_7(s)$

9.6.4 Discussion

According to the simulation results, it can be seen that the intermediate tuning rules for PID controllers proposed in [4] achieve the best IAE performance for all the simulated processes for the load disturbance rejection and also the best IAE performance for set-point step response for $P_7(s)$. However, this comes at the expense of a higher value of the maximum sensitivity and therefore of a higher control action (indeed, as already mentioned, the maximum sensitivity has not been taken into account in the development of the intermediate tuning rules proposed in [4]), as it can be observed by looking at the step responses. On the contrary, the proposed tuning rules for the M_s-*range* case ensure a medium-high robustness level ($M_s \approx 1.7$) and also a smoother response for both operation modes (less oscillations in the output response and a smoother control variable) in all the examples.

For the cases of $M_s = 1.4$ and $M_s = 2.0$, it can be noticed that using the tuning rules for the M_s-*value* case, in all the processes the set-point step response shows an overshoot, but not significantly higher than the one obtained with the rules for FOPID controllers proposed in [74]. However, in the load disturbance step

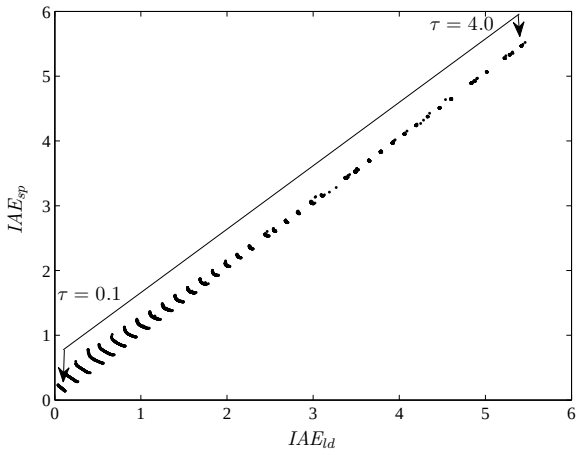


FIGURE 9.15: The Pareto front for the M_s-*valued* case, $M_s = 1.7$.

response, the rules for the M_s-*value* cases provide less oscillations in comparison with the rules proposed in [74]. Furthermore, it can be observed that, for $P_6(s)$ and $P_7(s)$, the optimal values of each operation mode with the rules in [74] are similar to those obtained with the M_s-*value* rules.

In general, it appears that the proposed rules are effective, especially considering their simplicity: indeed, the user is not forced to decide which task (servo or regulatory) is more important. Indeed, both operation modes are balanced in an optimal way by choosing the Nash solution.

It is also interesting to note that the value of M_s obtained by using the M_s-*range* tuning is quite close to 1.7, especially for processes with $\tau > 1$ (see Figure 9.3). Indeed, the NS in the M_s-*range* case is optimally balanced between three objectives, namely the set-point following, the load disturbance rejection and the robustness. By considering the results in Figure 9.4, as well as the simulation results, it turns out that there is no point in further increasing the value of M_s after 1.7 since the performance improvements are minimal in spite of the loss of robustness. As a final consideration, it is interesting to note that, for the normalized dead time greater than 2 ($\tau = 2$) the Pareto front curve (surface, in the M_s-*range* case) approximation tends to collapse into a single point (curve, in the M_s-*range* case) as it can be observed in Figure 9.15, where only the case M_s=1.7 is considered for the sake of readability (but the other cases lead to similar results). Hence, the trade-off between load disturbance rejection and set-point following task is no longer appreciable for high delays and the optimal tuning for pure load disturbance rejection tends to coincide with the one for pure set-point following.

9.7 Summary

A set of optimally balanced tuning rules for FOPID controllers has been presented in this paper. Based on a MOOD procedure, the problem of tuning the FOPID controller with several requirements is addressed. Such procedure implemented a multi-stage approach into a MOO process in order to improve the convergence properties for the Pareto front approximation. The primary goal of this work has been to minimize the IAE for either the load disturbance rejection task and the set-point following task with a constraint on the maximum sensitivity.
The obtained rules have the valuable features to be able to take into account at once both the servo and the regulatory modes in an optimal way. Moreover, the user can select the desired level of robustness or keep it between given bounds depending on his/her preferences.
The performance assessment for both cases has been presented. This allows the designer to know in advance the performance index he/she will obtain and evaluate if the performance obtained with an existing controller can be improved.

Part IV

Conclusions and perspectives

Chapter 10

Conclusions and perspectives

It is worth stressing to mention that the controller design is not a simple task. The designer should consider the controlled process dynamic characteristics, the stability of the system, the operation modes, the performance requirements and others. The main objective of this thesis is to provide a methodology to addresses controller tuning, involving different trade-off in order to improve the performance of the control system. The contributions of this thesis were commented in Chapter 1, and a brief summary have been provide in each chapter. This chapter summarizes the main conclusions and some perspective for the future work.

10.1 Conclusions

This thesis deals with controller tuning problems by implementing the Multi-objective Optimization Design (MOOD) procedure [88]. As it was mentioned before, this procedure consist in three steps:

- Multi-objective Problem (MOP): definition of the problem (decision space, variables and constraints).
- Multi-objective Optimization (MOO) process: optimizer to execute (the designer looks for desirable characteristic)
- Multi-criteria decision making (MCDM) stage: selection of one point from the Pareto front approximation according to the designer preferences.

This procedure have been implemented along all the development of this work, first because offers a good balance between the competitive objectives and helps the designer to analyze in a better way different trade-off that the controller faces.

The thesis has been divided in three parts. The first part, presented the fundamentals of the control system showing and discussing the different trade-offs between performance/robustness and servo/regulation operation modes. On the other hand a background on MOO has been provided. It has been pointed out, that the controller design have a muti-objective nature. Hereafter, an optimizer capable of searching for the optimal space in order to obtain a reliable Pareto set approximated was selected a deterministic algorithm, the Normalized Normal Constraint (NNC) algorithm for the MOO process, which generate a well distributed set of solutions even in numerically demanding situations.

In the second part, Chapter 4 introduces the Nash solution as a MCDM technique, to select a point from the Pareto front that represent the best compromise among the design objective. This solution provides a semi-automatic selection from the Pareto front approximation and offers a good trade-off between the goal objectives. Hereafter, in Chapter 5 a Multi-stage approach for the MOO process is presented. This approach involves two algorithms: the NNC and the a Multi-objective Differential Evolution algorithm with Spherical Pruning (sp-MODE). The first one in charge of searching for the convergence area (local search) and the other it is in charge of the global search. This approach allows both algorithms to complement each other in despite of their drawbacks and improve the results of the overall optimization in terms of convergence and accuracy. Further, the introduction of reliability based objective into the MOP is carried out, to measure the performance degradation. It is worth while to mention that, due to the existence of uncertainties in real-world designing and manufacturing having this design objective will give another perspective to the designer. The idea of using this objective is to minimize the deviation from the achieved nominal performance, by using the Montecarlo sampling approach. Nevertheless, the Multi-stage approach will reduce the numerical burden associated with the generation of the Pareto front approximations for reliability problems. In order to validate the approach, in Chapter 6 two different case studies has been considered, the Boiler control problem for controller tuning and as second case, a non-linear Peltier Cell. In both cases a MOOD procedure using a Multi-stage approach for the MOO process and a reliability-based optimization design for controller tuning have been applied, as

it can be see it for the result this approach improve the convergence properties and reliable solutions for the Pareto Set approximation.

In the third part of this thesis, the application to the controller tuning have been presented. First, a set of tuning rules based on the NS for a 2DoF proportional-integral (PI) controller have been devised, where the robustness/performance trade-off have been considered. For this case, NNC algorithm for the MOO process was implemented. Moreover, as a second case it is presented a tuning for 1DOF proportional-integral-derivative (PID) controller where the trade-off of the performance/robustness and servo/regulation operation mode have been considered. The tuning rules proposed for PI and PID controllers offers a range of maximum sensitivity Ms from $[1.4, 2.0]$.

Finally in Chapter 9, in order to find the best parameters for the fractional-order-PID (FOPID) controller, the MOOD procedura have been applied by using the Multi-stage approach for the MOO process. The obtained rules have the valuable features to be able to take into account at once both the servo and the regulatory modes in an optimal way. Moreover, the user can select the desired level of robustness or keep it between given bounds depending on his/her preferences. Further, the performance assessment for both cases has been presented. This allows the designer to know in advance the performance index he/she will obtain and evaluate if the performance obtained with an existing controller can be improved.

An important aspect of this devised tuning rules for PI, PID and FOPID controllers is that all of them were parameterized using the same form for the equations of the controllers parameters, this allows to keep simple expressions that can be easy to implemented for the users.

10.2 Perspective

There are always improvements or extensions which can be addressed to continue working. Some ideas are detailed below:

- Applying the Multi-stage approach on MIMO system as has been seen, this work has concentrated on SISO systems and the design for the multivariable case it will be an interesting challenging.

- The devised tuning rules for PI and PID controllers are constrained to an interval of robustness level $[1.4, 2.0]$. It will be interesting to obtain tuning rules for each value of this interval, meaning that the designer can choose which level of robustness he/she requires.

- In Chapter 7, the designing of the PI controller tuning rules concentrates on the load-disturbance performance; another extension work that it can be done is to consider the set-point performance.

- Extension to FOPID controller for integrator-plus-dead-time (IPDT) processes. Generation of tuning rules based on IPDT processes.

- Implementation of the MOOD procedure, in order to improve the performance of the control loops in mechatronic systems considering the multidisciplinary nature of them.

- Extension to feedforward feedback scheme. Development simultaneous tuning rules for both controllers, feedback and feedforward, but a sequential one.

Bibliography

[1] V. M. Alfaro. Low-Order Models Identification from the Process Reaction Curve. *Ciencia y Tecnología (Costa Rica)*, 24(2):197–216, 2006. (in Spanish).

[2] V. M. Alfaro and R. Vilanova. Model-reference robust tuning of 2DoF PI controllers for first- and second-order plus dead-time controlled processes. *Journal of Process Control*, 22(2):359–374, 2012.

[3] K. H. Ang, G. Chong, and Y. Li. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technologies*, 13(4):559–576, 2005.

[4] O. Arrieta, A. Visioli, and R. Vilanova. PID autotuning for weighted servo/regulation control operation. *Journal of Process Control*, 20(4):472–480, 2010.

[5] J. Arévalo. Gradual nash bargaining with endogenous agenda: a path-dependent model. *Colombian Economic Journal*, 2:189–212, 2004.

[6] K. J. Åström and T. Hägglund. *PID controllers: Theory, Design and Tuning*. ISA, Research Triangle Park, NJ, 1995.

[7] K. J. Åström and T. Hägglund. *Advanced PID Control*. ISA Press, Research Triangle Park, NJ, 2006.

[8] K .J. Åström and T. Hägglund. Benchmark Systems for PID Control. In *IFAC Digital Control: Past, Present and Future of PID Control (PID'00)*, 5-7 April, Terrasa, Spain, 2000.

[9] K. J. Åström, H. Panagopoulos, and T. Hägglund. Design of PI controllers based on non-convex optimization. *Automatica*, 34(5):585 – 601, 1998.

[10] K.J. Åström and T. Hägglund. Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20(5):645–651, 1984.

[11] R. J. Aumann and S. Hart. *Handbook of Game Theory with Economic Applications*, volume 2. Elsevier, 1994.

[12] R. S. Barbosa, J. A. Tenreiro Machado, and I. M. Ferreira. Tuning of PID controllers based on Bode's ideal transfer function. *Nonlinear Dynamics*, 38:305–321, 2004.

[13] R. Bell and K. J. Åström. Dynamic Models for Boiler-Turbine Alternator Units: Data Logs and Parameter Estimation for a 160 MW unit". Technical Report ISRN LUTFD2/TFRT--3192--SE,, Department of Automatic Control, Lund University, Sweden, jun 1987.

[14] M. Beschi, F. Padula, and A. Visioli. The generalised isodamping approach for robust fractional pid controllers design. *International Journal of Control*, in press:369–381, 2016.

[15] H. G. Beyer and B. Sendhoff. Robust optimization - A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190 – 3218, 2007.

[16] X. Blasco, J.M. Herrero, J. Sanchis, and M. Martínez. A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908–3924, 2008.

[17] X. Blasco, J.M. Herrero, J. Sanchis, and M. Martínez. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908 – 3924, 2008.

[18] R. Caponetto, G. Dongola, L. Fortuna, and A. Gallo. New results on the synthesis of FO-PID controllers. *Communications in Nonlinear Science and Numerical Simulation*, 15(4):997–1007, 2010.

[19] F. J. Castillo, V. Feliu, R. Rivas, and L. Sanchez. Comparative analysis of stability and robustness between integer and fractional-order PI controllers for first order plus dead time plants. In *Proceedings of the 18th IFAC World Congress*, 2011.

[20] R. Cela and M.H. Bollaín. New cluster mapping tools for the graphical assessment of non-dominated solutions in multi-objective optimization. *Chemometrics and Intelligent Laboratory Systems*, 114:72–86, 2012.

[21] Y. Q. Chen. Ubiquitous fractional order controls? In *Proceedings of the 4th IFAC Workshop of Fractional Differentiation and its Applications*, Porto (P), 2006.

[22] Y. Q. Chen, T. Bhaskaran, and D. Xue. Practical tuning rule development for fractional order proportional and integral controllers. *ASME Journal of Computational and Nonlinear Dynamics*, 3:0214031–0214037, 2008.

[23] Y. Q. Chen, I. Petras, and D. Xue. Fractional order control - a tutorial. In *Proceedings of the American Control Conference*, St. Louis (MO), 2009.

[24] C. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen. Multi-criteria decision making. In *Evolutionary algorithms for solving multi-objective problems, genetic and evolutionary computation series*, pages 515–545, New York (NY), 2007. Springer.

[25] S. Das and P. N. Suganthan. Differential Evolution: A Survey of the State-of the-Art. *Evolutionary Computation, IEEE Transactions on*, 15(1):4–31, 2011.

[26] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.

[27] A.R. de Freitas, P.J. Fleming, and F.G. Guimarães. Aggregation trees for visualization and dimension reduction in many-objective optimization. *Information Sciences*, 298:288–314, 2015.

[28] J. Derrac, S. García, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3 – 18, 2011.

[29] I. Fernández, C. Rodríguez, J. L. Guzmán, and M. Berenguel. Control predictivo por desacoplo con compensación de perturbaciones para el benchmark de control 2009–2010. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(2):112–121, 2011.

[30] J. Figueira, S. Greco, and M. Ehrgott. *Multiple criteria decision analysis: state of the art surveys*, volume 78. Springer Science & Business Media, 2005.

[31] M. W. Foley, N. R. Ramharack, and B. R. Copeland. Comparison of pi controller tuning methods. *Industrial & engineering chemistry research*, 44(17):6741–6750, 2005.

[32] C. M. Fonseca and P.J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 28(1):26–37, 1998.

[33] C. M. Fransson, T. Wik, B. Lennartson, M. Saunders, and P. O. Gutman. Nonconservative robust control: optimized and constrained sensitivity functions. *Control Systems Technology, IEEE Transactions on*, 17(2):298–308, 2009.

[34] Q. Gao, J. Chen, L. Wang, S. Xu, and Y. Hou. Multiobjective optimization design of a fractional order PID controller for a gun control system. *The Scientific World Journal*, 2013, 2013.

[35] O. Garpinger and T. Hägglund. Performance and robustness trade-offs in PID control. *Journal of Process Control*, 24:568–577, 2014.

[36] J. Garrido, F. Vázquez, and F. Morilla. Multivariable PID control by inverted decoupling: application to the benchmark PID 2012. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)*, 2012.

[37] M. Ge, M.S. Chiu, and Q.G. Wang. Robust PID controller design via LMI approach. *Journal of process control*, 12(1):3–13, 2002.

[38] E.N. Goncalves, R.M. Palhares, and Takahashi. A novel approach for H_2/H_∞ robust PID. *Journal of process control*, 18(1):19–26, 2008.

[39] J. J. Gude and E. Kahoraho. New tuning rules for PI and fractional PI controllers. In *Proceedings European Control Conference*, Budapest (HU), 2009.

[40] R. E. Gutierrez, J. M. Rosario, and J. A. Tenreiro Machado. Fractional order calculus: Basic concepts and engineering applications. *Mathematical Problems in Engineering*, 2010, 2010.

[41] T. Hägglund and K.J. Åström. Revisiting the ziegler-nichols tuning rules for PI control. *Asian Journal of Control*, 4(4):364–380, 2002.

[42] A. Hajiloo, N. Nariman-Zadeh, and A. Moeini. Pareto optimal robust design of fractional-order PID controllers for systems with probabilistic uncertainties. *Mechatronics*, 22(6):788–801, 2012.

[43] A. Herreros, E. Baeyens, and J. R. Perán. Design of PID-type controllers using multiobjective genetic algorithms. *ISA transactions*, 41(4):457–472, 2002.

[44] W. K. Ho, K. W. Lim, C. C. Hang, and L. Y. Ni. Tuning PID controllers based on gain and phase margin specifications. *Automatica*, 35(9):1579–1585, 1999.

[45] L. Huang, W. Ning, and Z. Jin-Hui. Multiobjective optimization for controller design. *Acta Automatica Sinica*, 34(4):472–477, 2008.

[46] M.-H. Hung, L.-S. Shu, S.-J. Ho, S.-F. Hwang, and S.-Y. Ho. A novel intelligent multiobjective simulated annealing algorithm for designing robust PID controllers. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(2):319–330, 2008.

[47] A. Inselberg. The plane with parallel coordinates. *The visual computer*, 1(2):69–91, 1985.

[48] J. F. Nash Jr. The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.

[49] I. Kaya. Tuning PI controllers for stable processes with specifications on gain and phase margins. *ISA transactions*, 43(2):297–304, 2004.

[50] H. A. Kiam, G. Chong, and L. Yun. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4):559 – 576, 2005.

[51] A. M. Lopez, J. A. Miller, C. L. Smith, and P. W. Murrill. Tuning controllers with error-integral criteria. *Instrumentation Technology*, 1967.

[52] R. H. Lopez, R. Holdorf, and A. T. Beck. Reliability-based design optimization strategies based on FORM: a review. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 34(4):506–514, 2012.

[53] A. Lotov and K. Miettinen. Visualizing the Pareto frontier. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors, *Multiobjective Optimization*, volume 5252 of *Lecture Notes in Computer Science*, pages 213–243. Springer Berlin/Heidelberg, 2008.

[54] R. Malti, X. Moreau, F. Khemane, and A. Oustaloup. Stability and resonance conditions of elementary fractional transfer functions. *Automatica*, 47(11):2462–2467, 2011.

[55] G. A. Mannella, V. La Carrubba, and V. Brucato. Peltier cells as temperature control elements: Experimental characterization and modeling. *Applied thermal engineering*, 63(1):234–245, 2014.

[56] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[57] M. A. Martínez, J. Sanchis, and X. Blasco. Multiobjective controller design handling human preferences. *Engineering applications of artificial intelligence*, 19(8):927–938, 2006.

[58] S. Masami, O. Kyosuke, and W. Nobutka. Application of data-driven loop-shaping method to multi-loop control design of benchmark PID 2012. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)*, March 2012.

[59] C. A. Mattson and A. Messac. Pareto frontier based concept selection under uncertainty, with visualization. *Optimization and Engineering*, 6(1):85–115, 2005.

[60] L. Meng and D. Xue. Design of an optimal fractional-order PID controller using multi-objective GA optimization. In *Proceedings Chinese Control and Decision Conference*, pages 3849–3853. IEEE, 2009.

[61] A. Messac, A. Ismail-Yahaya, and C. A. Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25(2):86–98, 2003.

[62] K. M. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, 1998.

[63] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu. *Fractional-order Systems and Controls*. Springer-Verlag, London (UK), 2010.

[64] C. A. Monje, B. M. Vinagre, A. J. Calderon, V. Feliu, and Y. Q. Chen. On fractional PI$^{\lambda}$ controllers: some tuning rules for robustness to plant uncertainties. *Nonlinear Dynamics*, 38:369–381, 2004.

[65] C. A. Monje, B. M. Vinagre, V. Feliu, and Y. Q. Chen. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Engineering Practice*, 16(7):798–812, 2008.

[66] F. Morilla. Benchmark 2009-10 grupo temático de ingeniería de control de cea-ifac: Control de una caldera. Avalaible at www.cea-ifac.es/w3grupos/ingcontrol, Febrero 2010.

[67] F. Morilla. Benchmark for PID control based on the boiler control problem, 2012. Internal report, UNED Spain.

[68] E. Nikmanesh, O. Hariri, H. Shams, and M. Fasihozaman. Pareto design of Load Frequency Control for interconnected power systems based on multi-objective uniform diversity genetic algorithm (MUGA). *International Journal of Electrical Power & Energy Systems*, 80:333–346, 2016.

[69] A. O'Dwyer. *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press, London, UK, 2nd. edition, 2006.

[70] M. D. Ortigueira. *Fractional Calculus for Scientists and Engineers*. Springer, 2011.

[71] A. Oustaloup. *La Commande CRONE: Commande Robuste d'Ordre Non Entier*. Hermes, Paris (F), 1991.

[72] A. Oustaloup, P. Lanusse, P. Melchior, X. Moreau, and J. Sabatier. The CRONE approach: theoretical developments and major applications. In *Proceedings of the 2nd IFAC Workshop on Fractional Differentiation and its Applications*, Porto (P), 2006.

[73] A. Oustaloup, J. Sabatier, P. Lanusse, R. Malti, P. Melchior, X. Moreau, and M. Moze. An overview of the crone approach in system analysis, modeling and identification, observation and control. In *Proceedings of the 17th IFAC World Congress*, Seoul (Korea), 2008.

[74] F. Padula and A. Visioli. Tuning rules for optimal PID and fractional-order PID controllers. *Journal of Process Control*, 21(1):69–81, 2011.

[75] F. Padula and A. Visioli. Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes. *IET Control Theory and Applications*, 6(6):776–786, 2012.

[76] F. Padula and A. Visioli. *Advances in Robust Fractional Control*. Springer, London (UK), 2014.

[77] I. Pan and S. Das. Chaotic multi-objective optimization based design of fractional order $PI^{\lambda}D^{\mu}$ controller in AVR system. *International Journal of Electrical Power & Energy Systems*, 43(1):393–407, 2012.

[78] H. Panagopoulos, K. J. Åström, and T. Hägglund. Design of PID controllers based on constrained optimisation. *IEE Proceedings-Control Theory and Applications*, 149(1):32–40, 2002.

[79] G. Pellegrinetti and J. Bentsman. Nonlinear control oriented boiler modeling-a benchmark problem for controller design. *Control Systems Technology, IEEE Transactions on*, 4(1):57–64, 1996.

[80] I. Podlubny. Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers. *IEEE Transactions on Automatic Control*, 44(1):208–214, 1999.

[81] G. Reynoso-Meza, X. Blasco, and J. Sanchis. Diseño multiobjetivo de controladores PID para el benchmark de control 2008–2009. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 6(4):93–103, 2009.

[82] G. Reynoso-Meza, X. Blasco, J. Sanchis, and J. M. Herrero. Comparison of design concepts in multi-criteria decision-making using level diagrams. *Information Sciences*, 221:124 – 141, 2013.

[83] G. Reynoso-Meza, X. Blasco, J. Sanchis, and J.M. Herrero. Comparison of design concepts in multi-criteria decision-making using level diagrams. *Information Sciences*, 221:124–141, 2013.

[84] G. Reynoso-Meza, S. García-Nieto, J. Sanchis, and X. Blasco. Controller tuning using multiobjective optimization algorithms: a global tuning framework. *IEEE Transactions on Control Systems Technology*, 21(2):445–458, 2013.

[85] G. Reynoso-Meza, J. Sanchis, X. Blasco, and R.Z Freire. Evolutionary multi-objective optimisation with preferences for multivariable pi controller tuning. *Expert Systems with Applications*, 2015.

[86] G. Reynoso-Meza, J. Sanchis, X. Blasco, and S. García-Nieto. Multiobjective evolutionary algortihms for multivariable PI controller tuning. *Appied Soft Computing*, 24:341 – 362, 2014.

[87] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez. Multiobjective design of continuous controllers using differential evolution and spherical pruning. In *Applications of Evolutionary Computation, Part I*, volume LNCS 6024, pages 532–541. Springer-Verlag, 2010.

[88] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez. Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice*, 28:58–73, 2014.

[89] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez. Preference driven multi-objective optimization design procedure for industrial controller tuning. *Information Sciences*, 2015.

[90] J. D. Rojas, F. Morilla, and R. Vilanova. Multivariable PI control for a boiler plant benchmark using the virtual reference feedback tuning. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)*, March 2012.

[91] D. Roos, U. Adam, and C. Bucher. Robust design optimization. *Proceedings. Weimarer Optimierung-und Stochastiktage*, 3, 2006.

[92] A. A. Rovira, P. W. Murrill, and C. L. Smith. Tuning controllers for setpoint changes. Technical report, DTIC Document, 1970.

[93] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*, volume 707. John Wiley & Sons, 2011.

[94] J. Sabatier, O. P. Agrawal, and J. A. Tenreiro Machado. *Advances in Fractional Calculus: Theoretical Developments and Applications in Physics and Engineering*. Springer, London, (UK), 2007.

[95] H. Sánchez, G. Reynoso-Meza, R. Vilanova, and X. Blasco. Comparación de técnicas de optimización multi-objetivo clásicas y estocásticas para el ajuste de controladores PI (spanish). In *XXXIV Jornadas de Automática*, Barcelona, Terrassa, Spain, 2013.

[96] H. Sánchez, G. Reynoso-Meza, R. Vilanova, and X. Blasco. Multistage procedure for pi controller design of the boiler benchmark problem. In *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*, pages 1–4, Luxembourg, 2015. IEEE.

[97] H. S. Sánchez and R. Vilanova. Multiobjective tuning of PI controller using the NNC method: Simplified problem definition and guidelines for decision making. In *Emerging Technologies & Factory Automation(ETFA), 2013 IEEE 18th Conference on*, Cagliary, Italy, 2013.

[98] H. S. Sánchez and R. Vilanova. Nash-based criteria for selection of pareto optimal pi controller. In *System Theory, Control and Computing (ICSTCC), 2013 17th International Conference*, pages 465–472, Sinaia, Romania, 2013. IEEE.

[99] H. S. Sánchez and R. Vilanova. Optimality comparison of 2DoF PID implementations. In *System Theory, Control and Computing (ICSTCC), 2014 18th International Conference*, pages 591–596, Sinaia, Romania, 2014. IEEE.

[100] H. S. Sánchez, A. Visioli, and R. Vilanova. Nash tuning for optimal balance of the Servo/Regulation operation in robust PID control. In *Mediterranean Conference on Control & Automation*, Málaga, Torremolinos, Spain, 2015.

[101] C. Scali and D. Semino. Performance of optimal and standard controllers for disturbance rejection in industrial processes. In *Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON'91., 1991 International Conference on*, pages 2033–2038. IEEE, 1991.

[102] F. G. Shinskey. *Feedback controllers for the process industries*. McGraw-Hill Professional, 1994.

[103] F.G. Shinskey. Process control: as taught vs as practiced. *Industrial & engineering chemistry research*, 41(16):3745–3750, 2002.

[104] A. Silveira, A. Coelho, and F. Gomes. Model-free adaptive PID controllers applied to the benchmark PID12. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)*, March 2012.

[105] S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Modeling, Identification and Control*, 25(2):85–120, 2004.

[106] S. Skogestad and C. Grimholt. The SIMC method for smooth PID controller tuning. In *PID Control in the Third Millennium*, pages 147–175. Springer, 2012.

[107] R. F. Stengel and C.I. Marrison. Robustness of solutions to a benchmark control problem. In *American Control Conference*, pages 1915–1916. IEEE, 1991.

[108] G. Stewart and T. Samad. Cross-application perspectives: Application and market requirements. *The Impact of Control Technology*, pages 95–100, 2011.

[109] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[110] G. Szita and C.K. Sanathanan. Model matching controller design for disturbance rejection. *Journal of the Franklin Institute*, 333(5):747–772, 1996.

[111] S. Tavakoli, I. Griffin, and P.J. Fleming. Robust PI controller for load disturbance rejection and setpoint regulation. In *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, pages 1015–1020. IEEE, 2005.

[112] S. Tavakoli, I. Griffin, and P.J. Fleming. Multi-objective optimization approach to the PI tuning problem. In *Proceedings of the IEEE congress on evolutionary computation (CEC2007)*, pages 3165 – 3171, 2007.

[113] S. Tavakoli and M. Tavakoli. Optimal tuning of PID controllers for first order plus time delay models using dimensional analysis. In *Control and Automation, 2003. ICCA'03. Proceedings. 4th International Conference on*, pages 942–946. IEEE, 2003.

[114] C. Tianyou, J. Yaochu, and S. Bernhard. Evolutionary complex engineering optimization: opportunities and challenges. *Computational Intelligence Magazine*, 8(3):12 – 15, 2013.

[115] T. Ting, X.S. Yang, S. Cheng, and K. Huang. Hybrid Metaheuristic Algorithms: Past, Present, and Future. In *Recent Advances in Swarm Intelligence and Evolutionary Computation*, pages 71–83. Springer, 2015.

[116] R. Toscano. A simple robust PI/PID controller design via numerical optimization approach. *Journal of process control*, 15(1):81 – 88, 2005.

[117] T. Tusar and B. Filipic. Visualization of pareto front approximations in evolutionary multiobjective optimization: a critical review and the prosection method. *Evolutionary Computation, IEEE Transactions on*, 19(2):225–245, 2015.

[118] D. Valerio and J. Sá da Costa. Tuning of fractional PID controllers with Ziegler-Nichols-type rules. *Signal Processing*, 86:2771–2784, 2006.

[119] D. Valerio and J. Sá da Costa. A review of tuning methods for fractional PIDs. In *Preprints IFAC Workshop on Fractional Differentiation and its Applications*, Badajoz (E), 2010.

[120] D. Valerio and J. Sá da Costa. Introduction to the single-input, single-output fractional control. *IET Control Theory and Applications*, 5(8):1033–1057, 2011.

[121] R. Vilanova and A. Visioli. *PID control in the third millennium.* Springer, 2012.

[122] B. M. Vinagre, C. A. Monje, A. J. Calderon, and J. I. Suarez. Fractional PID controllers for industry application. a brief introduction. *Journal of Vibration and Control*, 13:1419–1429, 2007.

[123] A. Visioli. *Practical PID Control.* Springer, London (UK), 2006.

[124] O. Yoshimasa. PID controller design for MIMO systems by applying balanced truncation to integral-type optimal servomechanism. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)*, March 2012.

[125] L. Yun, H. A. Kiam, and G. Chong. PID control system analysis and design. *IEEE Control Systems*, 26(1):32 – 41, 2006.

[126] G.-Q. Zeng, J. Chen, Y.-X. Dai, L.-M. Li, C.-W. Zheng, and M.-R. Chen. Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. *Neurocomputing*, 160:173–184, 2015.

[127] M. Zhuang and D. P. Atherton. Automatic tuning of optimum PID controllers. *IEE Proceedings - Control Theory and Applications*, 140:216–224, 1993.

[128] J. Ziegler and N. Nichols. Optimum settings for automatic controllers. *ASME Trans.*, pages 759–768, 1942.

[129] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132, 2003.