

José David Rojas, Orlando Arrieta, Ramon
Vilanova

Industrial PID Controller Tuning

with a multi-objective framework using
MATLAB®

October 23, 2020

Springer

Acknowledgements

This work wouldn't be possible without the hard work of many students along many years. The authors want to acknowledge the following students which were part of our research lab and made the subject of optimization and PID control part of their academic career:

- Macarena Céspedes
- Mónica P. Contreras-Leiva
- Joaquín Cordero
- Carlos Gamboa
- Felipe Moya
- Gustavo Montoya
- Francisco Rivas
- Sergio Rodríguez Rojas
- Rosario Ruiz Hernández
- Felipe Sáenz Cortés
- Karen Valverde
- Diana Valverde-Mendez

Also, a very special thanks goes to our mentor, professor Víctor M. Alfaro which showed us to love the strange art of industrial control.



Contents

1	Introduction	1
	References	3
2	Industrial PID Control	5
2.1	Control System Design Scenario	5
2.2	Industrial Process Characteristics	9
2.2.1	Controlled Process Model	11
2.3	The PID Controller	11
2.3.1	PID Controller formulations	14
2.3.2	Reference Processing and 2-DoF PID	16
2.3.3	Conversion of 2-DoF PID Controller Algorithms	16
2.4	Normalised Representations	17
2.4.1	Process Model Normalisation	17
2.4.2	Controller Normalisation	18
	References	18
3	PID Controller Considerations	21
3.1	Control System Evaluation Metrics	21
3.1.1	Performance	22
3.1.2	Robustness	23
3.1.3	Control Input Usage	24
3.2	Control System Tradeoffs	24
3.2.1	Servo vs. Regulation	25
3.2.2	Performance vs. Robustness	26
3.2.3	Input vs. Output Disturbances	28
	References	30
4	PID Controller Design	31
4.1	PID Controller Tuning	31
4.1.1	Analytical Tuning Methods	32
4.1.2	Tuning based on Minimisation of Performance Criteria	33

4.1.3	Tuning Rules for Robustness	34
4.2	Formalization of PID tuning as a multiobjective optimization problem	35
4.2.1	Cost Function and Constraint Selection	35
4.2.2	PID tuning problem formulation for integral cost functions	36
	References	39
5	Multiobjective optimization	43
5.1	Formalization of the multiobjective optimization problem	43
5.1.1	Definition of the Pareto front	44
5.1.2	Practical example	45
5.1.3	Different approaches to obtain the Pareto front	51
5.2	Scalarization algorithms to find the Pareto front	57
5.2.1	Weighted Sum	57
5.2.2	Normal Boundary Intersection	59
5.2.3	Normalized Normal Constraint	60
5.2.4	Enhanced Normalized Normal Constraint	62
5.3	Solution selection from the Pareto front	63
5.3.1	Visualization of the Pareto Front	64
5.3.2	The Pareto as a decision tool	65
	References	66
6	Application of the multiobjective approach	71
6.1	Comparison of the methods to obtain the Pareto front	71
6.1.1	Performance comparison of the scalarization methods	71
6.1.2	Analysis of the results from the control theory perspective	77
6.2	High-Order Benchmark Plant	84
6.3	LiTaO ₃ Thin Film Deposition Process	87
	References	91
7	PID tuning as a multi-objective optimization problem	95
7.1	Solution of the multi-objective optimization tuning	95
7.2	Viability for tuning rules	98
7.2.1	Tuning of a PI controller with two cost functions	98
7.2.2	Tuning for a Three-objective PID controller	101
7.2.3	Comments on creating tuning rules from Pareto fronts	110
7.3	Database approach for the final tuning	110
7.3.1	Example using MOOTuning	113
	References	116
8	Industrial application examples	119
8.1	Continuously Stirred Tank Heater	119
8.1.1	Description of the process	119
8.1.2	Simplified linear model	123
8.1.3	PID control of the CSTH considering two integral cost functions	125

8.1.4	PID control of the CSTR considering three integral cost functions	130
8.2	Continuously Stirred Tank Reactor	133
8.2.1	Description of the process	133
8.2.2	Linearization	137
8.2.3	Controller design	139
8.2.4	Validation of the controller designs	143
8.3	Final remarks	151
	References	153
	References	155

Abbreviations

- 2DoF two degrees of freedom.
CAD computer-aided design.
CSTH continuously stirred tank heater.
CSTR continuously stirred tank reactor.
ENNC enhanced normalized normal constraint.
FOPTD first order plus time delay.
IAE integral of the absolute value of the error.
ISE integral of the square error.
ITAE integral time absolute error.
MOO multiobjective optimization.
MOOP multiobjective optimization problem.
NBI normal boundary intersection.
NNC normalized normal constraint.
ODSOPTD overdamped second order plus time delay.
PI proportional integral.
PID proportional integral derivative.
TV total variation.
WS weighted sum.

Symbols

- $C_r(s, \boldsymbol{\theta})$ Servo component of the controller.
 $C_y(s, \boldsymbol{\theta})$ Regulator component of the controller.
 K_p Proportional gain.
 K Plant gain.
 L Time delay.
 M_s Maximum sensitivity.
 $P(s)$ Plant transfer function.
 T_d Derivative time.
 T_i Integral time.
 T Constant time.
 α Filter factor of the derivative part of the controller.
 β Weight to the reference signal in the proportional part of the two degrees of freedom controller.
 $\boldsymbol{\theta}$ Controller parameters vector.
 γ Weight to the reference signal in the derivative part of the two degrees of freedom controller.
 $d_i(s)$ Input disturbance signal.
 $d_o(s)$ Output disturbance signal.
 $r(s)$ Setpoint.
 $u(s)$ Control signal.
 $y(s)$ Feedback signal.

No preface?

Chapter 1

Introduction

The design of control systems always has to consider multiple and possibly conflicting design objectives. Under this perspective, the task of the engineer in charge of the control system, becomes to find the optimal point of compromise within this set of distinct objectives (Garpinger et al, 2012).

The most used control algorithm in industry is the proportional integral derivative (PID). This type of algorithm is used in a wide variety of applications, due to its limited number of parameters, its ease of implementation and its robustness (Åström and Häggglund, 2006) and represents an area of active study since the first tuning methodology was proposed in the 1940s (Ziegler and Nichols, 1942).

Practice?

It is common that the problem of tuning the parameters of industrial controllers is posed as an optimization problem. When all the objectives need to be taken into account at the same time, this problem becomes a multivariable, multiobjective optimization problem. In the particular case of industrial PID controllers, this problem is also non-linear and (possibly) non convex, therefore, the problem at hand is not trivial.

Regardless of the methodology to be used, it is generally computationally expensive to solve a multiobjective optimization problem, which can lead to a scenario of multiple solutions equally optimal so that in addition to solving the optimization problem, the control engineer, ends up with the extra responsibility of entering into a decision phase *a posteriori* to finally choose the best set of parameters for its specific application.

In this sense, multiobjective optimization (MOO) tuning of PID controllers remains as an open research subject, even though it has been studied for several decades. For example, in Seaman et al (1994) a type of MOO is used to tune PID controllers in a plastic injection molding process. In Abbas and Sawyer (1995), an algorithm based on several optimizations is proposed to find the optimal parameters of a PID controller; this algorithm took into account several variables such as stationary error, rise time, overrun, settling time and maximum controller output within the feedback loop. More recently, bio-inspired techniques such as neural networks, fuzzy logic or genetic algorithms have been used to solve the optimization problem Reynoso-Meza et al (2013). In Bagis (2011), A Tabu search algorithm is used to tune

PID controllers in real time, based in a set of closed loop specifications and a cost function. In Chiha et al (2012) the multiobjective optimization problem (MOOP) for PID controllers is solved using the ant colony approach, this methodology tries to simulate the behavior of real ants when they are looking of the shortest path to a given objective.

Besides bio-inspired methods for MOOP, there are several methodologies that transform the MOOP into a single function optimization problem, by rewriting the problem with extra constrains. The simplest method is the weighted sum (WS) (Marler and J.S., 2004). With the WS method, the multiobjective cost function is transformed into a one dimensional function using a weighted sum that gives a greater relative weight to a function in comparison to the others. For each set of weight values a different optimal solution is found for the same problem. The set of all solutions is part of the Pareto front (Marler and J.S., 2004). The Pareto front corresponds to all equally optimal solutions for a MOOP. The problem with the WS method is that, although the results obtained are from the Pareto front, it is not possible to satisfactorily construct the entire front (Das and Dennis, 1997; Messac et al, 2000; Marler and Arora, 2010).

In order to obtain the Pareto front correctly, other methodologies have emerged that surpass the WS. The normal boundary intersection (NBI) method consists in rewriting the optimization problem so that the feasible area is shortened by an equality constraint that depends on an extra parameter (Das and Dennis, 1998). The solution of this new problem will terminate at the Pareto border and by varying this extra parameter, it is possible to find the Pareto front so that each found point is equally spaced at the front. This feature is very useful since it gives an overall idea of the shape of the front. NBI has been applied to the tuning of controllers in Gambier (2009) where the controller is selected taking into account different performance indexes like the integral of the squared error (ISE), the integral time-weighted squared error (ITSE) and the integral of the squared time-weighted squared error (ISTSE). Other methodology similar to NBI is the normalized normal constraint (NNC) (Messac et al, 2003), which converts the MOOP in a single function optimization with an extra inequality constraint.

It should be noted that these methodologies have also been used in other areas apart from the control of industrial processes. A few examples of the areas in which it has been applied are: calculation of optimal power flow in power systems (Roman and Rosehart, 2006) and distributed generation planning (Zangeneh and Jadid, 2007), for the control of biochemical processes (Logist et al, 2009), circuit analysis (Stehr et al, 2003), development of optimal supply strategies for the participants of oligopolistic energy markets (Vahidinasab and Jadid, 2010).

The objective of this book is to present the methodology to tune PID controllers as a MOO problem. Along the book, several industrial examples are taken into account to exemplify the concepts and gain insight into the application. With some sections of the book, a companion software written on MATLAB® is included. The idea with the software is to be as open as possible, the reader will be able not only to have access to the code, but also to the data base that was obtained while solving the MOOP.

On Chapter 2 the fundamental concepts of process control are presented to set the basic foundations of this book. An Isothermal Continuously Stirred Tank Reactor is used as example to explain the methodology that is employed within the control field. Chapter 3 the metrics used for performance and robustness are presented for the case of PID control along with the tradeoffs that arises in a controlled system are considered, for example the well known relationship between servo and regulation responses, or between performance and robustness.

Chapter 4 presents the foundation of PID tuning. First the analytical tuning methods are presented in order to have the most fundamental mathematical description of a tuning rule. Then, the tuning based on the minimization of a performance criteria is considered. This subject is important for this particular book because the methodology that is presented is based on the minimization of multiple cost functions at the same time.

From Chapter 5 until the end of the book, the multiobjective case is considered. Particularly in this chapter, the basic formulation of the optimization problem is presented with the introduction of the Pareto front concept. The methodology chosen to solve the multiobjective optimization problem is to transform the multi-criteria situation into a single scalar cost function. A wastewater treatment plant model is used as example on how the Pareto can be applied to industrial processes.

The MOO techniques are tested and applied to different scenarios on Chapter 6. Different scalarization techniques are tested and the methodology is applied to a LiTaO₃ Thin Film Deposition Process. On Chapter 7, the PID tuning problem stated in Chapter 4 is solved using the ENNC methodology presented in Chapter 5. First the problem is solved using a MATLAB script that can be found in the appendix of this chapter and also downloaded as a companion software. The result of this script is a set of files that defines 2200 Pareto fronts with the optimal solutions of the problem of finding the tuning of two degrees of freedom (2DoF) PID controller for 2DoF plant families. Then two possible approaches are presented to use this results: first an attempt to find a tuning rule based on this data is presented. This approach was found to be very difficult to apply given the complexity of the data. Then the data was used as a data base and a GUI was created to serve as the bridge between the user and the results. This GUI was encapsulated as a MATLAB app and is also included as the companion software for this book.

Finally, on Chapter 8 different examples are provided to apply the tool presented in this book. The software is used to analyze the temperature control in a Continuously Stirred Tank Heater and the concentration of the product in an isothermal Continuously Stirred Tank Reactor.

References

- Abbas A, Sawyer P (1995) A multiobjective design algorithm: Application to the design of isiso control systems. Computers & Chemical Engineering 19(2):241–248, DOI 10.1016/0098-1354(94)00044-O, URL <https://linkinghub.elsevier.com/retrieve/pii/009813549400044O>

- Åström KJ, Hägglund T (2006) Advanced PID Control. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, USA
- Bagis A (2011) Tabu search algorithm based PID controller tuning for desired system specifications. *Journal of the Franklin Institute* 348(10):2795–2812, DOI 10.1016/j.jfranklin.2011.09.001
- Chiha I, Liouane N, Borne P (2012) Tuning PID Controller Using Multiobjective Ant Colony Optimization. *Applied Computational Intelligence and Soft Computing* 2012:1–7, DOI 10.1155/2012/536326, URL <http://www.hindawi.com/journals/acisc/2012/536326/>
- Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization* 14(1):63–69, DOI 10.1007/BF01197559
- Das I, Dennis JE (1998) Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization* 8(3):631–657, DOI 10.1137/S1052623496307510
- Gambier A (2009) Optimal PID controller design using multiobjective Normal Boundary Intersection technique. In: Asian Control Conference, 2009. ASCC 2009. 7th, Hong Kong, China, pp 1369–1374
- Garpinger O, Hägglund T, Åström KJ (2012) Criteria and Trade-offs in PID Design. In: Proceedings of the IFAC Conference on Advances in PID Control, Brescia, Italy
- Logist F, Erdeghem PMMV, Impe JFV (2009) Efficient deterministic multiple objective optimal control of (bio)chemical processes. *Chemical Engineering Science* 64(11):2527–2538, DOI 10.1016/j.ces.2009.01.054
- Marler RT, Arora JS (2010) The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization* 41(6):853–862, DOI 10.1007/s00158-009-0460-7, URL <http://link.springer.com/10.1007/s00158-009-0460-7>
- Marler RT, JS A (2004) Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 26(6):369–395, DOI 10.1007/s00158-003-0368-6
- Messac A, Puemi-Sukam C, Melachrinoudis E (2000) Aggregate Objective Functions and Pareto Frontiers: Required Relationships and Practical Implications. *Optimization and Engineering* 1(2):171–188, DOI 10.1023/A:1010035730904
- Messac A, Ismail-Yahaya A, Mattson C (2003) The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization* 25(2):86–98, DOI 10.1007/s00158-002-0276-1, URL <http://link.springer.com/10.1007/s00158-002-0276-1>
- Reynoso-Meza G, Garcia-Nieto S, Sanchis J, Blasco FX (2013) Controller Tuning by Means of Multi-Objective Optimization Algorithms: A Global Tuning Framework. *Control Systems Technology, IEEE Transactions on* 21(2):445–458, DOI 10.1109/TCST.2012.2185698
- Roman C, Rosehart W (2006) Evenly distributed pareto points in multi-objective optimal power flow. *Power Systems, IEEE Transactions on* 21(2):1011–1012, DOI 10.1109/TPWRS.2006.873010
- Seaman CM, Desrochers AA, List GF (1994) Multiobjective optimization of a plastic injection molding process. *Control Systems Technology, IEEE Transactions on* 2(3):157–168, DOI 10.1109/87.317974
- Stehr G, Graeb H, Antreich K (2003) Performance trade-off analysis of analog circuits by normal-boundary intersection. In: Proceedings of the 40th conference on Design automation - DAC '03, ACM Press, New York, New York, USA, p 958, DOI 10.1109/DAC.2003.1219159, URL <http://portal.acm.org/citation.cfm?doid=775832.776073>
- Vahidinasab V, Javid S (2010) Normal boundary intersection method for suppliers' strategic bidding in electricity markets: An environmental/economic approach. *Energy Conversion and Management* 51(6):1111–1119, DOI 10.1016/j.enconman.2009.12.019
- Zangeneh A, Javid S (2007) Normal boundary intersection for generating Pareto set in distributed generation planning. In: 2007 International Power Engineering Conference (IPEC 2007), Singapore, pp 773–778
- Ziegler J, Nichols N (1942) Optimum Settings for automatic controllers. *ASME Transactions* 64:759–768

Chapter 2

Industrial PID Control

Abstract In this chapter the fundamental concepts of process control are presented to set the basic foundations of this book. An Isothermal Continuous Tank Reactor is used as example to explain the methodology that is employed within the control field. Special attention is put in the usage of the proportional integral derivative (PID) controller as the fundamental algorithm to close the loop in any process system. The modeling of the plant is also considered as well as its normalised representation used for control tuning.

2.1 Control System Design Scenario

Control systems are used to maintain process conditions at their desired values by manipulating certain process variables to adjust the variables of interest. A common example of a control system from everyday life is the cruise control on an automobile. The purpose of a cruise control is to maintain the speed of the vehicle (the controlled variable) at the desired value (the set point) despite variations in terrain, hills, etc. (disturbances) by adjusting the throttle, or the fuel flow to the engine (the manipulated variable). Another example is the home thermostat. This control system is designed to maintain the temperature in the home at a comfortable value by manipulating the fuel flow or electrical input to the furnace. The furnace control system must deal with a variety of disturbances to maintain temperature in the house, such as heat losses, doors being opened and hope- fully closed, and leaky inefficient windows. The furnace must also be able to respond to a request to raise the desired temperature if necessary. For example, we might desire to raise the temperature by 5 degrees and we would like the system to respond smoothly and efficiently. From these examples, we can deduce that there are several common attributes of control systems:

- The ability to maintain the process variable at its desired value in spite of disturbances that might be experienced (this is termed disturbance rejection)

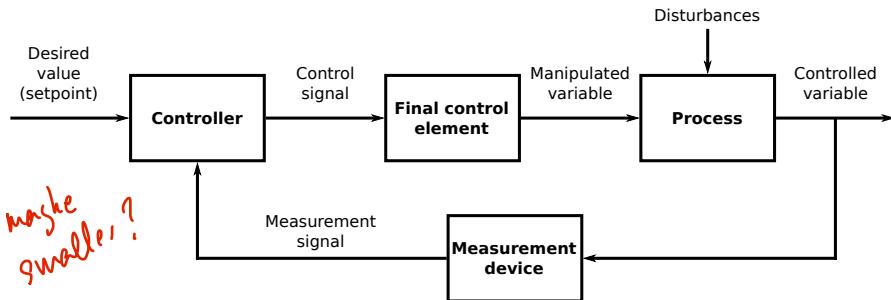


Fig. 2.1: Feedback Control System

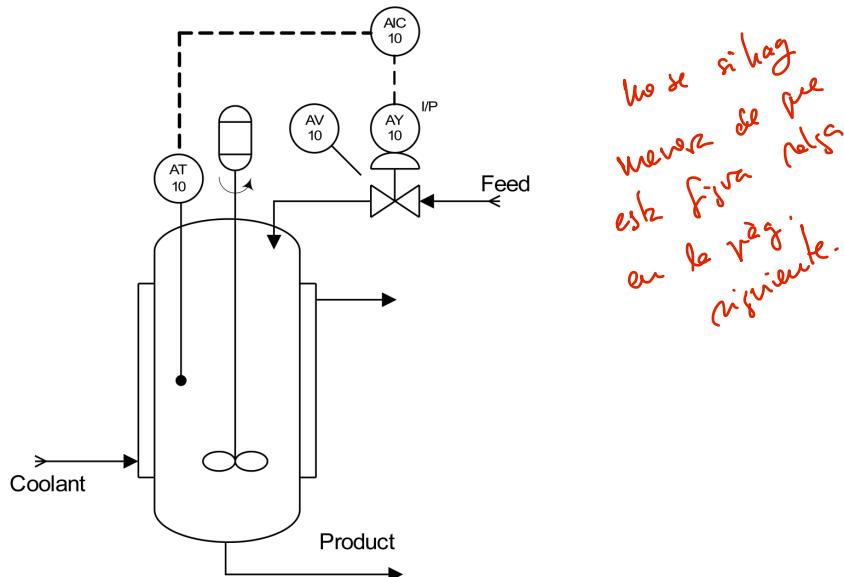


Fig. 2.2: Isothermal Continuous Stirred Tank Reactor (CSTR)

- The ability to move the process variable from one setting to a new desired setting (this is termed set point tracking)

A natural way to adjust or correct the behaviour over time of a dynamic system output, the controlled variable, is by using an actuating input computed on the basis of the comparison of the actual output with its desired value: the feedback error. This is, by means of closed-loop control. To compute the control action information about the feedback error evolution is required. Normally its current value, its past evolution, and a prediction of its future behaviour are used. The way we use this information to deliver the control action constitutes the control algorithm. Conceptually we can view the control systems in the general manner shown in figure (2.1).

As a detailed practical example, consider the isothermal Continuously Stirred Tank Reactor (CSTR), as the one in figure (2.2), where the isothermal series/ parallel Van de Vusse reaction takes place (Arrieta et al, 2008), (Kravaris and Daoutidis, 1990) . The reaction can be described by the following scheme:



From a mass balance, the system can be described by the following model:

$$\begin{aligned} \frac{dC_A(t)}{dt} &= \frac{F_r(t)}{V} [C_{Ai} - C_A(t)] - k_1 C_A(t) - k_3 C_A^2(t) \\ \frac{dC_B(t)}{dt} &= -\frac{F_r(t)}{V} C_B(t) + k_1 C_A(t) - k_2 C_B(t) \end{aligned} \quad (2.2)$$

where F_r is the feed flow rate of product A, V is the reactor volume which is kept constant during the operation, C_A and C_B are the reactant concentrations in the reactor, and k_i ($i = 1, 2, 3$) are the reaction rate constants for the three reactions.

From the point of view of the feedback control system depicted in figure (2.1), the controlled variable is the product concentration C_B , the manipulated variable the feed flow rate F_r . Expected disturbances on the system may come from variations on the input product concentration C_A as well as flow rate. In figure (2.3) we can observe how the reactor output concentration reacts to changes in each one of its two inputs: The inlet flow rate, F_r and concentration C_A . Whereas the first one can be manipulated the second one is considered *now known* as supplied externally. Therefore, F_r is considered as the manipulated variable and will be the one used to operate and control the reactor. On the other hand, changes in C_A will be seen as disturbances and the controller should be able to counteract such changes and prevent them to generate changes in the output concentration C_B .

not Known.

The feedback control structure has been used for a long time, but if we constraint ourselves to the industrial process control area, the proportional (present) integral (past) derivative (future) (PID) control algorithm age starts in the 40's with the PID controller and this will be the concern of this text. A control algorithm has a number of control parameters, which must be *tuned* (adjusted) to have acceptable performance. Often the tuning is done on a simulation model before implementing the control strategy on the actual process. In this text ~~we~~ will concentrate on the determination of the controller tuning, in fact, PID tuning, by means of optimisation methods. Specifically, optimisation methods that deal with multiple objectives at the same time. However, prior to this task, we should define the control scenario where the design will take place.

we

Consider the general two-degree-of-freedom closed-loop control system depicted in figure 2.4 where $P(s)$ and $\{C_r(s), C_y(s)\}$ are the controlled process model and the controller transfer functions, respectively. In this system, $r(s)$ is the set-point, $u(s)$ the controller output signal, $d(s)$ the disturbance, $y(s)$ the process controlled

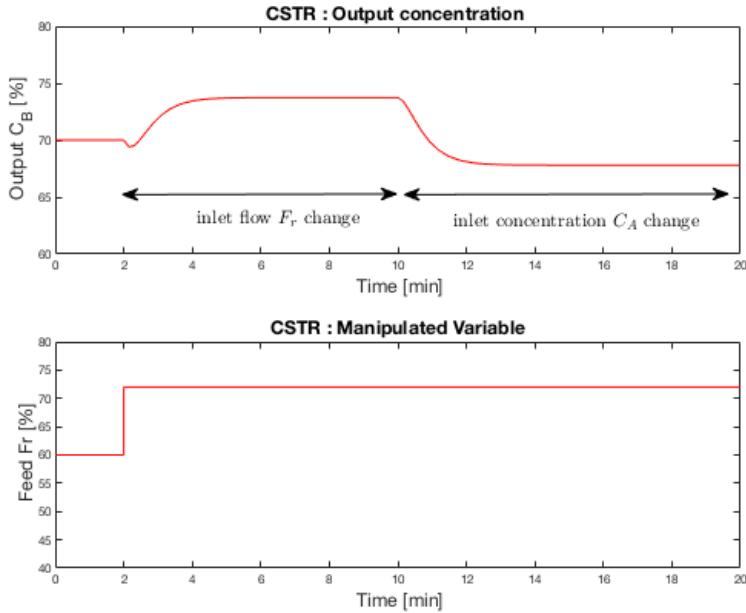


Fig. 2.3: CSTR Open-loop output to change in inlet flow and concentration

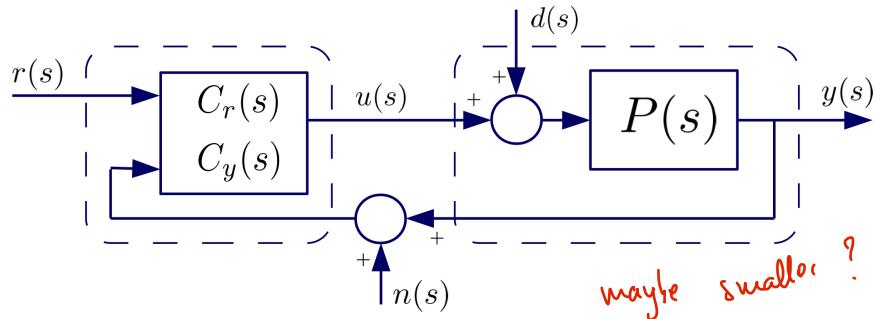


Fig. 2.4: Two-Degree-of-freedom closed-loop control system.

variable, and $n(s)$ the measurement noise. It is assumed that the disturbance enter at the process input (load-disturbance).

The closed-loop control system output $y(s)$ as a function of its inputs $r(s)$, $d(s)$, and $n(s)$ is:

$$y(s) = M_{yr}(s)r(s) + M_{yd}(s)d(s) + M_{yn}(s)n(s), \quad (2.3)$$

where:

$$M_{yr}(s) \doteq \frac{C_r(s)P(s)}{1 + C_y(s)P(s)}, \quad (2.4)$$

is the servo-control closed-loop transfer function,

$$M_{yd}(s) \doteq \frac{P(s)}{1 + C_y(s)P(s)}, \quad (2.5)$$

the regulatory control closed-loop transfer function, and

$$M_{yn}(s) \doteq \frac{-C_y(s)P(s)}{1 + C_y(s)P(s)}, \quad (2.6)$$

the measurement noise sensitivity function.

The regulatory control main objective is *load-disturbance rejection*; this is, to return the controlled variable to its set-point in the event a disturbance enters to the control system. For the servo-control, it is intended to *follow a changing set-point*; this is, to bring the controlled variable to its new desired value. Controller tuning for above operations must take also into account to not ~~amplified~~ the measurement noise, if any. In figure (2.5) we can see the CSRT example presented above, where a controller, is accomplishing the task of, first, tracking a set-point step change of, followed by a disturbance attenuation of two changes. One in the concentration, C_{Ai} , of A in the feed flow and finally a change in the supplied flow rate.

amplify

2.2 Industrial Process Characteristics

Before a controller for a process is specified, the process to be controlled should be characterised, at least in some broad sense. There are many different types of processes that are controlled automatically. Examples range from fluid levels in tanks to read-write heads of computer disk storage devices. The control inputs to a process are supplied through one or more actuators. For example a motor driven valve can be an actuator for fluid flowing into a tank. Process outputs that are being controlled (e.g., the fluid level in a tank) are measured using appropriate sensors. The basic control of one output variable by the use of one control input variable is called single-input single-output (SISO) control. For more complex systems, multi-input multi-output (MIMO) control may be required. PID control was developed initially as a SISO control strategy, but it has been extended in various ways to MIMO control (Wang et al, 2008). Event these extensions, it is common practice in the process industry to rely on PID control at single loop level control problems, whereas multi-variable control solutions, such as model predictive control are the ones applied for multivariable and supervisory control (Vilanova and Visioli, 2012).

In the great majority of process loops, applying a step to the ~~control~~ variable causes the controlled variable to reach a steady state and does not provoke an instantaneous variation of it. This means that the process model seen by the ~~regula-~~ controller

manipulated

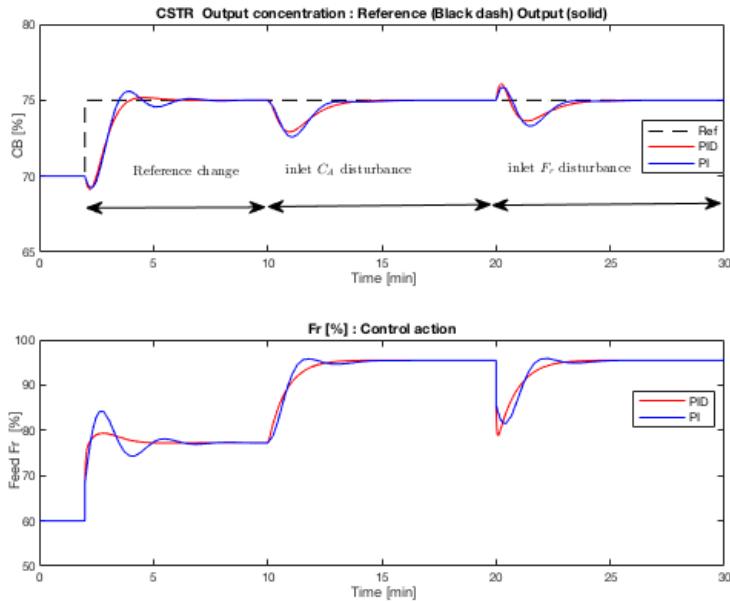


Fig. 2.5: CSTR Open-loop output to disturbance changes in inlet flow and concentration

tor can be described by an asymptotically stable, strictly proper transfer function. FOPTD dynamics are probably the most usual models used in the process industry for modelling self-regulating behaviours. For example when modelling dynamics that involve balances of temperature, matter, and energy in general. Also in a reactor where a reaction takes place, the components balance generates a dynamics that can be usually modelled as a FOPTD. When different reactors are connected in series, higher order systems arise that can be described by SOPTD models.

In a few loops, a control step causes the controlled variable to asymptotically assume a ramp-like behaviour. This case is commonly referred to as integrating or non self-regulating processes. These dynamics arises, for example, when dealing with level problems. Also when dealing with distillation columns control problems, integrative behaviour arises. Other cases (e.g. an oscillatory response with significant delay, unstable dynamics, etc) may exist, but they are unlikely to appear in practice. This is what motivates the current approaches to PID control to concentrate on stable self-regulating dynamics and these are the process models adopted in this book to define the working scenario. However, from the provided methodology and tools it should become clear that the work could be extended to some other process dynamics.

A complete and illustrative source of modelling for process control, illustrating how the previously commented dynamics arise is the extensive book (Marlin, 2015), whereas industrial applications can be sourced, for example, in (Vilanova and Visoli, 2012).

2.2.1 Controlled Process Model

The simple structure of the PID controller calls for simple process descriptions, this fact motivates the use of first or second order models. On that respect, the controlled process model in the control design scenario considered in this book will be the one aimed at representing the self-regulating non-oscillating (overdamped) step responses. The overdamped controlled processes will be represented by a linear model given by the transfer function

$$P(s) = \frac{K_p e^{-Ls}}{(Ts + 1)(aTs + 1)}, \quad \tau_o = L/T \quad (2.7)$$

where K_p is the model gain, T its main time constant, a the ratio of its two time constants ($0 \leq a \leq 1.0$), L its dead-time, and τ_o the model *normalized dead-time* ($0.1 \leq \tau_o \leq 2.0$). Model transfer function (2.7) allows to represent First-Order-Plus-Dead-Time (FOPDT) processes, with $a = 0$, over damped Second-Order-Plus-Dead-Time (SOPDT) processes, with $0 < a < 1$, and Dual-Pole-Plus-Dead-Time (DP-PDT) processes, the $a = 1$ case.

In the great majority of cases, a process description is obtained by performing an experiment on the process. In most cases, a step response will be used. They are also easy to apply. It suffices to switch the regulator to manual, wait until a reasonably steady state is reached, then change the control variable suddenly by an amount sufficient to make the response obtained easily distinguishable from noise. In addition, step tests permit the process to be maintained under reasonable control without perturbing it excessively or leading it to the stability boundary, as required e.g. by the closed loop Ziegler-Nichols method (Åström and Hägglund, 2006). From this point of view, to maintain the need for plant experimentation to a minimum is a key point when considering the industrial application of a technique. The parameters of the controlled process model (2.7), $\bar{\theta}_p = \{K_p, T, a, L, \tau_o\}$, may be identified from the process reaction curve by using, for example, the method presented in (Alfaro, 2006).

2.3 The PID Controller

Undoubtedly, since its introduction, PID controllers are the option most frequently used in different process control applications. Their success is mainly due to the

simplicity of their structure (three parameters to tune) and operation, which allows the control engineer a better understanding compared with other advanced control techniques. This has motivated the continuous research efforts aimed at finding alternative approaches to the design and new tuning rules in order to improve the performance of control loops based on PID controllers. Different reports confirm that currently the PID continues to be the workhorse of process industry, being completely integrated within more advanced control algorithms and providing the fundamental base layer for plant-wide solutions. The proper function of a PID-based control loop is, therefore, a key aspect in the current process industry and of continuing interest for researchers.

The application of a PID controller is, essentially, the result of weighting three different actions each one related to the information provided by the time history of the error signal: the instantaneous actual value provided by the *proportional term*, $u_P(t)$, past values provided by the *integral term*, $u_I(t)$, and the predicted future values provided by the *derivative term*, $u_D(t)$. In its simplest form, the PID control signal is computed as

$$u(t) = u_P(t) + u_I(t) + u_D(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right) \quad (2.8)$$

which corresponds, when expressed in the form of the transfer function from the error $e(s)$ to $u(s)$, to

$$u(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) e(s) \quad (2.9)$$

This form is usually referred to as the *ideal* PID. The three-term functionalities are highlighted by the following.

Proportional term

Also referred as the P mode, this mode is almost universal and present in all controllers. With reference to (2.9), the control law in this case is given by

$$u_P(t) = K_p e(t) + u_{ss}$$

where $u_P(t)$ is the (proportional) controller output, K_p is the controller gain and u_{ss} is a bias or reset value. The P action makes the control proportional to the error. Hence it obeys to the intuitive principle that, the bigger the error the bigger the control action must be.

The P action depends only on the instantaneous value of the error and is nonzero only if $e(t)$ is nonzero. In other words, the P action is ideally zero at steady state, but only provided that the required steady state can be reached with zero control. If this is not the case it will be necessary to *reset* $u(t)$, i.e. to add a constant term to it

so that it maintains the required steady state; if only the P action is used, this is the role of u_{ss} . However, the reset can also be accomplished by the integral action, and that is why in older regulators this action is also called *automatic reset*.

Controllers

Integral term

Integral (or reset) action produces a controller output that is proportional to the accumulated error. The control law in this case is given by

$$u_I(t) = \frac{K_p}{T_i} \int_0^t e(\tau) d\tau$$

where T_i is the integral, or reset, time constant. Note that $u_I(t)$ also depends on the controller gain. This is because $u_I(t)$ is proportional to the sum of the system errors, integral action is referred to as a *slow mode*. Åström and Hägglund (2006) point out that integral action can also be viewed as a device that automatically resets the bias term u_{ss} of a proportional controller. This follows immediately by considering that at steady state, the P action is zero except for u_{ss} . In other words, the I action guarantees zero steady state error because, whenever $e(t)$ is the input of an integrator, there cannot be any steady state if $e(t)$ is nonzero.

Derivative term

The final mode is the derivative action. Here the control is proportional to the rate of change of the error signal. It follows that whenever the error signal is constant, the derivative signal contributes zero. The control law in this case is given by

$$u_D(t) = K_p T_d \frac{d}{dt} e(t)$$

Where T_d is the derivative or rate time constant. Problems may arise when the error signal is entrenched in (high-frequency) noise or when step changes in the set point occur, since in these cases derivative action will generate large amplitude signals. Derivative action is referred to as a *fast mode* that generally improves the loop stability. It is often said that the D action *anticipates the future*. The message that increasing the derivative gain, will lead to improved stability is commonly conveyed from academia to industry. However, practitioners have often found that the derivative term can behave against such anticipation particularly when there exists a transport delay (Vilanova and Visioli, 2012). Frustration in tuning has hence made many practitioners switch off or even exclude the derivative term.

Another issue is that the D part of the PID controller in the ideal form (2.9), is not proper. To overcome this, it is commonly implemented as

$$U_D(s) = K_p \frac{T_d s}{\alpha T_d s + 1} E(s)$$

This is often referred to as using a real derivator. In this way, α becomes another parameter of the PID that has to be selected. It is worth noting that a small values for α makes the implementation of the D action similar to a true derivative but it also increases the high frequency gain, thus increasing noise sensitivity.

Taking into account this modification for the ideal derivative term, the PID controller can be expressed in the s -domain with the following overall transfer function:

$$u(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1} \right) e(s) \quad (2.10)$$

2.3.1 PID Controller formulations

The PID algorithm as presented is usually referred as the standard one. However, the combinations of the three basic control actions may come in other different formulations. In fact, usually, the control algorithm implementation is manufacturer dependent and not all of its variations are available in the same controller. Even more, the controllers manufacturers use different names for the same PID algorithm (Gerry, 1987) (Vilanova and Visioli, 2017). The diversity of the PID control algorithms is evident in O'Dwyer (2009) . In addition, it might be the case that a tuning rule of interest had been obtained using a control algorithm different from the one implemented in the controller to tune. In this case, as it is not guaranteed that the equivalent controller exists, controller parameters conversion relations are required, that will also indicate if the pursued equivalent controller exists. In what follows, the basic PID controller formulations are presented by using a different notation for the parameters in each one of them. This will allow later on the definition of transformation equations for computing the parameters for one formulation from another one.

- *Standard PID form:* The *textbook* proportional integral derivative control algorithm is the Standard PID whose output is given by the following expression (Åström and Hägglund, 1995):

$$u(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1} \right) e(s) \quad (2.11)$$

- *Parallel PID form:* The parallel or *independent gains* PID control algorithm is

$$u(s) = \left(K_p + \frac{K_i}{s} + \frac{K_d s}{\alpha_p K_d s + 1} \right) e(s) \quad (2.12)$$

where each control action has its own independent gain. The gains of the parallel form can be easily related to the gains for the standard form. Whereas the proportional gain is the same, for the integral and derivative gains we have:

$$K_i = \frac{K_p}{T_i} \quad K_d = K_p T_d \quad \alpha_p = \alpha K_p \quad (2.13)$$

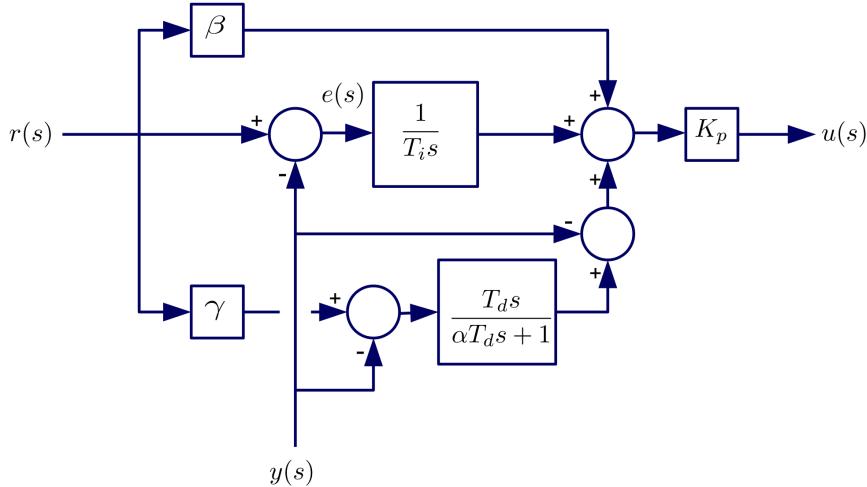


Fig. 2.6: Two-degree-of-freedom Standard PID controller.

- *Series PID form:* The series or *interacting* implementation of the PID algorithm corresponds to the serial connection of a PI and a PD controller. The resulting transfer function is

$$u(s) = K'_p \left(\frac{T'_i + 1}{T'_i s} \right) \left(\frac{T'_d s + 1}{\alpha' T'_d s + 1} \right) e(s) \quad (2.14)$$

In this case, parameters are denoted with a prime in order to distinguish them from the standard form ones. As a first difference with the previous formulations, it can be observed that with the series form we can not have complex conjugate zeros. If, for simplicity, we assume the ideal case, $\alpha = \alpha' = 0$, a series PID controller equivalent to a Standard one would exist only for $T_i \geq 4T_d$ (this ensures the Standard PID does not have complex conjugate zeros).

- *Filtered ideal PID form:* This formulation arises also as an alternative to the mentioned implementation problems with the ideal derivative. In this case, the overall control variable is filtered:

$$u(s) = K^*_p \left(1 + \frac{1}{T^*_i s} + T^*_d s \right) \left(\frac{1}{T_f s + 1} \right) e(s) \quad (2.15)$$

In this case, parameters are denoted with a star in order to distinguish them from the standard and series form ones. Notice that the controller variable filter introduces the filter time constant T_f as an additional controller parameter. This fact will make the relationship of this controller formulation parameters with the previous ones not so straightforward. It will be presented later on on a more general framework.

2.3.2 Reference Processing and 2-DoF PID

The previous PID controller formulations can be improved by introducing some considerations on the processing of the reference signal. The first one is regarding the effect of a step change Δr in the derivative part. Considering, for simplicity, a PD controller, an instantaneous change Δu in the control signal will be generated. This instantaneous change will be of magnitude

$$\Delta u = K_p \left(1 + \frac{1}{\alpha} \right) \Delta r \xrightarrow{\alpha=0.1} \Delta u = 11K_p \Delta r \quad (2.16)$$

This is known as the *derivative kick*. In order to avoid this, it is suggested to feed the derivative with just the output signal. In this case, the PD controller will take the form

$$u(s) = K_p e(s) - K_p \frac{T_d s}{\alpha T_d s + 1} y(s) \quad (2.17)$$

Along the same lines as with the derivative term, a sudden change of magnitude Δr in the reference signal generates an instantaneous change Δu in the control signal given by $\Delta u = K_p \Delta r$. Therefore, for a relatively high controller gain an excessively abrupt change in the actuator may be demanded. As this is an undesirable feature, the reference signal, in the proportional control action, is recommended to be scaled by a factor β known as the *set-point weighting* factor. The proportional part of the controller is therefore rewritten as

$$u(s) = K_p (\beta r(s) - y(s)) \quad (2.18)$$

By choosing $\beta < 1$, the control signal instantaneous change can be scaled down to $\Delta u = K_p \beta \Delta r$ without the need to reduce the controller gain.

When the *set-point weighting* factor is considered into the PID controller implementation, the resulting controller is said to have two-degrees-of-freedom (2-DoF) as a different processing of the reference and feedback signal is allowed. In such case, the 2-DoF versions of the previously presented PID controller formulations, also considering the avoiding of the derivative kick, results as:

2.3.3 Conversion of 2-DoF PID Controller Algorithms

As it can be observed from the presented PID formulations, whereas the reference controller aspect $C_r(s)$ takes the same form in all formulations, it is the feedback part $C_y(s)$ that prevents a direct translation of the controller parameters from one formulation to another. This is important because some of the existing tuning rules have been conceived for a specific PID formulation. Due to the possibility that the PID algorithm of the controller to tune be different to the one considered by the tuning rule to use, it is necessary to have conversion relations to obtain “equivalent”

parameters between two or more of them (Alfaro and Vilanova, 2012; Vilanova and Visioli, 2017). In what follows, we present conversion formulae to get the controller parameters for one specific PID formulation starting from the parameters got for another different one. In order to be as general as possible, the conversion formulae is presented for the more generic 2-DoF PID controller formulations just presented above.

*En todo caso, basar cf. a libro KROFIK
• ETFA 2012
(y en las RFS)*

2.4 Normalised Representations

The design approach that is to be presented in the following chapters is applied to controlled processes represented by stable over-damped models. These models encompass from first order to double pole stable models. For control system performance analysis and controller tuning it is convenient to work with dimensionless parameters to make it non dependent on the controlled process time scale and gain. Therefore in this section the process model as well as controller transfer functions to be considered will be rewritten in their normalised form in terms of dimensionless parameters. Therefore, in this book, all the results are based on normalised transfer function models as well as the corresponding normalized controller parameters. In this way we ensure that controller design is consistent from the point of view of being applicable to all transfer function models equivalent to the normalised one but to a time-scaling. An additional advantage is that the number of process model transfer function do have one parameter less.

2.4.1 Process Model Normalisation

The *over-damped* controlled process (first- and second-order) are represented by a linear model given by the transfer function presented in (2.7)

$$P(s) = \frac{Ke^{-Ls}}{(Ts+1)(aTs+1)}, \quad \theta_p = \{K, T, a, L\}, \quad (2.19)$$

where K is the model gain, T the main time constant, a the ratio of the two time constants ($0 \leq a \leq 1.0$), and L the dead-time.

Using the controlled process model gain K , and time constant T , as well as the transformation $\hat{s} \doteq Ts$, the controlled process (2.7) can be expressed in normalised form as follows:

$$\hat{P}(\hat{s}) = \frac{e^{-\tau_L \hat{s}}}{(\hat{s}+1)(a\hat{s}+1)}, \quad \tau_L \doteq \frac{L}{T}, \quad (2.20)$$

where τ_L is the normalised (dimensionless) dead-time.

The over-damped second-order plus dead-time (SOPDT) (2.20) model has two normalised parameters, $\hat{\theta}_p = \{a, \tau_L\}$. For the particular case of the first-order plus dead-time (FOPDT) model ($a = 0$) it has only one, $\hat{\theta}_p = \tau_L$. Using the same procedure normalised models are obtained for the other processes.

2.4.2 Controller Normalisation

Regarding the controller, to consider the control ~~transfer~~ function alone does not make sense. It has to be considered in conjunction with the process model transfer function to be controlled. Therefore, according to the normalisation of the process model, the controlled transfer function has also to be scaled. This will define the normalised controller parameters. Next we consider the normalization of the *Standard 2DoF PID controller PID₂* from where the normalized parameters of other 2DoF PID control algorithms can be found.

For example, the output equation of the normalized version of the *Standard 2DoF PID controller PID₂* in (2.11), with the $\hat{s} \doteq Ts$ transformation, is given by

$$u(\hat{s}) = \kappa_p \left\{ \beta r(\hat{s}) - y(\hat{s}) + \frac{1}{\tau_i \hat{s}} [r(\hat{s}) - y(\hat{s})] - \left(\frac{\tau_d \hat{s}}{\alpha \tau_d \hat{s} + 1} \right) y(\hat{s}) \right\}, \quad (2.21)$$

with parameters $\hat{\theta}_c = \{ \kappa_p, \tau_i, \tau_d, \alpha, \beta \}$. Therefore, for *over-damped first- and second-order plus dead-time*, models, using the corresponding model parameters the associated *PID₂* controllers parameters can be expressed in normalised form as follows:

$$\kappa_p \doteq KK_p, \quad \tau_i \doteq \frac{T_i}{T}, \quad \tau_d \doteq \frac{T_d}{T} \quad (2.22)$$

In case the controller is implemented as a *PID Parallel controller* the corresponding normalized parameters are:

$$\kappa_p \doteq KK_p, \quad \kappa_i \doteq KK_i T, \quad \kappa_d \doteq \frac{KK_d}{T}. \quad (2.23)$$

References

- Alfaro VM (2006) Low-Order Models Identification from the Process Reaction Curve. Ciencia y Tecnología (Costa Rica) 24(2):197–216
 Alfaro VM, Vilanova R (2012) Conversion Formulae and Performance Capabilities of Two-Degree-of-Freedom PID Control Algorithms. In: 17th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA 2012)

- Arrieta O, Vilanova R, Alfaro VM, Moreno R (2008) Considerations on PID Controller Operation: Application to a Continuous Stirred Tank Reactor. In: 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA08), September 15-18, Hamburg-Germany
- Åström KJ, Hägglund T (1995) PID Controllers: Theory, Design and Tuning. Instrument Society of America, Research Triangle Park, NC 27709, USA
- Åström KJ, Hägglund T (2006) Advanced PID Control. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, USA
- Gerry JP (1987) A comparison of PID algorithms. *Control Engineering* 34 (3):102–105
- Kravaris C, Daoutidis P (1990) Nonlinear state feedback control of second order nonminimum-phase nonlinear systems. *Computers Chem Eng* 14(4-5):439–449
- Marlin TE (2015) Process Control Designing Processes and Control Systems for Dynamic Performance. McGraw-Hill Science/Engineering/Math
- O'Dwyer A (2009) Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London, UK
- Vilanova R, Visioli A (2012) PID Control in the Third Millennium. Advances in Industrial Control, Springer London, London, DOI 10.1007/978-1-4471-2425-2, URL <http://link.springer.com/10.1007/978-1-4471-2425-2>
- Vilanova R, Visioli A (2017) The Proportional-Integral-Derivative (PID) Controller. In: Wiley Encyclopedia of Electrical and Electronics Engineering, American Cancer Society, pp 1–15, DOI 10.1002/047134608X.W1033.pub2, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/047134608X.W1033.pub2>
- Wang QG, Ye Z, Cai WJ, Hang CC (2008) PID Control for Multivariable Processes, Lecture Notes in Control and Information Sciences, vol 373. Springer Berlin Heidelberg, Berlin, Heidelberg, DOI 10.1007/978-3-540-78482-1, URL <http://link.springer.com/10.1007/978-3-540-78482-1>

Chapter 3

PID Controller Considerations

Abstract In this chapter the metrics used for performance and robustness are presented for the case of proportional integral derivative (PID) control. Special attention is given to the integral measure of error. Also the tradeoffs that arises in a controlled system are considered, for example the well known relationship between servo and regulation response, or between performance and robustness. This trade-offs are what gives meaning to the multiobjective approach that will be taken in the rest of the book.

evaluation
↓
presented ?
afforded ?
dealt ?

3.1 Control System Evaluation Metrics

The need for quantitative metrics that provide insight in which measure a given controller tuning provides answer to a specific design criteria can be considered a common sense acceptable fact. However, it becomes an indispensable tool when coming to the need to combine different control system specifications, as it happens in a multiobjective approach. It has been a common practice in the control systems literature to use different indexes to measure such properties accomplishment, both from the performance as well as robustness point of view.

Taking into account that in industrial process control applications, a good load-disturbance rejection (usually known as regulatory control) as well as a good transient response to set-point changes (known as servo-control operation) is required, the controller design should consider both possibilities of operation. Despite this, the servo and regulation demands cannot be optimally satisfied simultaneously with a one degree-of-freedom (1-DoF) controller, because the resulting dynamic for each operation mode is different and it is possible to choose just one for an optimal solution. Therefore, there will be the need to quantify the level of performance accomplishment regarding each one of the operational modes. On the other hand, the control system design is usually based on the use of low-order linear models; these models in turn are based on the normal operating point of the closed-loop control system. Because most industrial processes are non-linear, it is necessary to account

modo en cursiva

to do en wrist

for possible changes in the process characteristics by adopting certain relative stability margins or robustness requirements for the control system.

Therefore, in the design of a closed-loop control system with PI and PID controllers, we must consider the trade-off between two conflicting criteria: the time-response performance to the set point and load disturbances and the robustness to changes in the characteristics of the controlled process. In order to manage those conflicting objectives in a suitable way, suitable metrics are presented in what follows.

3.1.1 Performance

The performance of a control system may be evaluated by using different measures. In control textbooks it is usual to find a characterisation of the time response in terms of numerical quantities assimilated to a second order underdamped system such as: percentage overshoot, rise time, etc. However, in academic and research works it is more usual and convenient to use a cost function based on the error, i.e. the difference between the desired value (set-point) and the actual value of the controlled variable (system's output). Of course, as larger and longer in time is the error, the system's performance will be worse. In recent years have become very popular those related to the integrated error, given in general by the following formulation:

$$J_e \doteq \int_0^{\infty} t^p |e(t)|^q dt, \quad (3.1)$$

where the error can be generated because either of a set-point change or a load-disturbance.

A review of research history on PID controller design reveals that, among the most used ones, there have been the integrated absolute error (IAE), the integrated time-weighted absolute error (ITAE), or the integrated squared error (ISE). From an academic point of view, objective functions can take any one wished form. However, from an industrial point of view, realistic economic objectives need to be addressed. As it is desirable to use a performance indicator that takes into account economic considerations, the integrated error (IAE) is suggested Shinskey (2002) as a meaningful measure as it can be assimilated to product giveaway, excess consumption of utilities, and reduction in plant capacity. Taking this into account, to avoid the cancellation of positive and negative errors, there seems to be a *de facto* agreement with the use of the Integrated Absolute Error (IAE) ($p = 0, q = 1$ in 3.1) given by the following formula:

$$J_e \doteq \int_0^{\infty} |e(t)| dt = \int_0^{\infty} |r(t) - y(t)| dt. \quad (3.2)$$

3.1.2 Robustness

Robustness is an important attribute for control systems, because the design procedures are usually based on the use of low-order linear models identified at the closed-loop operation point. Due to the non-linearity found in most of the industrial process, it is necessary to consider the expected changes in the process characteristics assuming certain relative stability margins, or robustness requirements, for the control system. The robustness is a measure of how much the controller can tolerate changes in the process transfer function; more specifically, in its gain and in its phase lag.

Also in this case there are some useful indicators that can be used as design criteria for robustness. The most widely accepted in industrial practice are the gain and phase margin. The gain margin (A_m) is a measure of how much the process gain can change before the closed loop system becomes unstable. Control theory says it is the amount of gain increase or decrease required to make the loop gain unity at the frequency where the phase angle is -180° . Therefore leading the closed-loop system to the critical point. On the other hand, the phase margin (ϕ_m) is a measure of how much the process phase can change before the closed loop system become unstable. The phase could be increased because of an additional delay or because the process lag decrease.

The use of the gain and phase margins as robustness measures, has been replaced by the use of a single indicator, the maximum of the sensitivity function, denoted by M_s , given by the shorter distance from the Nyquist diagram to the real point -1 ; this maximum sensitivity is strictly related to the gain and phase margin through some simple inequalities. Then, for each controller parameter set obtained, the closed-loop control system robustness is measured using control system Maximum Sensitivity M_s defined as:

$$M_s \doteq \max_{\omega} |S(j\omega)| = \max_{\omega} \frac{1}{|1 + C(j\omega)P(j\omega)|} \quad (3.3)$$

The recommended values for M_s are typically within the range 1.4 - 2.0 (Åström and Hägglund (2006)). The use of the maximum sensitivity as a robustness measure, has the advantage that lower bounds to the gain, A_m , and phase, ϕ_m , margins (Åström and Hägglund (2006)) can be assured according to

$$A_m > \frac{M_s}{M_s - 1} \quad ; \quad \phi_m > 2 \sin^{-1} \left(\frac{1}{2M_s} \right)$$

Therefore, ensuring $M_s = 2.0$ provides what is commonly considered minimum robustness requirement (that translates to $A_m > 2$ and $\phi_m > 29^\circ$, for $M_s = 1.4$ we have $A_m > 3.5$ and $\phi_m > 41^\circ$). Even if there are different measures for the closed-loop system robustness, the idea spread today to a common use of the maximum of the sensitivity function (commonly called M_S) as a reasonable robustness measure.

3.1.3 Control Input Usage

Controller design problems are stated in terms of the controlled variable (usually the process output). Depending on how this problem is stated and solved, this may generate controller settings that produce command signals that are either undesirable or not realistic. It is therefore always needed to evaluate the control signal and take care of the controller bandwidth. This is usually related to the variation of the control signal as a measure of its *smoothness*. For the evaluation of the required *control effort* the control signal total variation TV_u given by the difference between the values of the control variable at two consecutive sampling time instants:

$$TV_u \doteq \sum_{k=1}^{\infty} |u_{k+1} - u_k|, \quad \text{no wisita.} \quad (3.4)$$

is used as main indication of the *smoothness* of the control action for input changes.

As a complementary measurements of the control effort it can also be considered the controller output instant change to a set-point step change (the “proportional kick”) given by

$$\Delta u_0 \doteq \beta K_p \Delta r, \quad (3.5)$$

3.2 Control System Tradeoffs

Goals

When considering a control task, there are different aspects the operator desires the control system to behave in a prescribed way or, why not, as better as possible. This is what we refer to control system performance with respect to, for example, set-point tracking, disturbance rejection, control effort reduction, and implicitly, with some degree of tolerance to change in process operating conditions: robustness. It is well known these ~~goals~~ are in conflict among them and cannot be achieved simultaneously Arrieta et al (2010); Alcantara et al (2013) In such case a trade-off between the objectives is required. Improving one objective may mean poor performance in another, or less robustness. As we have seen, every goal should be translated into design specifications, and specific indices in order to measure the performance of the PID controller.

Taking into account that in industrial process control applications, it is required a good load-disturbance rejection (~~usually known as regulatory-control~~), as well as, a good transient response to set-point changes (~~known as servo-control operation~~), the controller design should consider both possibilities of operation. Despite the above, the servo and regulation demands cannot be optimally satisfied simultaneously with a One-Degree-of-Freedom (1-DoF) controller, because the resulting dynamic for each operation mode is different and it is possible to choose just one for an optimal solution. Considering the previous statement, most of the existing studies have focused only in fulfilling one of the two requirements, providing tuning methods that are optimal to servo-control or to ~~regulation-control~~ regulatory

well known that if we optimise the closed-loop transfer function for a step-response specification, the performance with respect to load-disturbance attenuation can be very poor and vice-versa Arrieta and Vilanova (2010). Therefore it is desirable to get a compromise design, between servo/regulation when using 1-DoF controller.

Tuning is usually a compromise between performance and robustness. In fact, information about the process to be controlled takes the form of a model but this is always incomplete. Therefore, robustness is needed in order to preserve the basic properties that the model-based tuning provides. Among them, stability of the controlled system is a first need. Also, to minimise the degradation of the performance is desirable. As a basic tradeoff, as more robustness is imposed, the model-based tuning tends to provide lower performance. This is why some tunings focuses exclusively on loop performance, whereas others are aimed to ensure robust stability, or a compromised mix of both, etc.

3.2.1 Servo vs. Regulation

When tuning standard PID controllers, it is hard to achieve good tracking and fast disturbance rejection at the same time. From a specific tuning, if we want to improve disturbance rejection and get it faster, this requires more gain inside the bandwidth, which can only be achieved by increasing the slope near the crossover frequency. As a larger slope means getting closer to the critical (-1, 0i) point, this typically comes at the expense of more overshoot in response to ~~the~~ set-point changes. Therefore worsening the set-point tracking. This is a frequency domain reasoning that is sometimes not familiar. Therefore, the use of time domain indexes like the ones presented above. A relatively low IAE corresponds to a relatively fast closed loop response and relatively low oscillatory behaviour in the controlled variable (also represented by low values in the control effort measure).

As we already know, the closed loop transfer function between the set-point and error is different from that of the load disturbance and error; therefore a low IAE in fast tracking task leads to slow-moving behaviour with high IAE in the load disturbance rejection; conversely, a quick reaction to the disturbance means high overshoot in response to set-point step change, therefore increasing its IAE. This is a well known effect when zero-pole cancellation occurs in the closed loop transfer function. A solution usually taken for set-point tunings. It uses to work well for the operation as a servo control system but as the closed-loop relation from the load disturbance to the error still contains the process modes. Therefore explaining the regulation operation does not need to exhibit good performance measures.

It is usual practice, for One-Degree-of-Freedom (1-DoF) controllers, to relate the tuning method to the expected operation mode for the control system, *servo* or *regulation*. Therefore, controller settings can be found for optimal set-point or load-disturbance responses. This fact allows better performance of the controller when the control system operates on the selected tuned mode but, a degradation in the performance is expected when the tuning and operation modes are different.

Obviously there is always the need to choose one of the two possible ways to tune the controller, for set-point tracking or load-disturbances rejection. In the case of 1-DoF PID, tuning can be optimal just for one of the two operation modes.

In order to show how the performance of a system can be degraded when the controller is not operating according to the tuned mode, an example is provided. This motivates the analysis the servo/regulation *trade-off*.

Consider the following plant transfer function, taken from Zhuang and Atherton (1993), and the corresponding FOPDT approximation

$$P_1(s) = \frac{e^{-0.5s}}{(s+1)^2} \approx \frac{e^{-0.99s}}{1+1.65s} \quad (3.6)$$

PID controller parameters are found in Zhuang and Atherton (1993) by application of the ISE tuning formulae for optimal set-point and load-disturbance. Figure 3.1 shows the performance of both settings when the control system is operating in both, servo and regulation mode. It can be appreciated that the load-disturbance response of the set-point tuning (*sp*) is closer to the optimal regulation one than the load-disturbance tuning (*ld*) to the optimal servo tuning. Therefore the observed Performance Degradation is larger for the load-disturbance tuning. It is needed to search for a compromise among both tunings. If just one has to be taken, then it seems better to choose the set-point settings. However, as conflicting objectives, a more complex approach is needed in order find the best tradeoff (whatever best means).

3.2.2 Performance vs. Robustness

Robustness is an important attribute for control systems, because the design procedures are usually based on the use of low-order linear models identified at the closed-loop operation point. If only the system performance is taken into account, using by example an integrated error criteria (IAE, ITAE or ISE), or a time response characteristic (overshoot, rise-time or settling-time for example), as in Huang and Jeng (2002); Tavakoli and Tavakoli (2003), the resulting closed-loop control system probably will have a very poor robustness. On the other hand, if the system is designed to have good robustness, as in Hägglund and Åström (2008), and if the performance of the resulting system is not evaluated, the designer will not have any indication of the *cost* of having such highly robust system. System performance and robustness were take into account in Shen (2002); Tavakoli et al (2005), optimizing its IAE or ITAE performance but guarantee only the usually accepted minimum level of robustness ($M_S = 2$). Therefore, the design of the closed-loop control system must take into account the system performance to load-disturbance and set-point changes and its robustness to variation of the controlled process characteristics, preserving the well-known *trade-off* between all these variables.

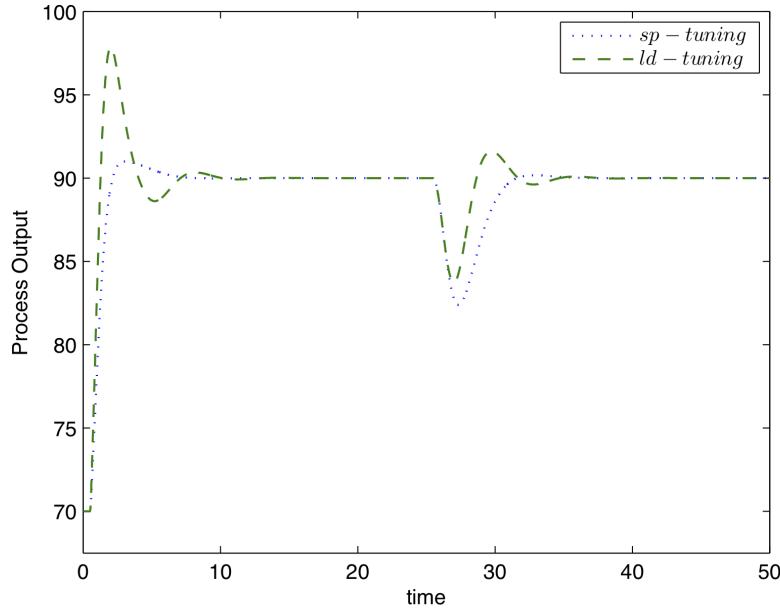


Fig. 3.1: Process responses for servo and regulation for system 3.6.

At this point, we can recover the CSTR reactor example of the previous chapter. In order to illustrate the performance / robustness tradeoff, we consider here the tuning of a PID controller in order to achieve different levels of robustness. The MoReRT method from Alfaro and Vilanova (2016). As the MoReRT allows a desired robustness level to be imposed as design specification, two cases have been considered here: (a) for $M_S = 2.0$ the controller parameters are: $K_p = 3.335$, $T_i = 0.685\text{min}$, $T_d = 0.181\text{min}$, (b) for $M_S = 1.6$, the controller parameters are: $K_p = 2.372$, $T_i = 0.663\text{min}$, $T_d = 0.162\text{min}$. As it can be seen, robust designs, corresponding to lower values of M_S , correspond to lower controller gains. As a result, time responses will be smoother but the performance indexes will worsen. Results are shown in table 3.1 and figure (3.2).

Table 3.1 shows the IAE performance indexes for the output concentration deviations as well as the TV for the control effort corresponding to a disturbance change at the input concentration C_A as well as to a desired change in the operating point, this is the desired output concentration C_B . It is seen that the performance is worse for the more robust design but, on the other hand, the control effort gets smoother.

C_{A_i}

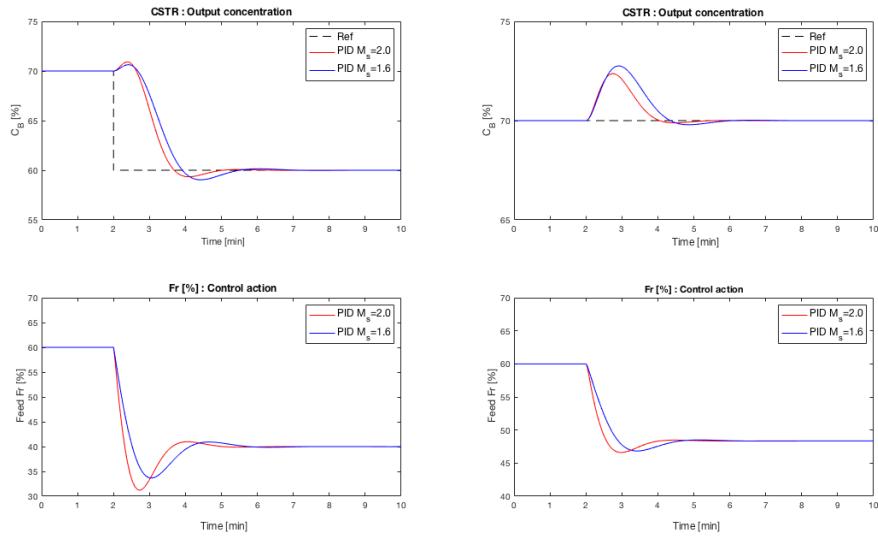


Fig. 3.2: CSTR reactor time output and control effort to a reference step change and disturbance at the inlet C_A concentration

Table 3.1: Robustness - Performance tradeoff CSTR example

	Set-point change for C_B		Disturbance at C_A concentration	
	IAE	TV	IAE	TV
$M_s = 2.0$ PID Design	12.00	35.42	2.58	15.36
$M_s = 1.6$ PID Design	14.05	34.30	3.66	13.60

3.2.3 Input vs. Output Disturbances

Disturbance attenuation is often recognised as the primary concern of a control system. Regulation of the operating conditions is the usual task work to be pursued by a feedback controller. However, much of the academic works almost concentrate on set-point experiments for controller evaluation. Therefore a controller design that emphasizes disturbance rejection rather than set-point tracking is an important design problem that, even if it has been the focus of research it may have not received the appropriate attention. Indeed much of the design approaches as well as application and/or simulation examples provided in academic works almost concentrate on set-point experiments for controller evaluation. Even those that explicitly concentrate on the disturbance attenuation problem, usually concentrate on input load disturbances.

There is however a disturbance attenuation consideration not taken into account, as far as the knowledge of the authors concern. This is that of considering a dif-

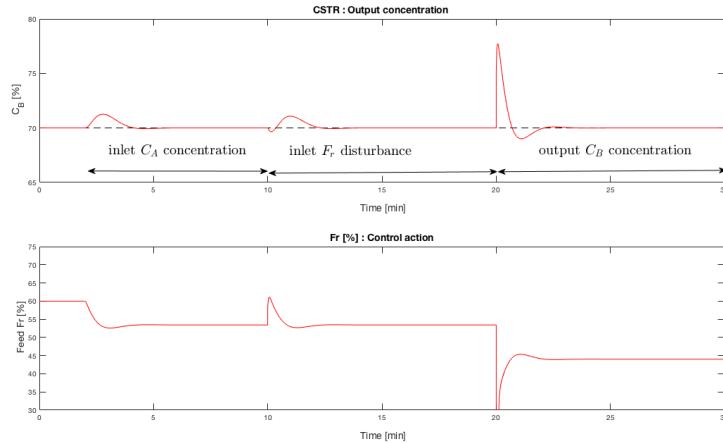


Fig. 3.3: CSTR output C_B concentration in response to different disturbances.

ferent load disturbance dynamics path. As mentioned in Shinskey (2002), there are some processes that exhibit different dynamics in the load path. These include heat exchangers, where the load can enter the tube bundle whereas the manipulated flow enters the shell (or vice versa), and distillation columns, where the load is the feed and the manipulated flow is boil up or reflux.

As a matter of a simple example, the ~~example~~ controller used in the previous section, the one with $M_S = 2.0$ is faced here to three different disturbances:

- A disturbance that enters at the process input, a load disturbance. This is exemplified here by a 10% disturbance at the inlet flow rate F_r .
- A disturbance that enters at an intermediate point of the process dynamics.. This is exemplified here by a 10% disturbance at the inlet concentration of the A component. C_A
- A disturbance that affects directly at the process output. This is exemplified here by a 10% disturbance at the output concentration of the B component. C_B

As it can be seen in figure (3.3), even the size of the disturbance is the same in all three situations, the response of the controller is different. In case of the output disturbance, its dynamics is commonly associated to the one for a reference change. In fact, the disturbance signal enters at the same point in the block diagram (except for eventual sensor measurement noise). Therefore, depending on the control system specifications, the attenuation of one disturbance or the other requires different considerations. Adding, in addition to the other two tradeoffs just presented above, a third potential source of conflicting objectives that motivates the need for a multiobjective approach.

References

- Alcantara S, Vilanova R, Pedret C (2013) PID control in terms of robustness/performance and servo/regulator trade-offs: A unifying approach to balanced autotuning. *Journal of Process Control* 23(4):527–542
- Alfaro VM, Vilanova R (2016) Model-Reference Robust Tuning of PID Controllers. *Advances in Industrial Control*, Springer International Publishing, Cham, DOI 10.1007/978-3-319-28213-8, URL <http://link.springer.com/10.1007/978-3-319-28213-8>
- Arrieta O, Vilanova R (2010) Performance degradation analysis of controller tuning modes: application to an optimal PID tuning. *Int J Innov Comput Inf Control* 6(10):4719–4729
- Arrieta O, Visioli A, Vilanova R (2010) PID autotuning for weighted servo/regulation control operation. *Journal of Process Control* 20(4):472–480, DOI 10.1016/j.jprocont.2010.01.002
- Åström KJ, Hägglund T (2006) Advanced PID Control. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, USA
- Hägglund T, Åström KJ (2008) Revisiting The Ziegler-Nichols Tuning Rules For Pi Control. *Asian Journal of Control* 4(4):364–380, DOI 10.1111/j.1934-6093.2002.tb00076.x, URL <http://doi.wiley.com/10.1111/j.1934-6093.2002.tb00076.x>
- Huang HP, Jeng JC (2002) Monitoring and Assessment of Control Performance for Single Loop Systems. *Industrial & Engineering Chemistry Research* 41(5):1297–1309, DOI 10.1021/ie0101285, URL <https://pubs.acs.org/doi/10.1021/ie0101285>
- Shen JC (2002) New tuning method for PID controller. *ISA Transactions* 41(4):473–484, DOI 10.1016/S0019-0578(07)60103-7, URL <https://linkinghub.elsevier.com/retrieve/pii/S0019057807601037>
- Shinskey FG (2002) Process Control: As Taught vs as Practiced. *Industrial & Engineering Chemistry Research* 41(16):3745–3750, DOI 10.1021/ie010645n, URL <https://pubs.acs.org/doi/10.1021/ie010645n>
- Tavakoli S, Tavakoli M (2003) Optimal Tuning of PID Controllers for First Order Plus Time Delay Models Using Dimensional Analysis. In: The Fourth International Conference on Control and Automation 2003 ICCA Final Program and Book of Abstracts ICCA-03, IEEE, pp 942–946, DOI 10.1109/ICCA.2003.1595161, URL <http://ieeexplore.ieee.org/document/1595161/>
- Tavakoli S, Griffin I, Fleming PJ (2005) Robust PI Controller for Load Disturbance Rejection and Setpoint Regulation. In: Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on, pp 1015–1020, DOI 10.1109/CCA.2005.1507263
- Zhuang M, Atherton DP (1993) Automatic Tuning of Optimum PID Controllers. *IEE Proc D, Control Theory and Applications* 140(3):216–224

Chapter 4

PID Controller Design

Abstract In this chapter the basics of the proportional integral derivative (PID) controller tuning is considered. First the analytical tuning methods are presented in order to have the most fundamental mathematical description of a tuning rule. Then, the tuning based on the minimization of a performance criteria is considered. This subject is important for this particular book because the methodology that is presented is based on the minimization of multiple cost functions at the same time. Then the minimization of a performance criteria is particularized to the case of PID controllers for integral cost functions.

4.1 PID Controller Tuning

The appropriate selection of the tuning parameters is one of the most important steps in the definition of a PID based control loop. This is known as the PID tuning. The selection of the PID controller parameters should be made according to the available knowledge of the process dynamics and stated performance specifications in terms of tracking and disturbance attenuation as well as desired robustness. One of the aspects that makes PID control specially appealing is the clear physical meaning associated to each one of its parameters.

Numerous studies have been made to develop assignment rules to specify PID parameters on the basis of characteristics of the process being controlled. The collected information about the process to be controlled can, in one form or another be assimilated to a model of the process. This can be referred either as a parametric process model (or, in other words, a form suitable for analyzing and simulating the closed-loop system), or concrete process data and/or measurements that in a suitable way can be directly employed to determine the PID controller parameters. There are many representative sources that can be consulted for details on a wide variety of alternative tuning rules (O'Dwyer, 2009; Vilanova and Visioli, 2012).

In what follows, a short presentation of the most usual existing methods for PID tuning are presented. The presentation will not be deep into details because it is not

- O'Byrne,
- HL & Vaidi 2012

the purpose of this book to explore existing tuning approaches, material that, on the other hand, can be accessed on numerous sources. However, it is important not to forget the existing panorama and main features of existing approaches. This will make possible, in the next section, to better understand the formulation of the PID tuning problem as a multiobjective optimization problem.

4.1.1 Analytical Tuning Methods

These approaches allow a specification of a desired closed-loop time response. This desired closed-loop behavior can take the form of desired poles for the closed-loop behavior, or a reference model the closed-loop system should try to mimic. These approaches were originated by the early works on algebraic controller design (of course in a more generic setting, not just for PID controllers) by (Ragazzini and Franklin, 1958). There are different approaches for doing that; some of them place only the closed loop poles, others also shape the zeros. In contrast, by using conventional PID controller, it may not be possible to place all of the poles, so only the dominant pole is placed for this scenario. The design notion in this class of designs is based on re-assigning the system's poles with faster modes. The drawback, however, is that some modes may become uncontrollable due to pole/zero cancellation and the performance degrades if they become excited.

The introduction of ideas on algebraic design, gave rise to the so called λ -tuning or Dahlin method (Dahlin, 1968). It is straightforward to see that for FOPTD and SOPTD models, PI and PID controllers can help achieve the desired performance. The number of poles that can be placed is equal to the number of controller parameters. Therefore these techniques can be used for process models with the maximum order of 2 if a PID controller is selected. This method, in turn, is closely related to the Smith predictor and the design method based on Internal Model Control (IMC) (Rivera et al, 1986). It is worth a special mention the fact that, for a FOPTD process, the IMC controller takes the form of a PI or a PID controller depending on the rational approximation used for the time delay. These approaches, however, use cancellation of the poles of the process, which may lead to quite undesirable responses to load disturbances, especially for processes with very large time constants. In Chien and Fruehauf (1990) a modification is presented that does not cancel the poles of the process, while Skogestad (2003), presents a variation of the IMC controller, applied to PI and PID controller tuning, denominated SIMC, in which the cancellation is avoided by means of a redefinition of the integral mode for the cases of systems dominated by large time constants.

In Chen and Seborg (2002) proposed a modification of the direct synthesis method adapted to disturbance rejection instead of set-point change. Tuning rules are provided for a wide range of process models. Following these lines, in 2007 Shamsuzzoha and Lee Shamsuzzoha and Lee (2008), reported that IMC demonstrates sluggish disturbance rejection, especially when the deadtime to time constant

A robust version of this approach is
presented in Vilanova ISA (2018)

ratio is small. To alleviate this problem they proposed an IMC-PID tuning method for improved disturbance rejection.

One of the advantages of IMC is the introduction of the desired closed loop time constant which can be used by operators to manipulate the degree of robustness. In Vilanova (2008) proposed a robust IMC based ISA tuning rule for set-point tracking. The tuning introduces two user defined parameters and also provides an automatic tuning rule.

4.1.2 Tuning based on Minimisation of Performance Criteria

The methods based on the application of optimization techniques are an alternative to the analytical methods. The basic idea is to try to capture different aspects of the desired operation in closed loop under the signature of a determined cost functional to be minimized. In Corripio (2001) and in Shinskey (1994), for example, controllers are optimized with respect to integral error criteria such as the ISE, IAE and ITAE.

However, among the set of tuning rules in this category, ITAE is claimed to yield better performance Ogata (1996). The first works in using optimization of integral criteria for deriving tuning rules where the ones of López et al (1967) and Rovira et al (1969) where, coming from the load disturbance based tunings of Ziegler-Nichols, tunings are provided for both load disturbance and set-point as different closed-loop operation modes. More recently, optimal tunings have been proposed (Arrieta et al, 2010) for a balanced operation among both modes when two degrees of freedom are not available or, when it is not clear the predominant operation mode the control system will work on.

Quite simple tuning rules, for different variants of the integral criteria are provided in Zhuang and Atherton (1993). The corresponding version for unstable and integrating systems is provided in Vissioli (2001). More recently, thanks in part to greater accessibility to optimization routines, powerful software and computing power, there have appeared approaches of Multiobjective optimization, such as Herreros et al (2002) and Toivonen and Totterman (2006), where a generic approach is presented and its application to the particular case of a PID controller exemplified.

The application of these optimization strategies, although effective, relies on the use of fairly complex numerical routines and not results. In general, in tuning rules as a solution of the problem. Through its application, you get the controller tuning as the solution to the optimization problem. However, as presented in some recent works, by solving the optimization problem for well defined process families and by interpolating the results, it is possible to obtain tuning formulae that give the optimal PID gains based on the process parameters.

4.1.3 Tuning Rules for Robustness

Tuning rules designed specifically to achieve a closed-loop with some robustness guarantees. Whereas there are some tuning approaches that provide tuning parameters that affect the system robustness (this is the case, for example of IMC control) they main aim is not to ensure a robust closed-loop system. Even if their application may derive in a control system with some robustness properties, its achievement will always be indirect. On the other hand, in recent years, there has been an increasing interest in including robustness. Starting point are the well-known design strategies based on setting the gain and phase margin, initiated in Åström and Hägglund (1984) that have given rise to numerous variants and extensions. In this case, the design parameter or specification is directly measuring the desired robustness for the closed-loop system. Later on, Ho et al (1995) proposed tuning rules for PID controllers for gain and phase specifications.

As previously mentioned, it was within the IMC approach that the work of Vilanova (2008) introduced robustness considerations into the formulation of the autotuning expressions. These ideas conducted lately to a series of works (Alcantara et al, 2010, 2013) where the robustness was explicitly considered as part of the design and by taking into account the robustness/performance tradeoff (Alfaro and Vilanova, 2013b).

The robustness idea has evolved in such a way that today it is common use to include a robustness constraint or consideration in whatever approach. One of the measures that has gained more popularity today is the maximum of the sensitivity function (commonly called M_S) as a reasonable robustness measure. It is also possible to distinguish between approaches that are attempting to achieve a closed-loop with a particular value of M_S and more flexible approaches providing tuning rules directly parameterized by the target M_S value (Arrieta and Vilanova, 2012; Vilanova and Visioli, 2012).

The robustness constraint has also been incorporated into more elaborated methods such as the Model reference Robust Tuning (MoReRT) approach (Alfaro and Vilanova, 2012). Such method incorporates a model reference based design within an optimization procedure and with the mentioned robustness constraint. Robust tuning rules are provided for all the most common process dynamics as well as different levels of robustness. The work also allows PID formulations where the reference and output signals are filtered. The filters are considered as parts of the design (Alfaro and Vilanova, 2013a).

4.2 Formalization of PID tuning as a multiobjective optimization problem

4.2.1 Cost Function and Constraint Selection

When dealing with PID tuning, it is common to define a metric either to optimize the parameters, or as a measure to check how well the tuning behaves with respect to other set of parameters.

In theory, to formulate the multiobjective optimization (MOO) problem, any cost function may be used. However, it is common to select cost functions that are contradictory to each other, which is something that arises naturally when dealing with real designs.

For example in Sabina Sánchez et al (2017) the integral of the absolute value of the error (IAE) and the total variation are used as contradictory cost functions. The total variation represents the control effort and is related to the robustness of the closed system, therefore, using these two cost functions, the compromise is made between performance and robustness. Something similar is done in Pierezan et al (2014), where a particle swarm optimization technique is applied to the tuning of PID controllers for a robotic manipulator. In Zhou et al (2018) also a two objectives optimization is done with similar cost functions, the difference consist in that a multivariable control loop is tackled at the same time using a compound weight sum of an IAE metric for three setpoints and a function related to the energy consumption based on the variation of the input variables. *has to conform as*

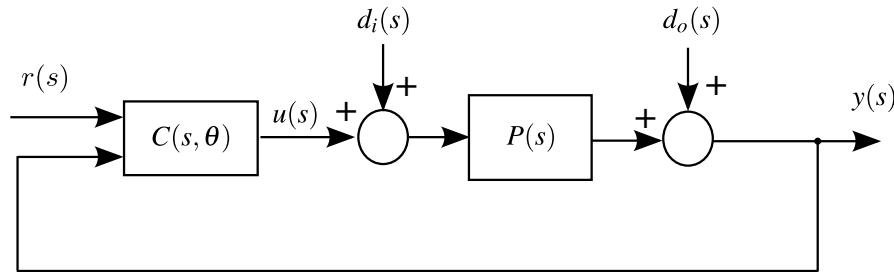
The cost functions~~s~~ not necessarily ~~is~~ an integral function. For example in Abbas and Sawyer (1995) the cost functions are the percent overshoot and the rise time, but the authors also indicate that the settling time and the maximum controller output can be also considered. A more classical approach takes into account frequency domain measures as phase and gain margins (Åström and Hägglund, 1984; Ho et al, 1995). In HUANG et al (2008) the \mathcal{H}_∞ norm is used in different frequency bands of different sensitivity functions to define several cost functions which are then used in a multi-objective approach.

min'stula
srch
o
o

Other authors tries to combine integral cost functions with other time domain measures, as in Chiha et al (2012) where the settling time, overshoot, rise time, IAE, integral of the square error (ISE) and integral time absolute error (ITAE) are taken into account in a weighted sum and optimized using an Ant Colony algorithm approach. *subject*

Definitively, the selection of the cost function is widely open and it entirely depends on the necessity of the task at hand. According to Shinskey (2002), the main objective of a process controller is to mitigate the effects of load of disturbances, setpoint changes are considered of secondary importance. The same author~~s~~ states that:

“minimum IAE is a preferred criterion that includes integral error and penalizes continued cycling. Minimum-IAE tuning also tends to be consistent with minimum error.”



's smaller!

Fig. 4.1: Feedback control loop.

control performance with respect to

Following this reasoning, in this work we propose to use integral cost functions to measure input and output disturbances and setpoint changes in a multi-objective framework, taking the maximum sensitivity as a constraint to find the optimal controller tuning in a Pareto sense.

The controller is considered to be a two degrees of freedom (2DoF) PID controller. Even with this topology, the cost functions that results of considering all three sources of disturbances are contradictory among them because minimizing one of them does not minimize the other functions, in fact you may find that the optimal controller for one cost function yields to the maximum value of other function. These details are covered in Section 5.1 where the multi-objective framework is stated.

4.2.2 PID tuning problem formulation for integral cost functions

A feedback control system like the one shown in Figure 4.1, also called ~~closed-loop control system~~, is designed to keep certain relationship between the process output $y(s)$ and the reference input $r(s)$. For such task, the difference between those signals is used to compute the control signal $u(s)$ needed in order to achieve $y(s) = r(s)$.

In Figure. 4.1, $C(s, \theta)$ is the 2DoF PID controller with parameters:

θ major \approx

$$\theta = [K_p \ T_i \ T_d \ \beta]^T$$

with K_p the proportional gain, T_i the integral time constant, T_d is the derivative time constant, β the weight on the reference signal. $P(s)$ represents the controlled process, modeled as a ~~overdamped~~ second ~~order plus~~ ~~time delay~~ (ODSOPTD) plant, with a transfer function of the form:

$$P(s) = \frac{Ke^{-Ls}}{(Ts+1)(aTs+1)}, \quad (4.1)$$

where K , L and T , correspond to the static gain, the time delay and main time constant respectively. The other pole of the system is represented with a time constant

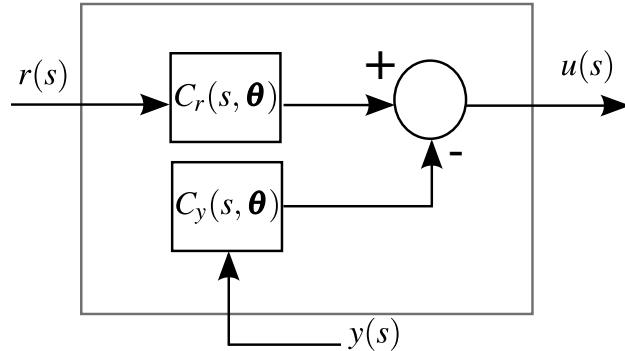


Fig. 4.2: Representation of the 2DoF controller.

that is fraction of T , therefore $0 \leq a \leq 1$. $d_i(s)$ represent the input disturbance while $d_o(s)$ is the output disturbance.

The relationship between the control signal, the reference and the process output is given by:

$$u(s) = C_r(s, \theta)r(s) - C_y(s, \theta)y(s), \quad (4.2)$$

where the part applied to the reference signal is given by:

$$C_r(s, \theta) = K_p \left(\beta + \frac{1}{T_i s} + \gamma \frac{T_d s}{\alpha T_d s + 1} \right), \quad (4.3)$$

and the part applied to the process output is:

$$C_y(s, \theta) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1} \right). \quad (4.4)$$

It is common to set $\alpha = 0.1$ and $\gamma = 0$. For this reason, the controller parameter vector is give as $\theta = [K_p \ T_i \ T_d \ \beta]^T$. A detailed depiction of the controller transfer function is presented in Figure 4.2. *description*

To simplify the analysis, the model of the controlled process is normalized:

$$\hat{s} = Ts, \ \tau_0 = \frac{L}{T}, \ \tau_i = \frac{T_i}{T}, \ \tau_d = \frac{T_d}{T}, \ \kappa_p = K_p K.$$

Then the normalized parameters of the controller become $\theta = [\kappa, \tau_i, \tau_d, \beta]^T$. and the response of the controlled system is computed as:

$$y(\hat{s}) = y_r(\hat{s}) + y_{di}(\hat{s}) + y_{do}(\hat{s}), \quad (4.5)$$

where $y_r(\hat{s})$, is the output response to a change in the setpoint $r(\hat{s})$, $y_{di}(\hat{s})$ is the response to a change in the input disturbance signal $d_i(s)$ and $y_{do}(\hat{s})$ is the response to a change in the ouput disturbance signal $d_o(s)$. From Figure 4.1 and Figure 4.2,

$$s \rightarrow \hat{s}$$

these signals can be computed as:

$$\begin{aligned} y_r(\hat{s}) &= \frac{P(\hat{s})C_r(\hat{s}, \boldsymbol{\theta})}{1+P(\hat{s})C_y(\hat{s}, \boldsymbol{\theta})} r(\hat{s}) \\ y_{di}(\hat{s}) &= \frac{C_r(\hat{s}, \boldsymbol{\theta})}{1+P(\hat{s})C_y(\hat{s}, \boldsymbol{\theta})} d_i(\hat{s}) \\ y_{do}(\hat{s}) &= \frac{1}{1+P(\hat{s})C_y(\hat{s}, \boldsymbol{\theta})} d_o(\hat{s}) \end{aligned}$$

Robustness is an indication of the relative stability of the controlled system and it measure the ability of the controller to keep the closed loop stable despite the variation in the process dynamics. A metric of the degree of relative stability is the maximum sensitivity M_s given by:

$$M_s = \max_{\omega} \left\{ \frac{1}{|1+C_y(j\omega)P(j\omega)|} \right\} \quad (4.6)$$

~~The recommended value range is $1.2 \leq M_s \leq 2.0$.~~

As it is widely established, the controller tuning can be solved as a multi-objective optimization problem (Gambier and Badreddin, 2007). One common indicator of performance, is the IAE given by:

$$J(\boldsymbol{\theta}) = \int_0^{\infty} |e(t, \boldsymbol{\theta})| dt. \quad (4.7)$$

The error signal $e(t, \boldsymbol{\theta})$ it is calculated using:

$$e(t, \boldsymbol{\theta}) = r(t) - y(t, \boldsymbol{\theta}). \quad (4.8)$$

When (4.7) is computed for a step change in the reference signal, the cost function becomes $J_r(\boldsymbol{\theta})$; for an input disturbance response, the function is defined as $J_{di}(\boldsymbol{\theta})$ and finally, for an output disturbance response, the cost function is named as $J_{do}(\boldsymbol{\theta})$.

When the output of the plant is disturbed only by the step change in $d_i(s)$, the error signal then becomes:

$$e_d(t) = -y_{di}(t). \quad (4.9)$$

And then, the cost function $J_{di}(\boldsymbol{\theta})$ is computed as:

$$J_{di}(\boldsymbol{\theta}) = \int_0^{\infty} |-y_{di}(t, \boldsymbol{\theta})| dt, \quad (4.10)$$

On the other hand, if the disturbance comes only from a step signal in $d_o(s)$, the cost function that has to be computed is $J_{do}(\boldsymbol{\theta})$ as:

$$J_{do}(\boldsymbol{\theta}) = \int_0^{\infty} |-y_{do}(t, \boldsymbol{\theta})| dt. \quad (4.11)$$

Finally, if the setpoint is the only source of disturbance for the plant, the corresponding cost function $J_r(\boldsymbol{\theta})$ is computed as:

$$J_r(\boldsymbol{\theta}) = \int_0^{\infty} |r(t) - y_r(t, \boldsymbol{\theta})| dt. \quad (4.12)$$

The problem of minimizing $J_r(\boldsymbol{\theta})$, $J_{di}(\boldsymbol{\theta})$ and $J_{do}(\boldsymbol{\theta})$ at the same time can be posed as a MOO problem. In addition, since in an industrial environment the robustness is very important, the obtained parameters are constrained to always satisfy $M_s \leq M_{s,max}$, where $M_{s,max}$ is the allowed limit of the maximum sensitivity. The combined cost function (vector of cost functions) then becomes:

$$\mathbf{J}(\boldsymbol{\theta}) = [J_{di}(\boldsymbol{\theta}), J_{do}(\boldsymbol{\theta}), J_r(\boldsymbol{\theta})]^T, \quad (4.13)$$

and solved by finding all possible optimal solutions of:

$$\begin{aligned} \mathbf{J}(\boldsymbol{\theta}^*) &= \min_{\boldsymbol{\theta}} \mathbf{J}(\boldsymbol{\theta}), \\ \text{s.t. } M_s &\leq M_{s,max} \end{aligned} \quad (4.14)$$

In general, it is not possible to find a set of parameters $\boldsymbol{\theta}$ that minimizes all those three functions at the same time. Such impossible point where all the cost functions are optimal is called the utopia point. As it names states, the utopia point is impossible to reach because optimizing one of the cost function always produce a degradation in the other remaining functions.

The particular cases that are the closest to the utopia point, are part of what is called the Pareto frontier. This set of possible solutions are considered to be equally optimal because there is no possibility to improve one of the functions without degrading the others.

This means that solving the solution of the control problem does not give a single solution, instead a family of optimal controller tunings are found.

References

- Abbas A, Sawyer P (1995) A multiobjective design algorithm: Application to the design of siso control systems. *Computers & Chemical Engineering* 19(2):241–248, DOI 10.1016/0098-1354(94)00044-O, URL <https://linkinghub.elsevier.com/retrieve/pii/009813549400044O>
- Alcantara S, Pedret C, Vilanova R, Zhang WD (2010) Simple Analytical min-max Model Matching Approach to Robust Proportional-Integrative-Derivative Tuning with Smooth Set-Point Response. *Industrial & Engineering Chemistry Research* 49(2):690–700
- Alcantara S, Vilanova R, Pedret C (2013) PID control in terms of robustness/performance and servo/regulator trade-offs: A unifying approach to balanced autotuning. *Journal of Process Control* 23(4):527–542
- Alfaro VM, Vilanova R (2012) 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, DOI 10.1016/J.JPROCONT.2012.01.001, URL <https://www.sciencedirect.com/science/article/pii/S0959152412000042>

- Alfaro VM, Vilanova R (2013a) Robust tuning of 2DoF five-parameter PID controllers for inverse response controlled processes. *Journal of Process Control* 23(4):453–462, DOI 10.1016/j.jprocont.2013.01.005, URL <https://www.sciencedirect.com/science/article/pii/S0959152413000152> <https://linkinghub.elsevier.com/retrieve/pii/S0959152413000152>
- Alfaro VM, Vilanova R (2013b) Simple Robust Tuning of 2DoF PID Controllers From A Performance/Robustness Trade-off Analysis. *Asian Journal of Control* 15(6):1700–1713, DOI 10.1002/asjc.653, URL <http://dx.doi.org/10.1002/asjc.653>
- Arrieta O, Vilanova R (2012) Simple Servo/Regulation Proportional-Integral-Derivative (PID) Tuning Rules for Arbitrary Ms-Based Robustness Achievement. *Industrial & Engineering Chemistry Research* 51(6):2666–2674
- Arrieta O, Visioli A, Vilanova R (2010) PID autotuning for weighted servo/regulation control operation. *Journal of Process Control* 20(4):472–480, DOI 10.1016/j.jprocont.2010.01.002
- Åström K, Hägglund T (1984) Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica* 20(5):645–651, DOI 10.1016/0005-1098(84)90014-1, URL <https://linkinghub.elsevier.com/retrieve/pii/0005109884900141>
- Chen D, Seborg DE (2002) PI/PID Controller Desing Based on Direct Synthesis and Disturbance Rejection. *Ind Eng Chem Res* 41:4807–4822
- Chien IL, Fruehauf PS (1990) Consider IMC Tuning to Improve Controller Performance. *Chemical Engineering Progress* 86:33–41
- Chiha I, Liouane N, Borne P (2012) Tuning PID Controller Using Multiobjective Ant Colony Optimization. *Applied Computational Intelligence and Soft Computing* 2012:1–7, DOI 10.1155/2012/536326, URL <http://www.hindawi.com/journals/acisc/2012/536326/>
- Corripio AB (2001) Tuning of Industrial Control Systems, 2nd edn. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, USA.
- Dahlin EG (1968) Designing and Tuning Digital Controllers. *Instrumentation and Control Systems* 41(6):77–81
- Gambier A, Badreddin E (2007) Multi-objective Optimal Control: An Overview. In: 2007 IEEE International Conference on Control Applications, pp 170–175, DOI 10.1109/CCA.2007.4389225
- Herreros A, Baeyens E, Peran JR (2002) Design of PID-type controllers using multiobjective genetic algorithms. *ISA Transactions* 41(4):457–472
- Ho WK, Hang CC, Cao LS (1995) Tuning PID Controllers Based on Gain and Phase Margin Specifications. *Automatica* 31(3):497–502
- HUANG L, WANG N, ZHAO JH (2008) Multiobjective Optimization for Controller Design. *Acta Automatica Sinica* 34(4):472–477, DOI 10.3724/SP.J.1004.2008.00472, URL <https://linkinghub.elsevier.com/retrieve/pii/S1874102908600245>
- López AM, Miller JA, Smith CL, Murrill PW (1967) Tuning Controllers with Error-Integral Criteria. *Instrumentation Technology* 14:57–62
- O'Dwyer A (2009) Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London, UK
- Ogata K (1996) Modern control engineering; 3rd ed. Prentice-Hall electrical engineering, Prentice-Hall, Englewood Cliffs, NJ
- Pierezan J, Ayala HH, Da Cruz LF, Freire RZ, Dos S Coelho L (2014) Improved multiobjective particle swarm optimization for designing PID controllers applied to robotic manipulator. *IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CICA 2014: 2014 IEEE Symposium on Computational Intelligence in Control and Automation, Proceedings* pp 1–8, DOI 10.1109/CICA.2014.7013255
- Ragazzini JR, Franklin GF (1958) Sampled-Data Control Systems. SMcGraw-Hill: New York
- Rivera DE, Morari M, Skogestad S (1986) Internal Model Control. 4. PID Controller Desing. *Ind Eng Chem Des Dev* 25:252–265
- Rovira AJA, Murrill PW, Smith CL (1969) Tuning controllers for setpoint changes. *Instruments and Control Systems* 42:67–69

- Sabina Sánchez H, Visioli A, Vilanova R (2017) Optimal Nash tuning rules for robust PID controllers. *Journal of the Franklin Institute* 354(10):3945–3970, DOI 10.1016/j.jfranklin.2017.03.012
- Shamsuzzoha M, Lee M (2008) Analytical design of enhanced PID filter controller for integrating and first order unstable processes with time delay. *Chemical Engineering Science* 63:2717–2731
- Shinskey FG (1994) Feedback controllers for the process industries. McGraw-Hill Professional
- Shinskey FG (2002) Process Control: As Taught vs as Practiced. *Industrial & Engineering Chemistry Research* 41(16):3745–3750, DOI 10.1021/ie010645n, URL <https://pubs.acs.org/doi/10.1021/ie010645n>
- Skogestad S (2003) Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13(4):291–309, DOI 10.1016/S0959-1524(02)00062-8, URL <https://www.sciencedirect.com/science/article/pii/S0959152402000628> <https://linkinghub.elsevier.com/retrieve/pii/S0959152402000628>
- Toivonen HT, Totterman S (2006) Design of fixed-structure controllers with frequency-domain criteria: a multiobjective optimisation approach. *IEE Proc D, Control Theory and Applications* 153(1)
- Vilanova R (2008) IMC based Robust PID design: Tuning guidelines and automatic tuning. *Journal of Process Control* 18:61–70
- Vilanova R, Visioli A (2012) PID Control in the Third Millennium. *Advances in Industrial Control*, Springer London, London, DOI 10.1007/978-1-4471-2425-2, URL <http://link.springer.com/10.1007/978-1-4471-2425-2>
- Visioli A (2001) Optimal tuning of PID controllers for integrating and unstable processes. *IEE Proc - Control Theory Appl* 148(1):180–184
- Zhou X, Zhou J, Yang C, Gui W (2018) Set-Point tracking and Multi-Objective Optimization-Based PID control for the goethite process. *IEEE Access* 6(2):36,683–36,698, DOI 10.1109/ACCESS.2018.2847641
- Zhuang M, Atherton DP (1993) Automatic Tuning of Optimum PID Controllers. *IEE Proc D, Control Theory and Applications* 140(3):216–224

Chapter 5

Multiobjective optimization

Abstract Before tackling the problem of multiple criteria for proportional integral derivative (PID) tuning, the multi-objective optimization is explained in general. First the basic formulation of the optimization problem is presented with the introduction of the Pareto front concept. The methodology chosen to solve the multiobjective optimization problem is to transform the multi-criteria situation into a single scalar cost function. However it is known that this scalarization procedure may not yield to a good Pareto front. For this reason several scalarization methods are presented and compared. The Pareto front found using this scalarization methods may be used as just data that can be later used or may be directly used as part of a decision tool useful to select the final solution to the problem.

5.1 Formalization of the multiobjective optimization problem

A multiobjective optimization problem (MOOP) arises when, in order to solve a given problem or design, it is necessary to optimize several cost functions at the same time.

In general, these cost functions depend on the same variables and usually are in conflict. In addition, they may be independent of one another, that is, the value of the variables that optimize one of the functions do not necessarily optimize the other cost functions.

In those cases, given a set of cost functions:

$$\mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), F_3(\mathbf{x}), \dots, F_k(\mathbf{x})]^T \quad (5.1)$$

that depends on n different variables $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, $x_i \in \mathbf{X}$, where \mathbf{X} is the feasible decision space. The MOOP may be formulated as follows (Marler and J.S., 2004):

*is x_i o x !
mas dejo si $x \in X$*

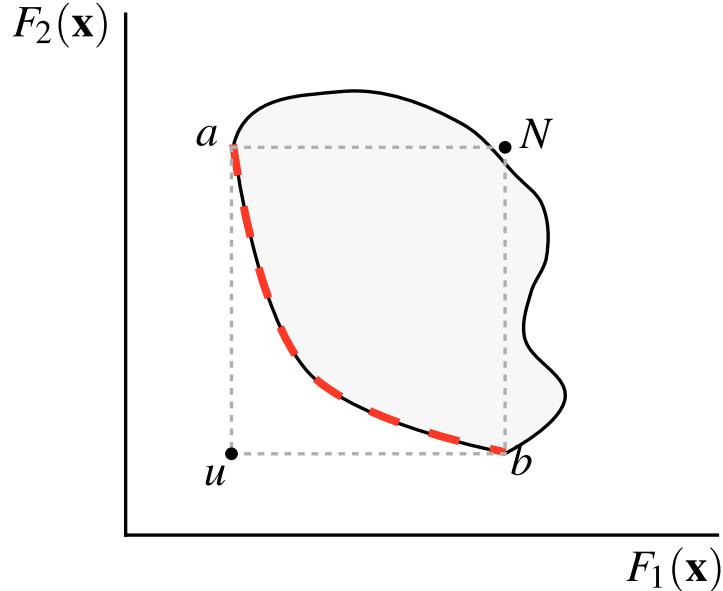


Fig. 5.1: All possible solutions and the Pareto front in the function space.

$$\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}), \quad (5.2a)$$

s.t.

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \quad (5.2b)$$

$$h_l(\mathbf{x}) = 0, \quad l = 1, 2, \dots, e \quad (5.2c)$$

where $g_j(\mathbf{x})$ is the j -th inequality constraint and $h_l(\mathbf{x})$ is the l -th equality constraint.

5.1.1 Definition of the Pareto front

simultaneously

In general, it is not possible to find a set of variables values that minimizes all F functions. In fact, the optimization problem in (5.2) have multiple equally optimal solutions in the sense of the Pareto optimality. According to Marler and J.S. (2004):

“A point $\mathbf{x}^* \in \mathbf{X}$, is Pareto optimal iff there does not exist another point, $\mathbf{x} \in \mathbf{X}$, such that $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$, and $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$ for at least one function”

The concept of Pareto optimality is represented in figure 5.1 for a two-function multi-objective optimization. The gray area represents the feasible function space, given by the value of $F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{X}$. From all those points, only the points in the curve from “a” to “b” (marked with a thicker dash line) are Pareto optimal because there is not another point in the feasible decision space with a lower

5.1 Formalization of the multiobjective optimization problem

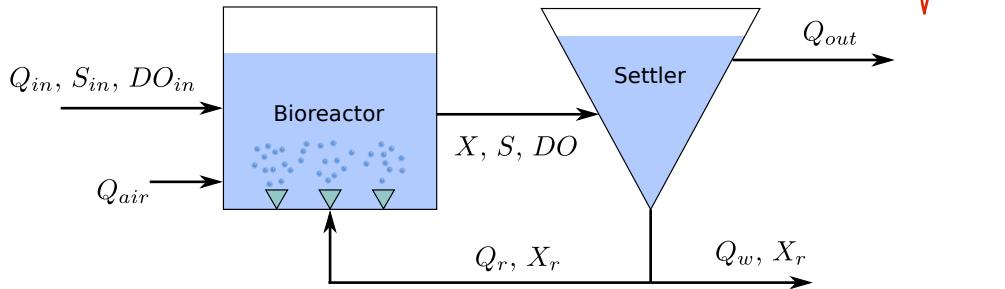


Fig. 5.2: Representation of a simple wastewater treatment process.

value of \mathbf{F} , but there is at least one point that has a lower value for either F_1 or F_2 . The curve from “a” to “b” is the Pareto front and contains all possible solutions to problem (5.2) that are Pareto optimal. These solutions are always in the edge of the feasible function space, closer to the utopia point (the “u” point in the figure). The utopia point is a point in the space where all the cost functions have their minimum value. As it can be seen from figure 5.1, this point is more likely to be outside of the feasible function space.

Points “a” and “b” are called anchor points and represent the combination of decision variables that optimizes at least one of the functions. In this case, “a” is the point where function $F_1(\mathbf{x})$ has its minimum value whereas “b” the one in which $F_2(\mathbf{x})$ has its minimum value.

Point “N” is called the pseudo-nadir point, and is defined as the point with the worst values of all the anchor points. *Cursiva?*

5.1.2 Practical example

To get an insight of the concept of Pareto Front, an industrial controlled process will be considered. In this particular case, the relationship between total variation (TV) and integral of the absolute value of the error (IAE) for the controller as regulator is studied. However, in the rest of the book, the compromises between the response for different sources of disturbances will be considered. The objective of this section is to show that the concept of Pareto front can be used with diverse cost functions. *will be*

In Section 5.1.3, different approaches to obtain the Pareto front are presented. For this particular example a “brute force” approach is taken. The process that is used as example is a wastewater treatment plant (WWTP) as presented in Figure 5.2.

The residual water after it has been used in residential commercial or industrial zones is known as wastewater. It is collected through sewers with the intention to be treated to be later deposited in receiving waters like rivers, lakes or the sea. *however*

cording to Olsson and Newell (1999) “while the primary goal of a treatment plant is to achieve an average reduction in nutrient levels, the secondary goal is disturbance rejection, to achieve good effluent quality in spite of the many disturbances”. One of the characteristics of this process is that it is subject to large variation in the influent characteristics like the substrate concentration, oxygen levels and even flow. For example in Henze et al (1997), it is shown an example were the influent flow at midday can reach up to 244% of the average flow in one day, while minimum could reach 32% of the average flow.

To reduce the substrate levels, the idea is to stimulate the grow of microorganism that consume the substrate, to later be remove in ~~a~~ settler. This biological process is known as Activated Sludge Process (ASP) and it is one of the most important methods for wastewater treatment (Henze et al, 1997). Bacteria is the most important component of the sludge. These bacteria can remove carbon components and also nitrogenous components from the influent. To control the grow of bacteria, air is pumped into the wastewater while being store in tanks.

these

As pointed out by Jeppsson (1996), from the point of view of the bacteria, the organic particles in the influent are used as its source of energy. Bacteria take the oxygen and the particles and produce other simpler compounds (methane for example). The air injected in the tanks is its main source of oxygen and therefore is the principal manipulated variable of the system. However it is common to also have anoxic tanks (that is, tanks without external oxygen) that are used to promote the growth of bacteria that takes the oxygen directly from the water in the tank. ~~This~~ anoxic tanks are used to remove the nitrogenous components. The suspended material, the sludge, is remove from the water by settlers. Part of the sludge is recirculated to the system in order to keep enough biomass in the tanks, while the rest is dispose out of the loop to be used as fertilizer.

Therefore, the basic layout of a WWTP using the ASP contains an aeration tank and a settling tank as represented in Figure 5.2. The “cleaned” water is withdrawn from the top of the settler while part of the sludge is recirculated to the bioreactor and the rest is used to produce where the treated wastewater is deposited in the receiving water while a concentrated sludge is withdrawn from the bottom (Henze et al, 1997). This concentrated sludge can be recycled in order to maintain a high density of biomass in the tanks.

One of the characteristics of WWTPs is their high energy consumption (Longo et al, 2016). The bioreactor needs electricity to produce the aeration, move the water using pumps and continuous stirring. At the same time it is necessary to cope with disturbances coming from the influent. With this train of logic, from a control perspective, it is necessary to keep the system regulated, at the same time that the energy is minimized.

In order to frame this problem mathematically, first a model has to be selected. In this case, the model first proposed in Nejjari et al (1999) and slightly modified in Han et al (2008) will be considered. This model is a fourth order non-linear set of equations found after a material balance is performed. The model is given by:

Table 5.1: Parameter values for the WWTP model

$DO_{max} = 10 \text{ mg L}^{-1}$	$S_{in} = 200 \text{ mg L}^{-1}$	$DO_{in} = 0.5 \text{ mg L}^{-1}$
$Y = 0.65$	$\mu_{max} = 0.15 \text{ mg L}^{-1}$	$r = 0.6$
$K_S = 100 \text{ mg L}^{-1}$	$K_{DO} = 2 \text{ mg L}^{-1}$	$\alpha = 0.018$
$K_0 = 0.5$	$\beta = 0.2$	

$$\begin{aligned}
\frac{dX(t)}{dt} &= \mu(t)X(t) - D(t)(1+r)X(t) + rD(t)X_r(t) \\
\frac{dS(t)}{dt} &= -\frac{\mu(t)}{Y}X(t) - D(t)(1+r)S(t) + D(t)S_{in}(t) \\
\frac{dDO(t)}{dt} &= -\frac{K_0\mu(t)X(t)}{Y} - D(t)(1+r)DO(t) \\
&\quad + \alpha Q_{air}(t)(DO_{max} - DO(t)) + D(t)DO_{in}(t) \\
\frac{dX_r(t)}{dt} &= D(t)(1+r)X(t) - D(t)(\beta + r)X_r(t) \\
\mu(t) &= \mu_{max} \frac{S(t)}{K_S + S(t)} \cdot \frac{DO(t)}{K_{DO} + DO(t)}
\end{aligned} \tag{5.3}$$

where,

- $X(t)$ is the biomass concentration
- $S(t)$ is the substrate concentration
- $DO(t)$ is the dissolved oxygen in the bioreactor
- $X_r(t)$ is the recycled biomass concentration
- $D(t)$ is the dilution rate
- $DO_{in}(t)$ is dissolved oxygen concentration in the influent
- $Q_{air}(t)$ is the aeration rate
- $\mu(t)$ biomass growth rate

and the parameters are:

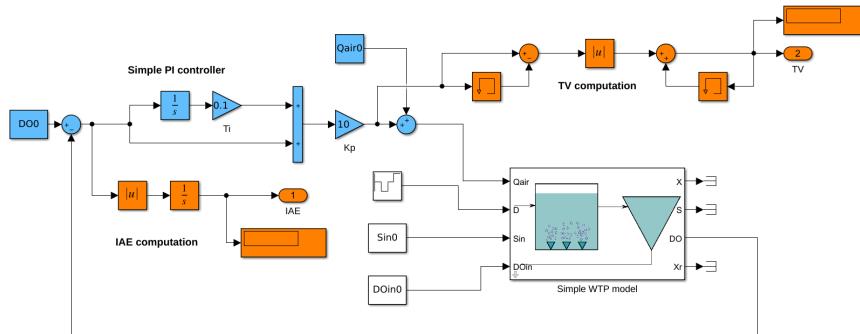
- DO_{max} is the maximum dissolved oxygen concentration
- S_{in} is the substrate concentration in the influent
- Y is the biomass yield factor
- μ_{max} is the maximum specific growth rate
- K_S is the affinity constant
- K_{DO} is the saturation constant
- α is the oxygen transfer rate
- K_0 is a model constant
- r is the recycled sludge rate
- β is the removed sludge rate

The values of all the parameters are given in Table 5.1, while the value of the variables in the selected operation point is given in Table 5.2.

The idea is to study the relationship between the IAE for the closed-loop regulation against the dilution rate disturbances and the TV of the aeration rate. The

Table 5.2: Variables values for WWTP in operation point

$Q_{air0} = 27.57$	$D_0 = 0.025$
$S_{in0} = 200 \text{ mg L}^{-1}$	$DO_{in0} = 0.5 \text{ mg L}^{-1}$
$X_0 = 298.24$	$S_0 = 10.29$
$DO_0 = 5.00$	$X_r0 = 596.47$

Fig. 5.3: Simulink implementation of the WWTP *and control system*

IAE represents the performance of the plant while the TV is an indirect measure of the energy required to accomplish that level of performance. It is expected that a good disturbance rejection response (lower IAE) implies a higher TV (which means a more aggressive control signal).

The closed-loop is controlled by means of a PI controller that manipulates the aeration rate in order to keep the dissolved oxygen concentration in the reactor around 5.0 in the presence of dilution rate disturbances. The model was implemented with an S-function and the simulation *were* performed using Simulink®.

func *blow* *→* The implementation of the model can be found in the software companion to this book and is represented in Figure 5.3. The parameters of the *WTP* can be varied by *WWTP?* means of a mask, as presented in Figure 5.4. The parameters and the initial conditions can be set manually, alternatively, an initialization script is also included, with the parameter and variables values presented in Table 5.1 and Table 5.2.

The implementation of the controller is very simple, a more complete version is used in Section 8.2, where a chemical process is used to show the multivariable approach presented in this book. The IAE and TV values are also computed within the Simulink model.

In Section 5.1.3, some methods to *produce* the front are presented, however, for this practical example, a more basic approach is taken. The idea can be summarized as follows: Varying the values of K_p and T_i within a certain range, compute both IAE and TV for all possible combinations. From this data, found the Pareto front.

This procedure is simple, however it is necessary to do a simulation of the system for every point, as presented in Figure 5.5. This figure represents 2500 simulations where the resulting IAE and TV are represented in the plane known as the function

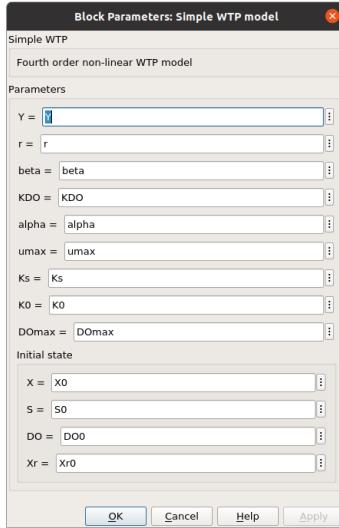


Fig. 5.4: Mask that allows the parameters of the WWTP model to be changed.

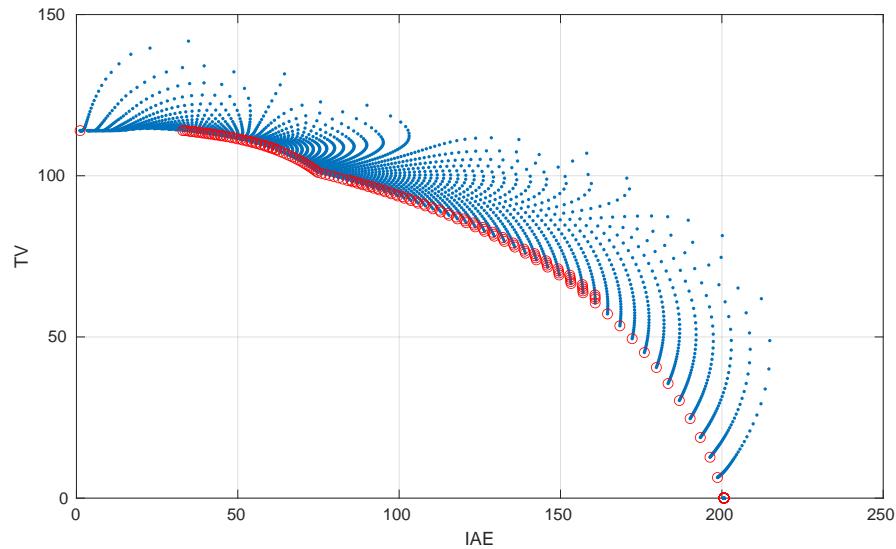


Fig. 5.5: Result of the simulation for ~~WTP~~ considering IAE and TV.

~~WWTP~~

space. One expect the Pareto to be shaped like the example presented in Figure 5.1, however this is not necessarily true for all cases, as can be verified with this particular example. Nevertheless, the Pareto arises once all the simulations are completed. All points represent~~y~~ a particular set of values of K_p and T_i . The points marked with a circle are the ones that belong~~t~~ to the Pareto. All ~~this~~ points represent~~y~~ optimal

~~these~~

values of *K_p* and *T_i* because, it is not possible to find any other solution capable to have a lower value for one of the functions without worsening the other function.

For this particular example, the left anchor point (the one with lower value of IAE) can be considered as the ~~the~~ case with the controller with the best performance but with a higher consumption of energy. Since there some points above this point, it can be said that there are more costly controllers but with worst performance. It is obvious that this set of controllers are not of our interest, that is the dominated solution can be disregarded.

On the other side, the anchor point on the right (the one with lower TV) is an interesting case, in which the controller is set in open loop (the value of *K_p* is equal to zero). The particular value of IAE for this case is totally dependent of the disturbance signal used for the simulation. However the behavior found in this example is what would be expected: better performance implies a more aggressive control signal.

However, it is clear that is not efficient to find the Pareto front using brute force. Consider that, from all the points computed, only a small fraction (8.4% to be exact) correspond to cases in the front. A lot of computer power was literally wasted. Also, note that, by varying *K_p* and *T_i* evenly, it was not possible to found an evenly distributed front. It is also important to note that not all points that were selected as part of the Pareto may not actually belong to it. If more points were found, it may be possible to find a better representation of the front but the computation needed (an the time) will not be productive.

However, once the Pareto is obtained, the user is capable to make decisions regarding the final selection of the parameters. For example, the decision maker can allow a level of degradation in IAE to obtain certain improvement on the TV. Let's define the IAE degradation α_{iae} as:

$$\alpha_{iae} = \frac{IAE - IAE_{min}}{IAE_{max} - IAE_{min}},$$

with such definition, a value of $\alpha_{iae} = 0$ represents zero degradation, that is, the case where the *IAE* value is at its minimum. On the other hand, a value of $\alpha_{iae} = 1$ represents the maximum value of *IAE* within the Pareto front, that is, the maximum possible degradation.

With this in mind and looking at the Pareto in Figure 5.5, it is clear that, to obtain an improvement on TV of about 20%, it is necessary to degrade the IAE about 50%. This may or may not be appropriate for the application. But, whichever point the decision maker takes as the final controller, if it is one of the points in the front, he or she can be sure that it is optimal.

But the efforts should be directed to find the Pareto from the beginning, not as a subproduct of a brute force task. Now, certain techniques are shown that are intended to directly find the best representation of the Pareto with as little effort as possible. Later in the book, some tools are proposed to take advantage from the Pareto, once it has been found.

5.1.3 Different approaches to obtain the Pareto front

one

In its most basic form, the Pareto front is found by performing several optimizations, each of which are computed by varying some kind of parameters. Because of this, the idea is to be able to use standard optimization methods to find each point of the Pareto. However, these standard methods are meant to solve single objective problems.

As for single objective optimization there ~~is~~ two big families of methods to solve the problem: Bio-inspired methods that use some kind of heuristic in order to find the minimum of the cost function and deterministic methods mostly based on certain gradient of the ~~function~~ *cost function*

In this section, a short review on both families is presented. However, because of the deterministic nature of the gradient based methods, in this book the later family is chosen tool for solving MOOP.

the In Zhou et al (2011) a review on multi-objective evolutionary algorithm is presented. The author indicates that several ~~of the~~ algorithms are similar to the non-dominated sorting genetic algorithm II (NSGA-II) (Deb et al, 2002). Genetic algorithms are based ~~in~~ on the idea of random mutation across generations and ~~interchange~~ *exchange* ^(?) of genes from parents to children. They also explain other kind of algorithms like particle swarm optimization which is based on the social behavior of bird flocking or fish schooling (Eberhart and Kennedy, 1995). Originally this method was employed for single function optimization, but it has been extended ~~for~~ *to* multiple cost functions. Other methods that ~~has~~ been used for solving MOOP, have a probabilistic nature like Ant Colony Optimization (Dorigo and Blum, 2005) or the Cross Entropy method (Rubinstein and Kroese, 2004). One drawback is that these methods are heuristic, and may yield different results each time they are computed. However, the main advantage is that they probably are able to find the global minimum of the functions.

Specifically, for PID control, evolutionary algorithms have been used in Reynoso-Meza et al (2013) for the multivariate process of the Wood and Berry distillation column. Another case is presented in Pierezan et al (2014) where multi-objective Particle Swarm Optimization is applied on multivariable PID controllers tuning to improve the performance of a robotic manipulator. This method was also used in Tian et al (2014) but applied to a nonlinear ~~process~~ continuous stirred tank reactor. Also in Mahdavian and Wattanapongsakorn (2014), a multi-objective optimization for PID control of a greenhouse electrical lighting system based on the cost of electricity is investigated and solved using ~~an~~ evolutionary algorithm. Multiobjective salp swarm algorithm (MSSA) with opposition based learning initialization and evolution was used in Domingues et al (2019) for tuning the parameters of a PID controller for an Antilock Braking Systems with good results over NSGA-II, but the later was more consistent with the results.

5.1.3.1 Comparison of bioinspired methods for process control

Now let's compare ~~how~~ the performance of some bioinspired methods for industrial process control. In Cespedes et al (2016) Ant Colony Optimization, Invasive Weed Colony Optimization (Mehrabian and Lucas, 2006b), Linear Biogeography-based optimization (Simon, 2008), Genetic Algorithms and Particle swarm optimization are compared when solving the tuning of an industrial PID controller for overdamped second order plus time delay (ODSOPTD) plants. The main results are summarized below. These methods were used to minimize J_{di} , but it is important to note that, for this particular study, the methods were not implemented to produce a Pareto front (i.e. to minimize J_r at the same time). The idea is just to compare different bioinspired methods computationally. However, they can be adapted to be used in multiobjective optimization problems with small changes.

Ant Colony Optimization

Ants can naturally find the shortest path between its nest and the food source by producing some kind of chemical signaling that ~~let~~ the complete population know the path that most ants are using. As pointed by Dorigo et al (2006): "these ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization (ACO) exploits a similar mechanism for solving optimization problems"

Initially a fixed number of "artificial ants" are assigned a given random path which represent the different values of the decision variables that minimize ~~some~~ ~~the~~ cost function. The quantity of "artificial pheromone" is also assigned to each path according to the fitness of the solution. The algorithm then ~~start~~ to discard some paths and in the end, the path with more "pheromone" is supposed to represent the optimal solution (Goss et al, 1989).

out allows to
 cursing?
 starts

Invasive Weed Colony Optimization

Invasive Weed Optimization (IWO) is a search algorithm first presented in Mehrabian and Lucas (2006a). The idea behind the method is based on how weed colonize and distribute the space around (Binitha and Sathya, 2012). IWO can solve multidimensional optimization problems following the next steps:

1. Initialize the population: First a set of random initial solutions widespread over the multidimensional problem space are selected. These solutions are considered as members of the weed colony.
2. Reproduction: Each member of the population is allowed to produce seeds depending on its own, as well as the colony's, lowest and highest fitness, such that, the number of seeds produced by a weed increases linearly from lowest possible seed for a weed with worst fitness to the maximum number of seeds for a plant

- with best fitness (which corresponds to the lowest objective function value for a minimization problem)(Kundu et al, 2011)
3. Spatial dispersal: The seeds are distributed randomly across the search space with a varying variance. The idea behind this is that the seeds will not be near (that is, be similar) to the parent plant.
 4. Competitive exclusion: When all seeds have a defined position, they are ranked using the cost function to minimize. The weeds with lower function value are discarded, since this cost function is considered to represent the fitness of each plant, when discarding the ones with worst value, it is ensure that only the specimens with better fitness survive and are able to reproduce in the next generation.

Linear Biogeography-based optimization *animals*

In its original form, Biogeography is the branch of biology that studies the geographical distribution of plants and ~~animal~~ and the mathematical models associated with the extinction and migration of species (MacArthur and Wilson, 1967).

Biogeography-based optimization (BBO) is an evolutionary algorithm in which each possible solution to the problem is treated as an habitat (or island). The solution with a better cost function then is considered to be a better habitat because the value of the solution is analogous to the characteristics of the habitat that ~~let~~ different species to thrive (Simon, 2008).

Good habitats are considered to have high rate of emigration, because its good features allows an increase in the number of species. This increase may lead to a saturation in its capacity to house more species. The solutions with lower fitness has a high rate of immigration, because animals and plants search for less concentrated habitats to grow and reproduce.

New solutions are found by mixing the characteristics of each habitats according to its emigration and immigration rates. Then, this new habitats are compared against each other, and the best ones are used to form new solutions until an optimal is found.

Genetic Algorithms

The Genetic Algorithms (GA) are based on the ideas of evolution, genetics and natural selection. The main characteristics of GA are (Simon, 2013):

- It tries to simulate the sexual reproduction of a biological population.
- The individuals have a finite life span.
- In each generation, some new characteristics of the population arises due to random mutation.
- There is a positive correlation between the ability to survive and the ability to reproduce.

its

As in nature, the idea behind GA is that only the best fitted specimen in a population are able to reproduce, and therefore, pass ~~it~~ genome to the next generation. This "fitness" value is considered to be the cost function that is intended to be minimized (Mitchell, 1995).

In this case, each individual is viewed as a possible solution to the problem. Its value is coded as a binary chain, simulating the genetic information of living beings. The algorithm then decides which individuals of each generation are allowed to ~~be~~ reproduced and therefore, only the ones with a minimal the minimal cost function survive.

Particle swarm optimization

is

Particle swarm optimization (PSO) is also a search algorithm. But in this case, each solution in the pool of initial solutions ~~are~~ considered to represent an individual in a swarm (for example a flock of birds). In general, each individual is called a particle which represents a possible solution to the problem. The interesting characteristic of this method is that the adjustment of each particle depends on its history and its relationship with the neighboring particles (Shi, 2004).

If ~~for~~ example, a flock of birds is considered as the biological counterpart, the objective is to define an algorithm that mimics the movement of this flock that naturally occurs without an apparent leader, this phenomenon is usually called swarm intelligence (Kennedy and Eberhart, 1995). At each iteration, the acceleration of each particle is changed to move them to the best solutions found.

Comparison of each method for industrial controller tuning

In this comparison a PID controller is tuned as regulator using the methods presented above, for a second order overdamped system with pure time delay. The results of each method were obtained with a computer equipped with an Intel Core i5-3470 CPU at 3.20GHz and 8 GB of RAM using MATLAB as programming language.

For comparison purposes, the result of the optimization using the ~~interior-point~~ to perform (IP) algorithm is presented. In all cases it was required ~~at least~~ 125 iteration ~~where~~ performed in order to let all methods to explore the complete solution space. Also, in al cases, the initial solutions where selected around the same point. A total of 100 different experiments were applied to 9 different plants. The difference in the plants were the time delay and the relationship between the larger and smaller time constant. ~~This~~ plants are presented in Table 5.3. It has to be noticed in these plants that they cover all the spectrum of plants, from ~~plants with small time delay ($L = 0.1$)~~ to plants where the time delay is twice the value of the larger time constant. *Plants*

*these
methods*

It was found that all ~~method~~ were able to find an optimal similar to the one found using the deterministic method IP. When comparing the disturbance rejection response, presented in Fig. 5.6, it was found that the results of each method are acceptable. The values presented on the figure are the IAE of the regulator response,

Table 5.3: Parameter of the test plants.

Plant Parameters $\{K, T, L, a\}$	
P1	{1, 1, 0.1, 0.0}
P2	{1, 1, 0.1, 0.5}
P3	{1, 1, 0.1, 1.0}
P4	{1, 1, 1.0, 0.0}
P5	{1, 1, 1.0, 0.5}
P6	{1, 1, 1.0, 1.0}
P7	{1, 1, 2.0, 0.0}
P8	{1, 1, 2.0, 0.5}
P9	{1, 1, 2.0, 1.0}

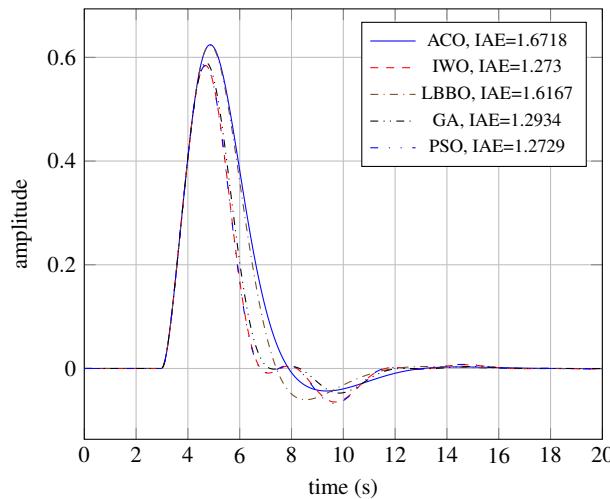


Fig. 5.6: Simulation of the different bio-inspired optimization methods to an unitary step change in the disturbance for plant P5.

applied by taking

that is, the algorithm were ~~used to take~~ J_{di} as its cost function. The lowest value was found with the PSO and IWO methods. The simulation in Fig. 5.6 is the application of the average parameters of the controller with plant P5. Remember that the methods were applied 100 times for each case, the average was taken to minimize the random nature of the bio-inspire methods. The computational cost ~~of~~ each method is presented in Table 5.4. It can be seen that the method with higher number of iteration, in average, is LBBO, followed by PSO. IWO and GA has the lowest number of mean iterations.

Compared to the base case of the IP algorithm, the bio-inspired methods have a ~~bigger~~ function calls, because the bio-inspired methods has a large number of “agents” (particles, ants, genes, habitats, seeds, etc). Regarding the spent time, the

larger number

*associated
to*

Table 5.4: Computational performance for different optimization methods and different plants.

Method	Number of iterations		Function count	Iteration time		
	mean	max		mean	max	average
IP	51	90	284	0.011	0.036	0.007
ACO	135	135	6750	0.118	0.125	0.001
IWO	125	125	6106	0.059	0.127	0.021
LBBO	337	500	7706	0.032	0.052	0.008
GA	125	125	6300	0.073	0.078	0.001
PSO	148	253	2976	0.060	0.072	0.002

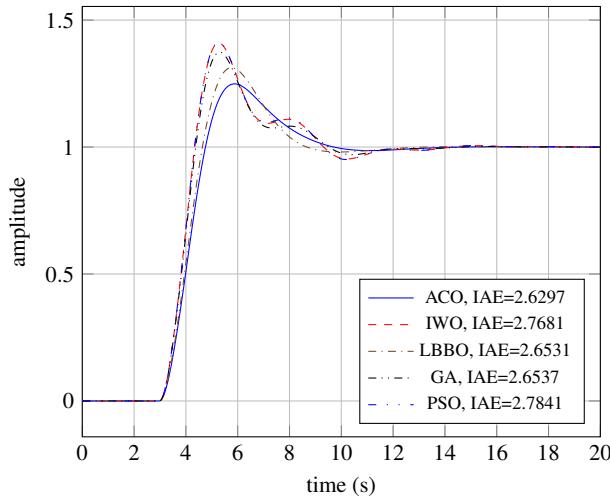


Fig. 5.7: Simulation of the different bio-inspired optimization methods to a unitary step change in reference for plant P5.

algorithm that takes longer time to finish in average was ACO, and the fastest was LBBO.

Of course, the servo response can also be analyzed with the obtained controllers. In Figure 5.7, the response to a step change in the setpoint can be observed for all methods. Since the controller used had only one degree of freedom, the controller that has better response for regulation response has the worst response for setpoint tracking. This compromise exists even when a two degrees of freedom (2DoF) controller is employed. It is true that for most cases, the disturbance rejection is more important ~~than~~ the setpoint tracking, however, in industrial processes, both responses may have to be taken into account for a correct functioning of the system. In those cases, multiobjective optimization comes handy in order to let the decision maker find the best set of parameters for the controlled loop.

than

to

5.1.3.2 Deterministic methods for multiobjective optimization

In its core conception, bioinspired methods are stochastic. They depend on a random set of initial solutions or they add some randomness within the algorithm. For this reason, the final solution obtained may be (hopefully) slightly different each time the optimization is performed.

The classical deterministic methods are based ~~more~~ in the computation of a gradient, in order to minimize the cost function. Given that it exists a lot of results and well-proven algorithms to minimize the problem with this idea, it is logical to try to pose a multiobjective problem in such a way that the standard methods can be applied. One way to do this is performing a scalarization.

The main idea behind this scalarization is to take all the cost functions and formulate the problem in such a way that ~~it~~ minimize a single function gives a result such that all functions are taken into account simultaneously. Repeatedly solving this scalar optimization methods while varying a parameter ~~lead~~ leads to find the Pareto front. According to Marler and J.S. (2004), the main methods are the weighted sum (WS), ~~the~~ normal boundary intersection (NBI) (Das and Dennis, 1998), the normalized normal constraint (NNC) (Messac et al, 2003) and the enhanced normalized normal constraint (ENNC) (Sanchis et al, 2008). These methods are explored in the next section and used in the rest of the book.

5.2 Scalarization algorithms to find the Pareto front

In general, the algorithms to find the optimal value of a function are designed to be used in a single objective paradigm. In order to be able to use the same standard algorithms with a multi-objective problem, some scalarization method has to be employed.

The most important methods are summarized next and then applied in Chapter 6.

5.2.1 Weighted Sum

WS methodology is a popular procedure to transform a MOOP into a single objective problem by creating a new objective function that is the result of the aggregation of all the functions involved with certain weight for each one (Marler and J.S., 2004):

$$F_{WS}(\mathbf{x}) = \sum_{i=1}^k \alpha_i f_i(\mathbf{x}), \quad \text{where } \sum \alpha_i = 1! \quad (5.4)$$

where α_i is the weight ~~of~~ associated with function f_i . The idea behind the utility function F_{WS} is to be able to take into account all individual cost functions at the same time. It is known that when minimizing (5.4), the solution belongs to the Pareto

front. Therefore, it is of great importance to select the values of the weights that better reflect the desire of the decision-maker.

The weights have two different roles that are entangled, in one hand the weights can be used to represent the importance of one function over the others (the bigger the weight, the higher the importance) and in the other hand the weights ~~can~~ be used to equalize the relative values of the functions (one function may yield higher values than shadows the others).

However, choosing the values of the weight can be difficult. In Marler and Arora (2010) it is shown that the weight can be interpreted as a first order approximation of a preference function, and therefore, cannot fully take into account the desires of the decision-maker.

Lets take a two function MOOP as an example. If the Pareto front wants to be computed, one may try ~~to~~ first normalized ~~the~~ function:

$$F_{WS}(\mathbf{x}) = \alpha_{1WS}\hat{f}_1(\mathbf{x}) + \alpha_{2WS}\hat{f}_2(\mathbf{x}), \quad (5.5)$$

with $\alpha_{1WS} + \alpha_{2WS} = 1$, and $\hat{f}_1(\mathbf{x})$ and $\hat{f}_2(\mathbf{x})$ the normalized versions of $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, respectively. One possible normalization (see Marler and J.S. (2004)) is given by:

$$\hat{f}_1(\mathbf{x}) = \frac{f_1(\mathbf{x}) - \min(f_1(\mathbf{x}))}{\max(f_1(\mathbf{x})) - \min(f_1(\mathbf{x}))}. \quad (5.6)$$

With this normalization, the utopia point is moved to the origin and the maximum value of the new normalized function is 1.

The optimization problem then is written as:

$$\begin{aligned} & \min_{\mathbf{x}} F_{WS}(\mathbf{x}), \\ & \text{s.t. } h(\mathbf{x}) = 0, \quad \text{is to be solved by} \\ & \quad g(\mathbf{x}) \leq 0, \end{aligned} \quad (5.7)$$

where $h(\mathbf{x})$ and $g(\mathbf{x})$ are the equality and inequality constraints of the original problem. To find the Pareto front, the problem in (5.7) ~~is solve~~ varying the weights. However, it is known that the WS method is not appropriate to find the Pareto front. In first place, when (5.5) is minimized for different values of α_{1WS} and α_{2WS} in order to obtain the Pareto front, an even distribution of the weights does not assure an even distribution of the points in the front. Also, with WS it is not possible to obtain Pareto points in the non-convex region of the front, and therefore, not all the possible solutions can be found (Das and Dennis, 1997). In order to tackle this issue, alternative problem formulation ~~s~~have been proposed in the literature to obtain the Pareto front which are presented next.

5.2.2 Normal Boundary Intersection

The NBI is a variation in the way that the MOOP is posed as a single objective optimization problem, in order to obtain an even spaced Pareto front (Das and Dennis, 1998). In figure 5.8, a representation of the method is shown for two normalized objective functions. If the utopia plane (the plane that contains the anchor points, in the case of a bi-objective problem, the straight line that joins the anchor points) is parameterized by $\Phi\beta$, where $\Phi(:, i) = \mathbf{F}(\mathbf{x}_i^*) - \mathbf{F}(\mathbf{x}^*)$, $\mathbf{F}(\mathbf{x}_i^*)$ is the value of the multi-objective function evaluated in the i th anchor point, $\mathbf{F}(\mathbf{x}^*)$ is the value of the function at the utopia point, and β is chosen as:

$$\beta = \begin{bmatrix} \alpha_{1NBI} \\ \alpha_{2NBI} \end{bmatrix}, \quad (5.8)$$

with $\alpha_{1NBI} + \alpha_{2NBI} = 1$.

The central idea behind the NBI method is to find the maximum distance from the utopia plane towards the utopia point (with direction $\hat{\mathbf{n}}$) that is normal (or quasi normal as proposed in Das and Dennis (1998)) to the utopia plane. In other words, this method finds the border of the feasible region that is closer to the utopia point (farther from the utopia plane). This problem is considered a subproblem, because with one given β , only one point of the Pareto front is found but, by varying this parameter β ~~evenly~~^{evenly}, it is possible to obtain an even spaced realization of the front.

The problem then is posed as follows:

evenly

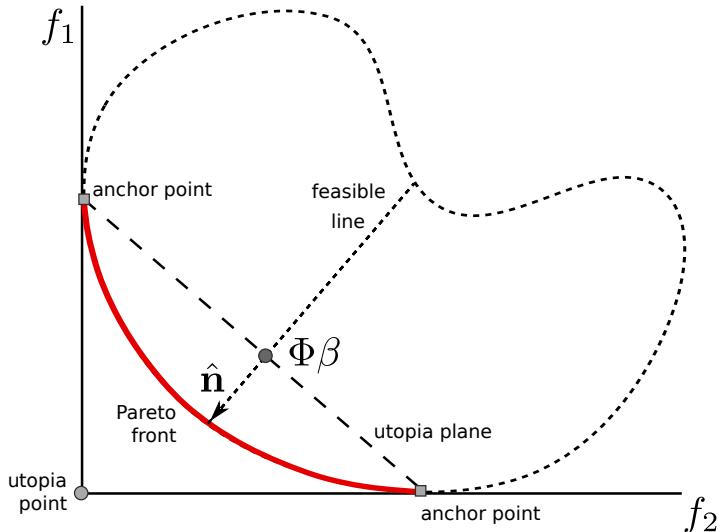


Fig. 5.8: NBI optimization method.

$$\begin{aligned}
 & \max_{\mathbf{x}, v} v, \\
 \text{s.t. } & \Phi \boldsymbol{\beta} + v \hat{\mathbf{n}} = \mathbf{F}(\mathbf{x}), \\
 & h(\mathbf{x}) = 0, \\
 & g(\mathbf{x}) \leq 0.
 \end{aligned} \tag{5.9}$$

The NBI method converts the original problem by adding an equality constraint. By maximizing the new variable v (which represents the distance from the utopia plane towards the utopia point), the front that is closer to the utopia point is found. An alternative formulation of (5.9) was proposed in Shukla (2007) to ensure that only the points that really belong to the Pareto front are found. *space*

This method has been widely used in several areas. For example, in Stehr et al (2003) was used to analyze the compromise between gain and phase margin when designing analog circuits. In Sendín et al (2004), the NBI was considered in the design of nonlinear bioprocesses. In Ierapetritou and Jia (2007) the NBI is used to optimize the scheduling of a chemical process with uncertainty. In Vahidinasab and Jadid (2010) NBI is applied to develop optimal bidding strategies for the participants of oligopolistic energy markets; the social welfare and the emissions are considered as ~~the~~ cost functions and the constraints take into account the characteristics of the generators and the power flow of the system. In Ganesan et al (2013), NBI is used in conjunction with a meta-heuristic algorithm to generate optimal solution options to the green sand mould system problem. In Brito et al (2014) the method is used coupled with mean-squared error functions in a robust parameter design of the surface roughness in end milling process. In Rubio-Largo et al (2014) the method is adapted to solve a traffic grooming problem in the telecommunication field. In Rojas et al (2015) a comparison between several scalarization methods, including NBI, was presented for a first order plus time delay (FOPTD) plant where different disturbance sources are considered. In Naves et al (2017) NBI is used for the optimization of methyl orange treatment with ozone. In Simab et al (2018) a model for short-term hydrothermal scheduling problem in the presence of the pumped-storage technology and stated as Mixed-Integer Non-Linear Programming, while the scalarization was done using NBI. Finally, in Moura et al (2018) NBI was used in the construction of a Pareto boundary chart of a treatment of a synthetic solution of amoxicillin in a reactor with ozone bubbling.

5.2.3 Normalized Normal Constraint

The NNC is presented in Messac et al (2003) and is intended to improve the results of the NBI by formulating the optimization problem only with inequality constraints and by filtering all the non-Pareto optimal points. The main idea of the methodology is presented in figure 5.9: the utopia plane is parameterized in a similar way as the NBI but, instead of constraining the points to be within a line, the new constrained feasible region is constructed with the original feasible region and a line that is nor-

mal to the utopia plane. With this new feasible region it is only required to minimize one of the functions (e.g. f_1) in order to find the Pareto front.

By varying the parameter $\bar{\mathbf{X}}_{pj}$ along the utopia plane, it is possible to find an even spaced front. $\bar{\mathbf{X}}_{pj}$ is computed as

$$\bar{\mathbf{X}}_{pj} = \alpha_{1NNC}\hat{\mathbf{F}}(\mathbf{x}_1^*) + \alpha_{2NNC}\hat{\mathbf{F}}(\mathbf{x}_2^*). \quad (5.10)$$

with $\alpha_{1NNC} + \alpha_{2NNC} = 1$ and where $\hat{\mathbf{F}}(\mathbf{x}_1^*)$ is the first anchor point and $\hat{\mathbf{F}}(\mathbf{x}_2^*)$ is the second. The methodology can be extended to higher dimensions.

The optimization problem can be written as follows:

$$\begin{aligned} & \min_{\mathbf{x}} \hat{f}_1(\mathbf{x}), \\ \text{s.t. } & \bar{\mathbf{N}}_1^T (\hat{\mathbf{F}}(\mathbf{x}) - \bar{\mathbf{X}}_{pj}) \leq 0, \\ & h(\mathbf{x}) = 0, \\ & g(\mathbf{x}) \leq 0, \end{aligned} \quad (5.11)$$

where $\bar{\mathbf{N}}_1$ is the vector that contains the direction of the utopia plane. In some cases, the optimization may yield points that do not belong to the Pareto front. In Messac et al (2003), the authors propose to use a filter algorithm to eliminate those points.

This method has been used in several cases. Also in Hosseini et al (2016) the method was implemented to optimally solve the transmission congestion management taking into account the cost of congestion management, voltage stability margin, and transient stability margin. In Sánchez et al (2017), the NNC was considered

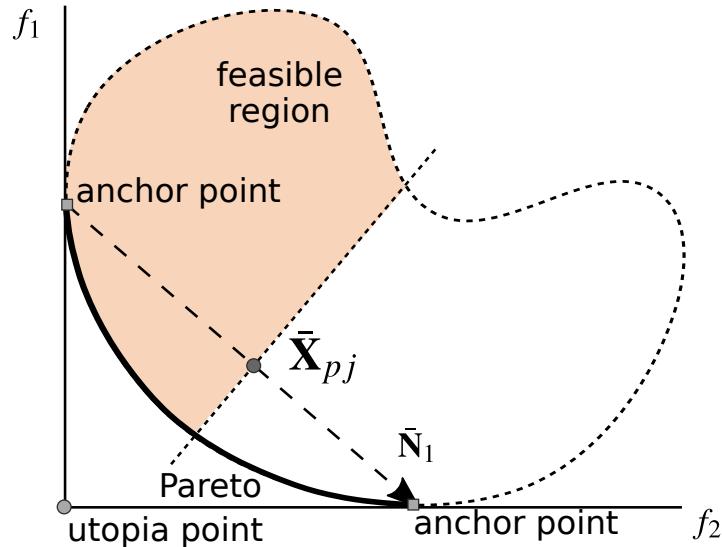


Fig. 5.9: NNC optimization method.

to find optimally balanced tuning rules for fractional-order proportional-integral-derivative controllers for FOPTD process models subject to a robustness constraint. In Mittal and Mitra (2017) the NNC was used in conjunction with an evolutionary algorithm to find the optimum number and location ~~of~~ wind turbines in a wind farm. In Benki et al (2018), the NNC was implemented ~~in~~ the design of an aerosol can, taking into account both the dome growth and the dome reversal pressure ~~loss~~. In Tan et al (2018), the NNC was applied ~~in~~ the design of microvascular panels for battery cooling applications. The NNC is also applied in Liu et al (2019) within their algorithm to optimally control the glycerol in a 1,3-propanediol batch process.

5.2.4 Enhanced Normalized Normal Constraint

The ENNC Sanchis et al (2008), is a new perspective of the original NNC method. Implicitly, the NNC method supposes that in each anchor point, the other functions that are not optimal, have their worst value. For a two functions optimization, this is always the case; however for more than two functions this supposition is not true in general. The ENNC method redefines the anchor points in such a way that the supposition of the NNC holds true, and then the same method may be used. Other advantage of the ENNC is that it is possible to expand the explored regions of the problem, given a better representation of the Pareto front. The new anchor points (called pseudo anchor points) are defined as:

$$\text{is } F_i^{**} = [f_1^N \ f_2^N \ \cdots \ f_i^* \ \cdots \ f_n^N], \quad (5.12)$$

where f_i^N is the value of function f_i at the pseudo nadir point. The effect of this new definition is to enlarge the utopia hyper plane and scaling the functions in such a way that the Pareto front is evenly obtained while the unexplored regions of the Pareto are reduced.

The Pareto is then computed using the same methodology as in the NNC case. This method has also been applied in several cases, for example in Contreras-Leiva et al (2016), the ENNC is applied in the optimization of the tuning parameters of a 2DoF PID controller for an ODSOPTD plant. In Pereira et al (2017) an augmented version of the ENNC is applied to optimize the milling process of aluminum alloy Al 7075 taking into account the axial cutting force, the energy consumption and the material removal rate. The ENNC is tested in the optimization of a multiobjective model based predictive controller and compared with different techniques in Toro et al (2011) and for the nonlinear case in Vallerio et al (2014).

5.3 Solution selection from the Pareto front

The optimizations techniques presented in Section 5.2 are very well suited to find the Pareto Front of a multiobjective optimization problem. However, in the end, it is necessary to decide which of the multiple equally optimal points is the one that is going to be selected as the final solution.

Consider the general Pareto Front presented in Figure 5.10. Once the Pareto is computed, it is certainly very useful to plot it to see how the cost functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ vary with the change in the decision variables \mathbf{x} . In this figure, the arbitrary solution p_1 is pointed out. By definition, all points in the Pareto Front are equally optimal, therefore, what makes p_1 any special from other points? More important, how any^{one} of the Pareto points can be selected from any other?

One may then conclude that obtaining the Pareto front is half the solution to the MOOP. The final task to fully solve the problem is to be able to select one of the (maybe infinite) possible points of the Pareto. With a front like the one presented in Figure 5.10, it may be easy to explore all possible solutions, but with more than three objective functions, it is impossible to directly plot the front. ~~and~~^{even} even with three objectives, the visualization of the results may be cumbersome.

Even more, the plot of Figure 5.10 contains the solutions viewed from the function space, but what is really necessary is to know the value of the decision variables. But each^{one} of every point in the Pareto is associated with a n -size vector representing one of the optimal solutions. Once again, plotting the Pareto is not enough to help the decision maker to solve the problem.

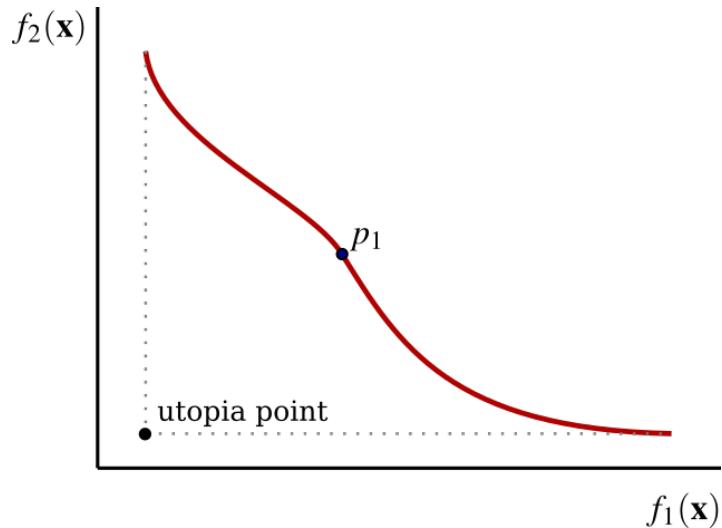


Fig. 5.10: General Pareto Front with one solution selected.

For these reasons, once the Pareto is found, it may be useful to accomplish some tasks to help the decision maker:

- Visualize the Pareto to understand the relation between the cost functions and the decision variables.
- Use the Pareto as data for the construction of a decision tool.
- Use the Pareto as part of a decision tool.

In the following, these three tasks are going to be explored.

5.3.1 Visualization of the Pareto Front

One of the advantages of using a multiobjective problem approach, is its ability to take into account many cost functions at the same time, and being able to find the set of the *best* solutions. This is the ultimate goal of a multiobjective optimisation, to provide useful information to the decision makers in order to help to choose according to its preferences. This can be done in terms of a large set of raw data that has lately to be processed accordingly. This introduces some cognitive issues that become harder and a clear obstacle for decision making stage, especially when having many objectives (say more than 3). Although there are some performance metrics that can evaluate the quality of a Pareto front approximation set (Inverted Generational Distance (Bosman and Thierens, 2003) and Hypervolume (Zitzler and Thiele, 1999) among others), ~~it is still not intuitive~~ this approach really helps the decision maker to understand the trade-off relationship among objectives.

In contrast, high-dimensional data visualisation is a widely recognised effective way to facilitate the analysis and understanding of multidimensional data. ~~M~~ore effective and intuitive to facilitate the decision maker stage to understand the trade-off solutions and thus make a meaningful decision. In the context of multi-objective optimisation, comparing to quantitative performance metrics, visualisation is, in principle, able to provide a decision maker better insights about Pareto front approximation sets (e.g. the distribution of solutions, the geometric characteristics of Pareto front approximation) thus to facilitate the decision-making (e.g. the exploration of trade-off relationship, the knee region or region of interest).

According to Gao et al (2019) a high quality visualisation tool must provide three types of information in the high dimensional space:

- Should provide accurate shape, location, and range of the approximate Pareto front.
- From the provided visualisation, decision makers can observe trade-off between objectives, monitor the evolution progress, assess the quality of the approximate front, and select their preferred solutions if desired.
- Should be scalable to any dimensions, handle a large number of individuals on the approximate front, and simultaneously visualise multiple fronts for the purpose of visual comparison. Moreover, the resulted visualisation plot should be robust and insensitive to the addition or removal of an individual.

To tackle these issues, different kinds of plots have been proposed in the literature that tries to present all the information of the Pareto in a bi-dimensional graph. Generally speaking, they can be divided into three categories. Interested readers are encouraged to found a recent taxonomy from Filipič and Tušar (2018).

- A. Visualisation of All Objective Information: These visualisation techniques aim to reveal all individual objective information of the underlying approximation set. Specifically, scatter plot is the most commonly utilised data visualisation technique that provides a holistic exploration of the population distribution.
- B. Visualisation via Dimension Reduction: Techniques in this category aims to transform the high-dimensional data into a lower dimensional space to facilitate the human cognition. In modern data analytics, there are many dimension reduction techniques available to implement such transformation. Instead of using dimension reduction techniques from machine learning, Blasco et al (2008), proposed a new visualisation technique called level diagrams to visualise the approximation set in a objective-wise manner. More specifically, each diagram is a two-dimensional scatter plot where the horizontal coordinate represents the objective value at the corresponding objective while the vertical coordinate indicates the distance with respect to the ideal point. As claimed by the authors, the level diagrams are able to facilitate the investigation of some of the Pareto front characteristics, i.e. discontinuities, closeness to ideal point and ranges of attainable values.
- C. Visualisation via a Transformed Coordinate System: Techniques in this category share some similarity with the dimension reduction. They also aim to visualise the original high-dimensional data in a lower-dimensional space. But they try to maintain the original information as much as possible. For example, Ibrahim et al (2016) developed a variant of the classic Radial coordinate visualisation (RadViz) (Hoffman et al, 2002), called 3D-RadViz, by adding an additional dimension. In particular, this additional dimension represents the perpendicular distance between a solution and a hyper-plane formed by the extreme points of the underlying population. By doing so, the 3D-RadViz is able to provide the information of the convergence of the population.

As usual ~~when~~ ^{where} there do exist different approaches to deal with a complex problem, ~~there~~ ^{an} no single visualisation technique is able to provide a comprehensive understanding of the characteristics of the approximation set. It is important to know the specific characteristics, pros and cons of each approach and evaluate the suitability for the problem at hand.

5.3.2 The Pareto as a decision tool

Since the Pareto front cannot be considered as the final solution of the MOOP, one may use it as the basis to construct a tool that takes advantage of the multiple optimal points found.

In other words, the Pareto front becomes the raw material that is used to produce the final decision tool. This tool may be an algorithm (as a tuning rule for PID controllers for example) that incorporates the information obtained from the optimization to produce a final decision.

For example in Zhao (2007) the concept of Pareto optimality is used to define a genetic programming approach for optimal decision trees. The author presented a Java code for the final tool and use code in two different study cases.

Another example can be found in Das et al (2012). In this paper, the route for the transportation of hazardous waste has to be selected taking into account the cost of the route and the mortality and morbidity of incidents. Once the Pareto frontier has been obtained, the authors present a selection criteria where the Cost Elasticity of risk and the Knees on the Pareto are taken into account for the final decision.

References

- Benki A, Habbal A, Mathis G (2018) A metamodel-based multicriteria shape optimization process for an aerosol can. *Alexandria Engineering Journal* 57(3):1905–1915, DOI 10.1016/J.AEJ.2017.03.036, URL <https://www.sciencedirect.com/science/article/pii/S1110016817301291>
- Binitha S, Sathya SS (2012) A Survey of Bio inspired Optimization Algorithms. *International Journal of Soft Computing and Engineerin* 2(2):137 –151, URL <http://www.ijscce.org/wp-content/uploads/papers/v2i2/B0523032212.pdf>
- Blasco X, Herrero J, Sanchis J, Martínez M (2008) A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences* 178(20):3908–3924, DOI 10.1016/J.IINS.2008.06.010, URL <https://www.sciencedirect.com/science/article/pii/S0020025508002016>
- Bosman PAN, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 7(2):174–188, DOI 10.1109/TEVC.2003.810761
- Brito T, Paiva A, Ferreira J, Gomes J, Balestrassi P (2014) A normal boundary intersection approach to multiresponse robust optimization of the surface roughness in end milling process with combined arrays. *Precision Engineering* 38(3):628–638, DOI 10.1016/j.precisioneng.2014.02.013, URL <https://www.sciencedirect.com/science/article/pii/S0141635914000439>
- <https://linkinghub.elsevier.com/retrieve/pii/S0141635914000439>
- Cespedes M, Contreras M, Cordero J, Montoya G, Valverde K, Rojas JD (2016) A comparison of bio-inspired optimization methodologies applied to the tuning of industrial controllers. In: 2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI), IEEE, pp 1–6, DOI 10.1109/CONCAPAN.2016.7942340, URL <http://ieeexplore.ieee.org/document/7942340/>
- Contreras-Leiva MP, Rivas F, Rojas JD, Arrieta O, Vilanova R, Barbu M (2016) Multi-objective optimal tuning of two degrees of freedom PID controllers using the ENNC method. In: 20th International Conference on System Theory, Control and Computing, Sinaia, Romania
- Das A, Mazumder T, Gupta A (2012) Pareto frontier analyses based decision making tool for transportation of hazardous waste. *Journal of Hazardous Materials* 227-228:341–352, DOI 10.1016/j.jhazmat.2012.05.068, URL <https://www.sciencedirect.com/science/article/pii/S0304389412005717>
- <https://linkinghub.elsevier.com/retrieve/pii/S0304389412005717>

- Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization* 14(1):63–69, DOI 10.1007/BF01197559
- Das I, Dennis JE (1998) Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization* 8(3):631–657, DOI 10.1137/S1052623496307510
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197, DOI 10.1109/4235.996017, URL <http://ieeexplore.ieee.org/document/996017/>
- Domingues PH, Freire RZ, Coelho LDS, Ayala HV (2019) Bio-Inspired Multiobjective Tuning of PID-Controlled Antilock Braking Systems. 2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings pp 888–895, DOI 10.1109/CEC.2019.8790023
- Dorigo M, Blum C (2005) Ant colony optimization theory: A survey. *Theoretical Computer Science* 344(2-3):243–278, DOI 10.1016/j.tcs.2005.05.020, URL <https://linkinghub.elsevier.com/retrieve/pii/S0304397505003798>
- Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Computational Intelligence Magazine* 1(4):28–39, DOI 10.1109/MCI.2006.329691, URL <http://ieeexplore.ieee.org/document/4129846/>
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, vol 0-7803-267, pp 39–43, DOI 10.1109/MHS.1995.494215, URL <http://ieeexplore.ieee.org/document/494215/>
- Filipič B, Tušar T (2018) A taxonomy of methods for visualizing pareto front approximations. In: Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO '18, ACM Press, New York, New York, USA, pp 649–656, DOI 10.1145/3205455.3205607, URL <http://dl.acm.org/citation.cfm?doid=3205455.3205607>
- Ganesan T, Vasant P, Elamvazuthi I (2013) Normal-boundary intersection based parametric multi-objective optimization of green sand mould system. *Journal of Manufacturing Systems* 32(1):197–205, DOI 10.1016/j.jmsy.2012.10.004, URL <https://www.sciencedirect.com/science/article/pii/S0278612512000933> <https://linkinghub.elsevier.com/retrieve/pii/S0278612512000933>
- Gao H, Nie H, Li K (2019) Visualisation of Pareto Front Approximation: A Short Survey and Empirical Comparisons. In: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 1, pp 1750–1757, DOI 10.1109/CEC.2019.8790298, URL <https://ieeexplore.ieee.org/document/8790298/>
- Goss S, Aron S, Deneubourg JL, Pasteels JM (1989) Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* 76(12):579–581, DOI 10.1007/BF00462870, URL <http://link.springer.com/10.1007/BF00462870>
- Han Y, Brdys M, Piotrowski R (2008) Nonlinear PI control for dissolved oxygen tracking at wastewater treatment plant. *IFAC Proceedings Volumes* 41(2):13,587–13,592, DOI 10.3182/20080706-5-KR-1001.02301, URL <http://dx.doi.org/10.3182/20080706-5-KR-1001.02301> <https://linkinghub.elsevier.com/retrieve/pii/S1474667016411675>
- Henze M, Harremoës P, Arvin E, Jansen JIC (1997) Wastewater treatment, biological and chemical process, 2nd edn. Springer Verlag, New York, USA
- Hoffman PE, Grinstein GG, Marx K, Grosse I, Stanley E (2002) A survey of visualizations for high-dimensional data mining. In: Information visualization in data mining and knowledge discovery, pp vol. 104, p. 4
- Hosseini SA, Amjadi N, Shafie-khah M, Catalão JP (2016) A new multi-objective solution approach to solve transmission congestion management problem of energy markets. *Applied Energy* 165:462–471, DOI 10.1016/j.apenergy.2015.12.101, URL <https://www.sciencedirect.com/science/article/pii/S0306261915016748> <https://linkinghub.elsevier.com/retrieve/pii/S0306261915016748>
- Ibrahim A, Rahnamayan S, Martin MV, Deb K (2016) 3D-RadVis: Visualization of Pareto front in many-objective optimization. In: 2016 IEEE Congress on Evolution-

- ary Computation (CEC), IEEE, pp 736–745, DOI 10.1109/CEC.2016.7743865, URL <http://ieeexplore.ieee.org/document/7743865/>
- Ierapetritou MG, Jia Z (2007) Short-term scheduling of chemical process including uncertainty. *Control Engineering Practice* 15(10 SPEC. ISS.):1207–1221, DOI 10.1016/j.conengprac.2006.10.009, URL <http://www.sciencedirect.com/science/article/pii/S0967066106001808>
- Jeppsson U (1996) Modelling Aspects of Wastewater Treatment Processes. PhD thesis, Lund Institute of Technology (LTH)
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, Perth, WA, Australia, vol 4, pp 1942–1948, DOI 10.1109/ICNN.1995.488968, URL <http://ieeexplore.ieee.org/document/488968/>
- Kundu D, Suresh K, Ghosh S, Das S, Panigrahi B, Das S (2011) Multi-objective optimization with artificial weed colonies. *Information Sciences* 181(12):2441–2454, DOI 10.1016/j.ins.2010.09.026, URL <https://linkinghub.elsevier.com/retrieve/pii/S0020025510004809>
- Liu C, Gong Z, Joseph Lee HW, Teo KL (2019) Robust bi-objective optimal control of 1,3-propanediol microbial batch production process. *Journal of Process Control* 78:170–182, DOI 10.1016/J.JPROCONT.2018.10.001, URL <https://www.sciencedirect.com/science/article/pii/S09595152418303706>
- Longo S, D'Antoni BM, Bongards M, Chaparro A, Cronrath A, Fatone F, Lema JM, Mauricio-Iglesias M, Soares A, Hospido A (2016) Monitoring and diagnosis of energy consumption in wastewater treatment plants. A state of the art and proposals for improvement. *Applied Energy* 179:1251–1268, DOI 10.1016/j.apenergy.2016.07.043, URL <http://dx.doi.org/10.1016/j.apenergy.2016.07.043>
- MacArthur R, Wilson E (1967) The theory of Island Biogeography, 1st edn. Princeton University Press, Princeton New Jersey
- Mahdavian M, Wattanapongsakorn N (2014) Multi-objective optimization of PID controller tuning for greenhouse lighting control system considering RTP in the smart grid. 2014 International Computer Science and Engineering Conference, ICSEC 2014 pp 57–61, DOI 10.1109/ICSEC.2014.6978129
- Marler RT, Arora JS (2010) The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization* 41(6):853–862, DOI 10.1007/s00158-009-0460-7, URL <http://link.springer.com/10.1007/s00158-009-0460-7>
- Marler RT, JS A (2004) Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 26(6):369–395, DOI 10.1007/s00158-003-0368-6
- Mehrabian A, Lucas C (2006a) A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics* 1(4):355–366, DOI 10.1016/j.ecoinf.2006.07.003, URL <https://linkinghub.elsevier.com/retrieve/pii/S1574954106000665>
- Mehrabian AR, Lucas C (2006b) A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics* 1(4):355–366, DOI 10.1016/j.ecoinf.2006.07.003
- Messac A, Ismail-Yahaya A, Mattson C (2003) The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization* 25(2):86–98, DOI 10.1007/s00158-002-0276-1, URL <http://link.springer.com/10.1007/s00158-002-0276-1>
- Mitchell M (1995) Genetic algorithms: An overview. *Complexity* 1(1):31–39, DOI 10.1002/cplx.6130010108, URL <http://doi.wiley.com/10.1002/cplx.6130010108>
- Mittal P, Mitra K (2017) Decomposition based multi-objective optimization to simultaneously determine the number and the optimum locations of wind turbines in a wind farm. *IFAC-PapersOnLine* 50(1):159–164, DOI 10.1016/J.IFACOL.2017.08.027, URL <https://www.sciencedirect.com/science/article/pii/S2405896317300393>
- Moura D, Barcelos V, Samanamud GRL, França AB, Lofrano R, Loures CCA, Naves LLR, Amaral MS, Naves FL (2018) Normal boundary intersection applied as multivariate and multiobjective optimization in the treatment of amoxicillin synthetic solution. *Environmental Monitoring and Assessment* 190(3):140, DOI 10.1007/s10661-018-6523-8, URL <http://link.springer.com/10.1007/s10661-018-6523-8>

- Naves FL, de Paula TI, Balestrassi PP, Moreira Braga WL, Sawhney RS, de Paiva AP (2017) Multivariate Normal Boundary Intersection based on rotated factor scores: A multiobjective optimization method for methyl orange treatment. *Journal of Cleaner Production* 143:413–439, DOI 10.1016/J.JCLEPRO.2016.12.092, URL <https://www.sciencedirect.com/science/article/pii/S0959652616321564>
- Nejjari F, Dahhou B, Benhammou A, Roux G (1999) Non-linear multivariable adaptive control of an activated sludge wastewater treatment process. *International Journal of Adaptive Control and Signal Processing* 13(5):347–365, DOI 10.1002/(SICI)1099-1115(199908)13:5;347::AID-ACSS543_3.0.CO;2-8, URL <http://doi.wiley.com/10.1002/%28SICI%291099-1115%28199908%2913%3A5%3C347%3A%3AAID-ACSS543%3E3.0.CO%3B2-8>
- Olsson G, Newell B (1999) *Wastewater Treatment Systems. Modelling, Diagnosis and Control*, 1st edn. IWA Publishing, London, UK
- Pereira RBD, Leite RR, Alvim AC, de Paiva AP, Ferreira JR, Davim JP (2017) Multi-objective robust optimization of the sustainable helical milling process of the aluminum alloy Al 7075 using the augmented-enhanced normalized normal constraint method. *Journal of Cleaner Production* 152:474–496, DOI 10.1016/j.jclepro.2017.03.121, URL <https://www.sciencedirect.com/science/article/pii/S0959652617305656> <https://linkinghub.elsevier.com/retrieve/pii/S0959652617305656>
- Pierezan J, Ayala HH, Da Cruz LF, Freire RZ, Dos S Coelho L (2014) Improved multiobjective particle swarm optimization for designing PID controllers applied to robotic manipulator. *IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CICA 2014: 2014 IEEE Symposium on Computational Intelligence in Control and Automation, Proceedings* pp 1–8, DOI 10.1109/CICA.2014.7013255
- Reynoso-Meza G, Garcia-Nieto S, Sanchis J, Blasco FX (2013) Controller Tuning by Means of Multi-Objective Optimization Algorithms: A Global Tuning Framework. *Control Systems Technology, IEEE Transactions on* 21(2):445–458, DOI 10.1109/TCST.2012.2185698
- Rojas JD, Valverde-Mendez D, Alfaro VM, Arrieta O, Vilanova R (2015) Comparison of multi-objective optimization methods for PI controllers tuning. In: *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, IEEE, pp 1–8, DOI 10.1109/ETFA.2015.7301410, URL <http://ieeexplore.ieee.org/document/7301410/>
- Rubinstein RY, Kroese DP (2004) *The Cross-Entropy Method*, 1st edn. Information Science and Statistics, Springer New York, New York, NY, DOI 10.1007/978-1-4757-4321-0, URL <http://link.springer.com/10.1007/978-1-4757-4321-0>
- Rubio-Largo Á, Zhang Q, Vega-Rodríguez MA (2014) A multiobjective evolutionary algorithm based on decomposition with normal boundary intersection for traffic grooming in optical networks. *Information Sciences* 289(1):91–116, DOI 10.1016/j.ins.2014.08.004, URL <https://www.sciencedirect.com/science/article/pii/S0020025514007932> <https://linkinghub.elsevier.com/retrieve/pii/S0020025514007932>
- Sánchez HS, Padula F, Visioli A, Vilanova R (2017) Tuning rules for robust FOPID controllers based on multi-objective optimization with FOPDT models. *ISA Transactions* 66:344–361, DOI 10.1016/j.isatra.2016.09.021, URL <https://linkinghub.elsevier.com/retrieve/pii/S0019057816303652>
- Sanchis J, Martínez M, Blasco X, Salcedo JV (2008) A new perspective on multi-objective optimization by enhanced normalized normal constraint method. *Structural and Multidisciplinary Optimization* 36(5):537–546, DOI 10.1007/s00158-007-0185-4, URL <http://dx.doi.org/10.1007/s00158-007-0185-4>
- Sendín OH, Otero I, Alonso AA, Banga JR (2004) Multi-objective optimization for the design of bio-processes. In: *Computer Aided Chemical Engineering*, vol 18, Elsevier, pp 283–288, DOI 10.1016/S1570-7946(04)80113-5, URL <https://www.sciencedirect.com/science/article/pii/S1570794604801135> <https://linkinghub.elsevier.com/retrieve/pii/S1570794604801135>
- Shi Y (2004) Particle swarm optimization. *IEEE Connections* 2(1):8–13

- Shukla PK (2007) On the Normal Boundary Intersection Method for Generation of Efficient Front. Computational Science - ICCS 2007 4487:310–317, DOI 10.1007/978-3-540-72584-8_40, URL http://dx.doi.org/10.1007/978-3-540-72584-8_40
- Simab M, Javadi MS, Nezhad AE (2018) Multi-objective programming of pumped-hydro-thermal scheduling problem using normal boundary intersection and VIKOR. Energy 143:854–866, DOI 10.1016/j.energy.2017.09.144, URL <https://linkinghub.elsevier.com/retrieve/pii/S0360544217316651>
- Simon D (2008) Biogeography-Based Optimization. IEEE Transactions on Evolutionary Computation 12(6):702–713, DOI 10.1109/TEVC.2008.919004, URL <http://ieeexplore.ieee.org/document/4475427/>
- Simon D (2013) Evolutionary Optimization Algorithms, 1st edn. John Wiley and Sons
- Stehr G, Graeb H, Antreich K (2003) Performance trade-off analysis of analog circuits by normal-boundary intersection. In: Proceedings of the 40th conference on Design automation - DAC '03, ACM Press, New York, New York, USA, p 958, DOI 10.1109/DAC.2003.1219159, URL <http://portal.acm.org/citation.cfm?doid=775832.776073>
- Tan MYH, Najafi AR, Pety SJ, White SR, Geubelle PH (2018) Multi-objective design of microvascular panels for battery cooling applications. Applied Thermal Engineering DOI 10.1016/j.applthermaleng.2018.02.028
- Tian Y, Wang Q, Wang Y, Jin Q (2014) A novel design method of multi-objective robust PID controller for industrial process. Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, ICIEA 2014 pp 242–246, DOI 10.1109/ICIEA.2014.6931166
- Toro R, Ocampo-Martínez C, Logist F, Impe JV, Puig V (2011) Tuning of Predictive Controllers for Drinking Water Networked Systems. IFAC Proceedings Volumes 44(1):14,507–14,512, DOI 10.3182/20110828-6-IT-1002.00415, URL <https://www.sciencedirect.com/science/article/pii/S1474667016459598>
- Vahidinasab V, Jadid S (2010) Normal boundary intersection method for suppliers' strategic bidding in electricity markets: An environmental/economic approach. Energy Conversion and Management 51(6):1111–1119, DOI 10.1016/j.enconman.2009.12.019
- Vallerio M, Van Impe J, Logist F (2014) Tuning of NMPC controllers via multi-objective optimisation. Computers and Chemical Engineering 61:38–50, DOI 10.1016/j.compchemeng.2013.10.003, URL <https://www.sciencedirect.com/science/article/pii/S0098135413003256>
- Zhao H (2007) A multi-objective genetic programming approach to developing Pareto optimal decision trees. Decision Support Systems 43(3):809–826, DOI 10.1016/j.dss.2006.12.011, URL <https://www.sciencedirect.com/science/article/pii/S016792360600217X>
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1(1):32–49, DOI 10.1016/j.swevo.2011.03.001
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 3(4):257–271, DOI 10.1109/4235.797969

Chapter 6

Application of the multiobjective approach

of

Abstract In this chapter, the multiobjective optimization (MOO) techniques are tested and applied to different scenarios. A comparison ~~to~~ the implementation of several linearization techniques is presented. Then, a well known benchmark problem is tackled with the presented methodology and compared with different tuning methods. Finally, a LiTaO₃ Thin Film Deposition Process is used as a prototype for a large dead-time process and tested using a two cost function arrangement.

6.1 Comparison of the methods to obtain the Pareto front

6.1.1 Performance comparison of the scalarization methods

In order to compare ~~the~~ efficiency of different linearization methods, different test were performed on the normalized process given by:

$$\hat{P}(\hat{s}) = \frac{e^{-\tau_0 \hat{s}}}{\hat{s} + 1}, \quad (6.1)$$

with values of τ_0 from 0.1 to 2. In order to show the results, the simulations presented in this section only contains the case for $\tau_0 = 0.5$. The other values of τ_0 give similar results.

For this particular case, the controller is supposed to be represented by the transfer function $C(s, \boldsymbol{\theta})$ given by two degrees of freedom (2DoF) proportional integral (PI) controller:

$$u(s) = C_r(s, \boldsymbol{\theta})r(s) - C_y(s, \boldsymbol{\theta})y(s), \quad (6.2)$$

where $C_r(s, \boldsymbol{\theta})$ is the reference controller given by the transfer function:

$$C_r(s, \boldsymbol{\theta}) = K_p \left(\beta + \frac{1}{T_{is}} \right), \quad (6.3)$$

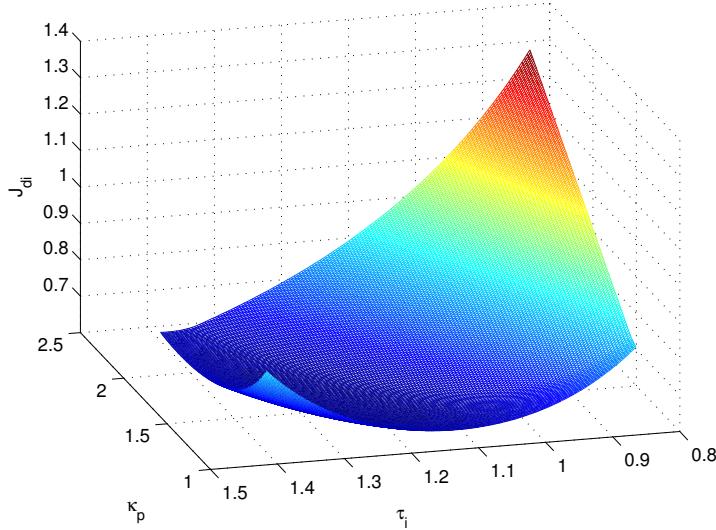


Fig. 6.1: $J_{di}(\hat{\theta})$ function for a value of $\tau_0 = 0.5$.

and $C_y(s, \theta)$ is the feedback controller given by the transfer function:

$$C_y(s, \boldsymbol{\theta}) = K_p \left(1 + \frac{1}{T_i s} \right), \quad (6.4)$$

The parameters K_p , T_i and β are as usual, the proportional gain, the integral time and the set-point weight, respectively, which can be grouped as a single vector variable denoted by $\boldsymbol{\theta} = [K_p, T_i, \beta]^T$.

In Figure 6.1, the cost function $J_{di}(\hat{\theta})$ is plotted as a function of κ_p and τ_i , which represent the normalized parameters of the PI controller:

$$\begin{aligned} \kappa_p &\doteq K K_p, \\ \tau_i &\doteq \frac{T_i}{T}. \end{aligned} \quad (6.5)$$

From Figure 6.1, it can be concluded that the cost function J_{di} is rather convex, and very flat close to its minimal value. The graph in Fig. 6.1 was plotted with ten thousand controller tunings of the controller taking the optimal tuning of $J_{di}(\theta)$ and $J_{do}(\theta)$ as central points. The computation time to obtain this set of data was in the range of hours. In Fig. 6.2, the complete set of points is plotted in the objective functions plane with the corresponding Pareto front highlighted. The points that represent the ~~front~~ represents only 3% of all the points plotted in that figure. If the decision maker is only interested in the This fact shows the necessity to use some scalarization methods like normal boundary intersection (NBI) or normalized normal constraint (NNC) in order to obtain only the front, without the need to compute points that will be dismissed later in the process.

Pareto
front

fronte incompleta

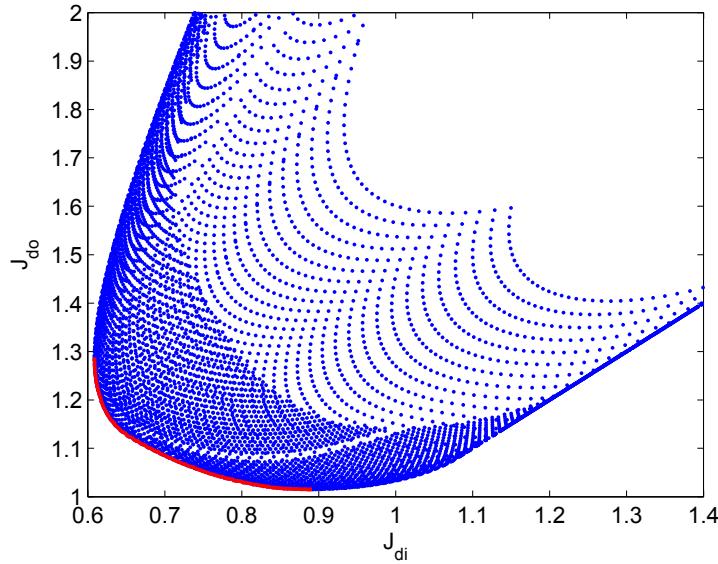


Fig. 6.2: Pareto front obtained directly from a ten thousand points data set for (6.1) with $\tau_0 = 0.5$.

In Fig. 6.3, the result using weighted sum (WS) ~~are~~ is presented. The solid line represents the real Pareto front and the points marked with a plus sign correspond to the obtained values. As it can be seen, all the points obtained with the WS method are Pareto optimal, however its distribution is not evenly spaced and are grouped for low values of $J_{di}(\theta)$. This is expected since ~~the~~ the shape of the Pareto front does not correspond to the relation needed to obtain ~~a~~ even spaced front with an even spaced parametrization of α_1 and α_2 as presented in Das and Dennis (1997).

The result for NBI method is presented in Fig. 6.4. As it can be seen, the frontier obtained with this method is evenly spaced. It is important to note that the method finds different Pareto optimal points than the WS method

Finally, when the Pareto front is found using the NNC method, the points computed are very similar to the ones found with the NBI scalarization. The NNC case is presented in Fig. 6.5. Comparing Figure 6.3 with Figures 6.4 and 6.5, it is clear that both the NBI and NNC both are able to find a more accurate approximation of the Pareto front than WS. Of course, all the methods are able to find Pareto optimal points, but in order to have a good understanding of the problem, it is important to have a set of point that are representative of the actual behavior of the front. It also is important to note that, if the actual Pareto front is convex, the results with the NBI and NNC should be the same. In case of non-convexity, it is possible that the two methods yield to different points in the non-dominated points that should be filtered, following the NNC method Messac et al (2003).

In Fig. 6.6 the comparison between the results for κ_p are presented while the values for τ_i are shown in Fig. 6.7. As it can be seen, NBI and NNC obtain the exact

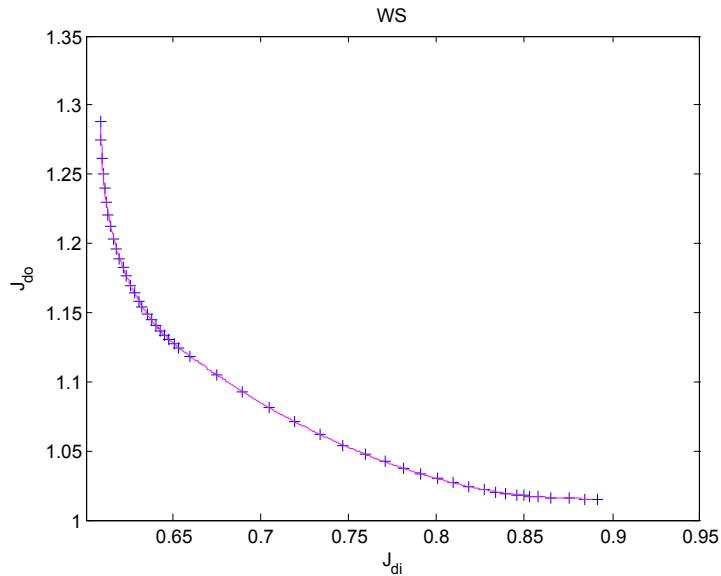


Fig. 6.3: Pareto front obtained with WS method and $\tau_0 = 0.5$.

same points in the Pareto for all values of α . However, there are certain differences in the results for τ_i . Since NBI depends on a equality constraint, it is more difficult

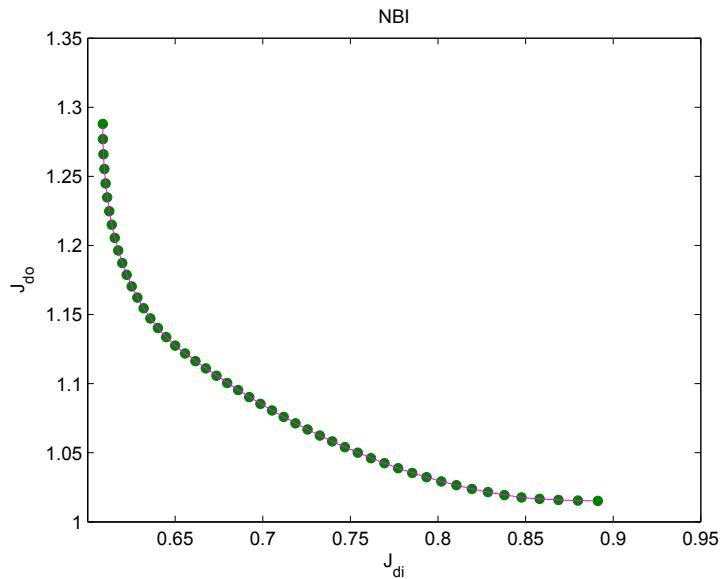


Fig. 6.4: Pareto front obtained with NBI method and $\tau_0 = 0.5$.

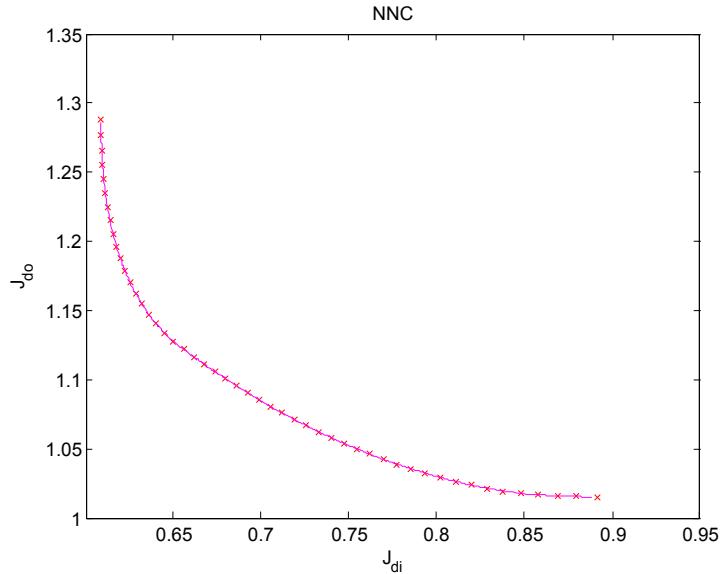


Fig. 6.5: Pareto front obtained with NNC method and $\tau_0 = 0.5$.

for the optimization algorithm¹ to converge to the solution. In fact, for some cases,

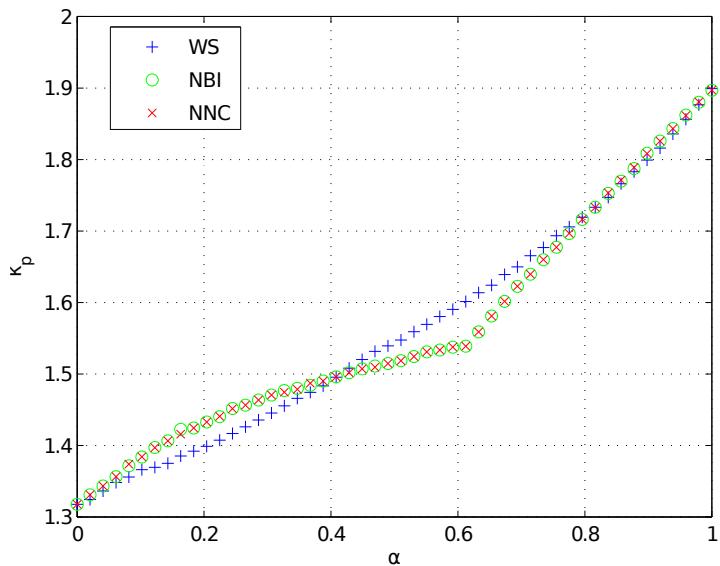


Fig. 6.6: κ_p values for all the methods wrt α and with $\tau_0 = 0.5$.

¹ For all the methods, the optimization problem was solved using an active-set strategy with a maximum of 1000 iterations and 1000 function evaluations.

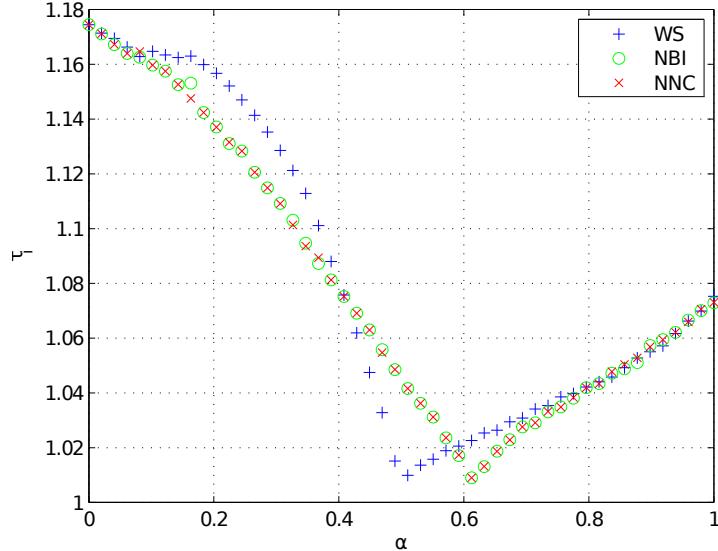


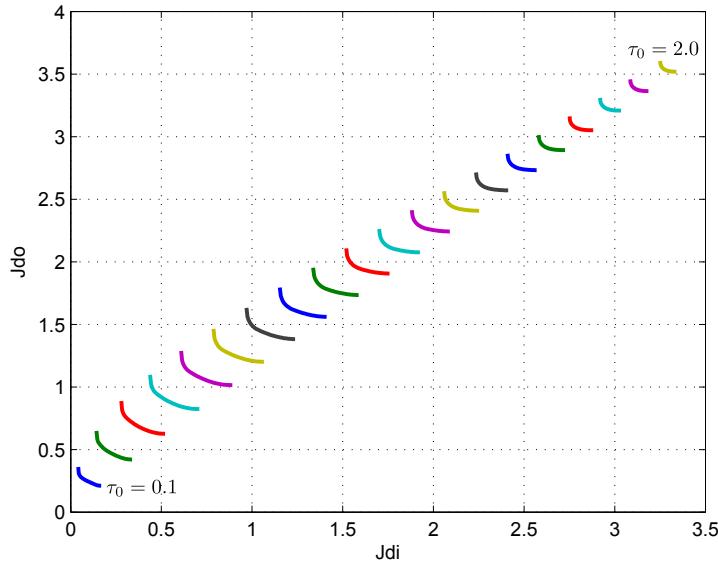
Fig. 6.7: τ_i values for all the methods wrt α and with $\tau_0 = 0.5$.

Table 6.1: Performance comparison for different optimization methods

Method	Iterations		Function evaluations		Computation time (s)	
	Average	Max	Average	Max	Average	Max
WS	68.82	109	132.34	212	39.11	63.77
NBI	30.78	225	142.86	1003	44.88	320.81
NNC	41.28	297	147.02	1002	71.21	486.47

using the NBI methodology *leads* to more than a thousand function evaluations. Since the maximum was set to one thousand evaluations, for some point the results do not exactly match. However, it is interesting to note that for all three cases, the variation in the values of κ_p and τ_i follows certain pattern.

The computational performance of the methods has also been considered. Using MATLAB in a PC running Linux with a 3.2.0.2-amd64 kernel and an Intel Core i7 at 1.60GHz, the results are given in Table 6.1. Each Pareto front corresponds to 50 points for $\tau_0 = 0.5$ and the table gives the values obtained regarding computational performance: number of iterations, number of function evaluations and the time spent during the process. Interestingly, it was found that NBI has better performance than NNC for this particular case. However, during the computation it was noticed that the NBI reached the maximum number of evaluations in four of the fifty points whereas the NNC method reach the same limit for just one point. If the results of the WS is taken as the base reference, it was found that NBI and NNC required a lower number of iterations on average, however, they needed more function evalua-

Fig. 6.8: Fifty point Pareto fronts varying τ_0 .

tion (8% and 11%, respectively) and more computation time (14.74% and 82.05%, respectively).

Regarding the computation of all points of the Pareto, it was found that using the WS method 32.6 minutes where needed, using the NBI 37.4 minutes where spend while using the NNC method took 59.33 minutes. However, it is clear that performing the computation of the Pareto front is a computational expensive task, that may no be suitable for an “online” tuning procedure. However, it is interesting to consider the case where the front is computed “offline” for a family of plants (using the normalized version) and then, use the data directly for the decision making process of choosing the final tuning of the controller.

In the following, the results are analyzed from a control theory point of view and compared with other methodologies.

6.1.2 Analysis of the results from the control theory perspective

When the plant is varied from $\tau_0 = 0.1$ to $\tau_0 = 2$ in steps of 0.1, the corresponding Pareto fronts are plotted in Fig. 6.8. It was expected that for increasing values τ_0 , the values of J_{di} and J_{do} also increase because of the inherent delay of the plant which directly affects the integral of the absolute value of the error (IAE), *the achievable performance*

As it can be seen, depending *of* τ_0 , the values that the cost functions have can be very different. For this reason it may be interesting to compare the Paretos with a normalized version of the cost functions for the same variation on τ_0 . This study is

on

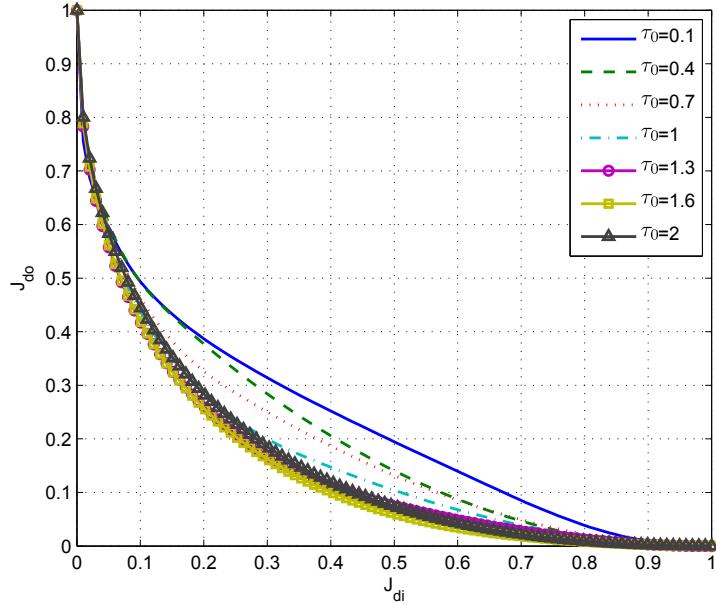


Fig. 6.9: Normalised version of the Pareto front for different values of τ_0 .

presented in Figure 6.9. In this figure, all the Paretos have been scaled in order to have its cost functions between 0 and 1. The value 0 represents the lowest value of the cost function while the 1 represents the highest value. Therefore, it can be seen as a degradation value: a value of one represents a total degradation of one of the function (but, preserving the fact that it is Pareto optimal).

It is interesting to note that, if a degradation of around 10% in the input disturbance cost function is allowed, it yields to an improvement of up to 50% in the servo response for all values of τ_0 . However, it is interesting to note that, in the other case where the system is initially set to be optimal for servo response (a value of $J_{do} = 0$) if a 10% of degradation is allowed, an improvement between 30% and 60% is achieved depending of the value of τ_0 . These results show that, by analyzing the Pareto front, an important improvement can be achieved if one of the functions is allowed to be degraded by just a small amount. But of course, the decision on how much to degrade is entirely up to the decision maker. But it is clear that the Pareto is a good tool to support that decision.

Is it possible to identify a single point in the Pareto front that gives the best compromise between these objectives? Although how much degradation is allowed is a subjective decision of the designer, it may be possible to give an alternative. The point that may be a good start is the point in the Pareto front that is closer to the utopia point. This point can be obtained by minimizing for example:

$$\min_{\hat{\theta}} \sqrt{(\hat{J}_{di}(\hat{\theta}))^2 + (\hat{J}_{do}(\hat{\theta}))^2}, \quad (6.6)$$

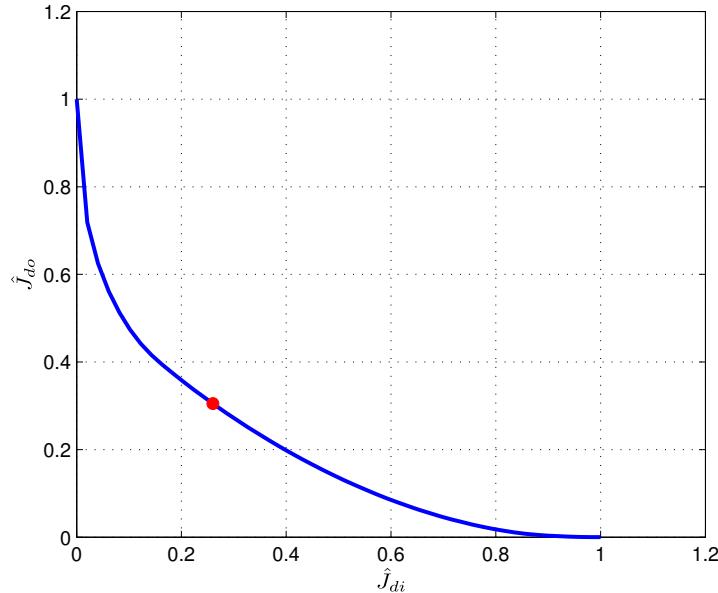


Fig. 6.10: Best compromise in the Pareto front as given by the Cartesian distance.

which represent the Cartesian distance between the Pareto front and the Utopia point

The result for $\tau_0 = 0.5$ is presented in Fig. 6.10. It is necessary to use the normalized cost functions in order to give the same importance to both objective functions. As it can be seen, the tuning of the parameters that are closer to the utopia point is the one that produces a degradation of 26% in $\hat{J}_{di}(\hat{\theta})$ and a corresponding 30.52% degradation in $\hat{J}_{do}(\hat{\theta})$. This point may not be suitable for the requirements of the problem at hand, but is a good ~~start~~ point from where the designer can choose the “best” (not necessarily the optimal) solution for the particular problem.

In Figures 6.11 and 6.12 the variation of the parameters κ_p and τ_i with respect to α for different values of τ_0 are presented. For example, in the case of κ_p higher values of τ_0 gives lower values for this parameter. The variation is small with respect to the increase in the degradation of J_{di} (which is represented by α in the figures), except for lower values of τ_0 . Something similar happens when τ_i is analyzed, however, in this case the value of τ_i increases when τ_0 is larger, but again the variation with respect to the degradation is not very large. This is important to note, because, if the controller does not allow a precise input of the parameter tuning, it may be possible that the ~~desire~~ point in the Pareto front cannot be achieved for some values of τ_0 , since small changes in the tuning produce important changes in the degradation (in percentage). However, it has to be noted in Figure 6.8 that the possible change in IAE for both input and output disturbances rejection is smaller for higher values of τ_0 than for lower values. The relationship between κ_p and τ_i is presented in Figure 6.13. The possible range of values is also smaller for higher values of τ_0 .

starting

desired

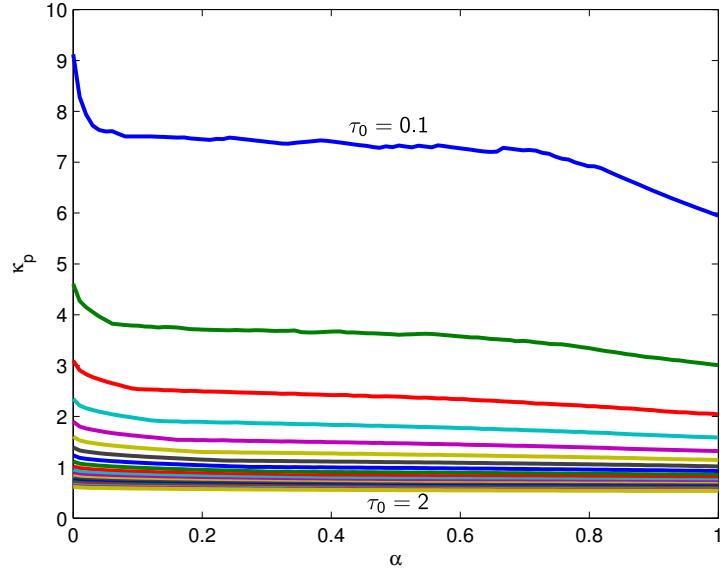


Fig. 6.11: Variation of κ_p wrt the degradation of the J_{di} cost function (α) and $\tau_0 = 0.5$.

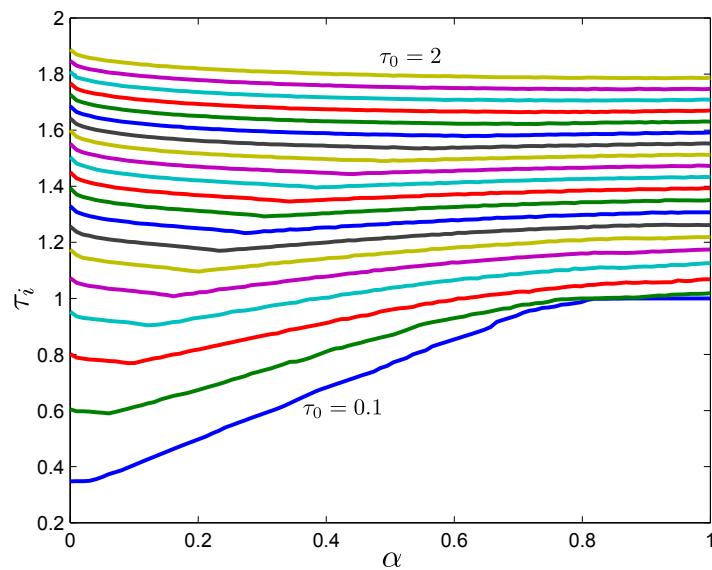
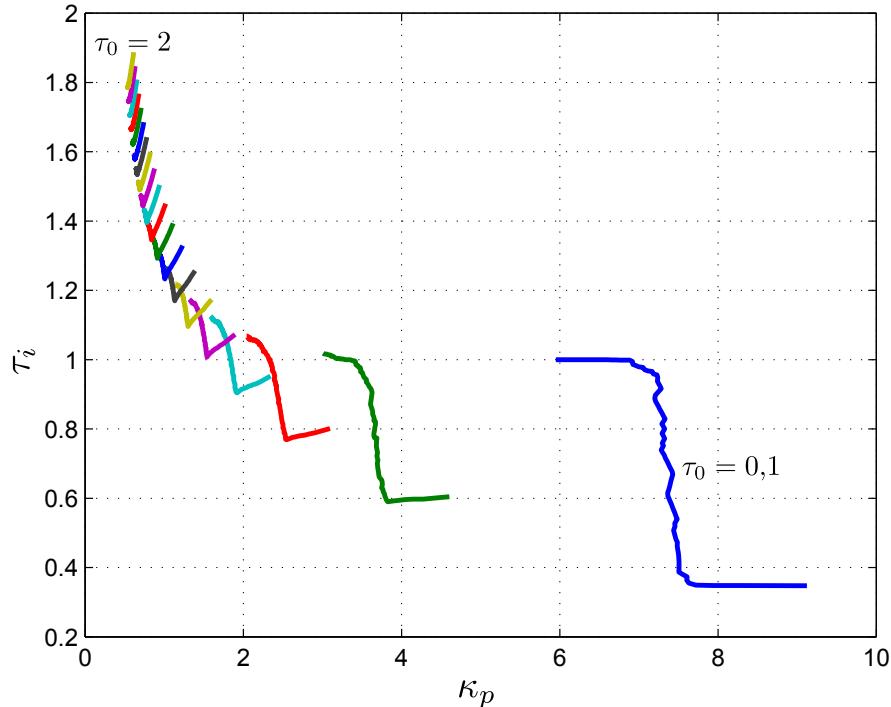


Fig. 6.12: Variation of τ_i wrt the degradation of the J_{di} cost function (α) and $\tau_0 = 0.5$.

Fig. 6.13: τ_i vs κ_p for different values of τ_0 .

It is important to remember that each point of the Pareto front represent a different tuning of a PI controller. Therefore it is possible to compare the relationship between more classical ways to tune the controller with the front itself. The controller was tuned using the rules in O'Dwyer (2000); Åström and Hägglund (1995); Murril (1967); Rovira et al (1969); Grimholt and Skogestad (2012); Smith and Corripio (1985); Ziegler and Nichols (1942). Each tuning also can be represented by a point along with the Pareto. This plot is shown in Figure 6.14. It is important to note that ~~this~~ methods are not necessarily the result of an optimization problem and therefore, they may not ~~be~~ in the Pareto. *belong to*

Mosc

The tuning of Murril (1967) was created with the intention of minimize the IAE for input disturbances. This explains the fact that the value of J_{di} for this controller is very close to the anchor point where J_{di} is minimal. In the case of Rovira (Rovira et al, 1969), it was designed to minimize the servo response of the closed loop controller. In this particular case, given that the controller used in the Pareto is a one degree of freedom PI.

In this particular case, minimizing the servo response is exactly the same as minimizing the output disturbance rejection (J_{do}) for a two degrees of freedom controller. This is the reason why its tuning point is located at the right hand side of the Pareto front. It was interesting to found out that the tuning in Åström and Hägglund (1995)

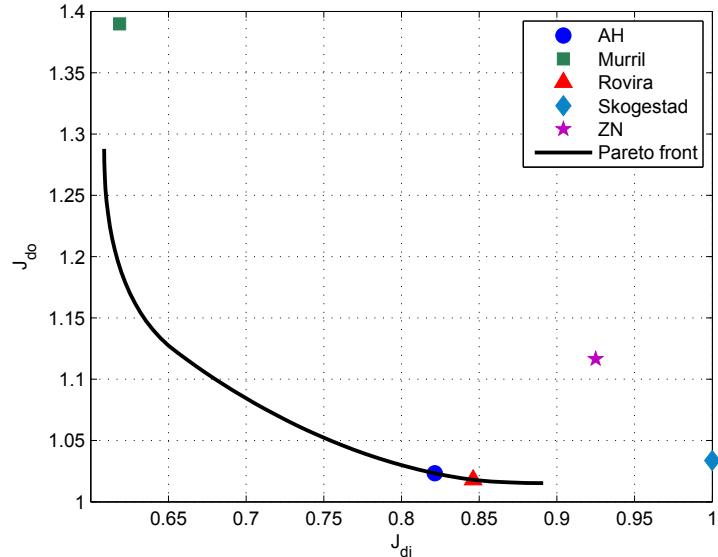


Fig. 6.14: Comparison of several tuning methods within the Pareto front for $\tau_0 = 0.5$.

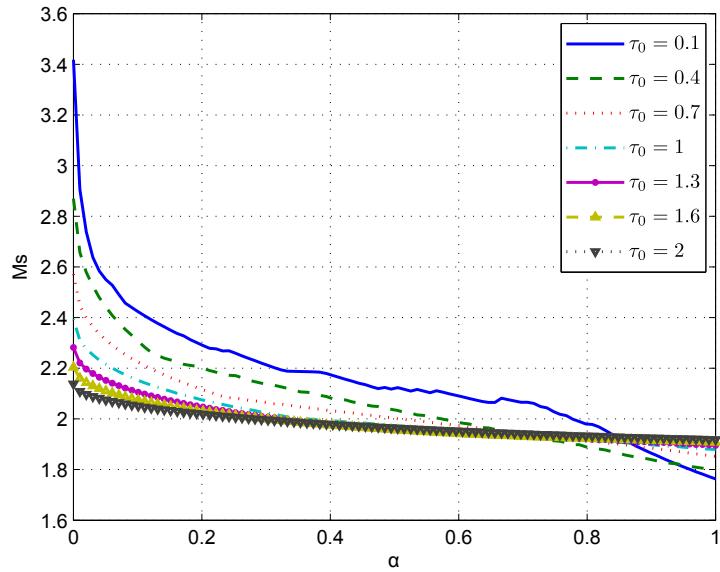
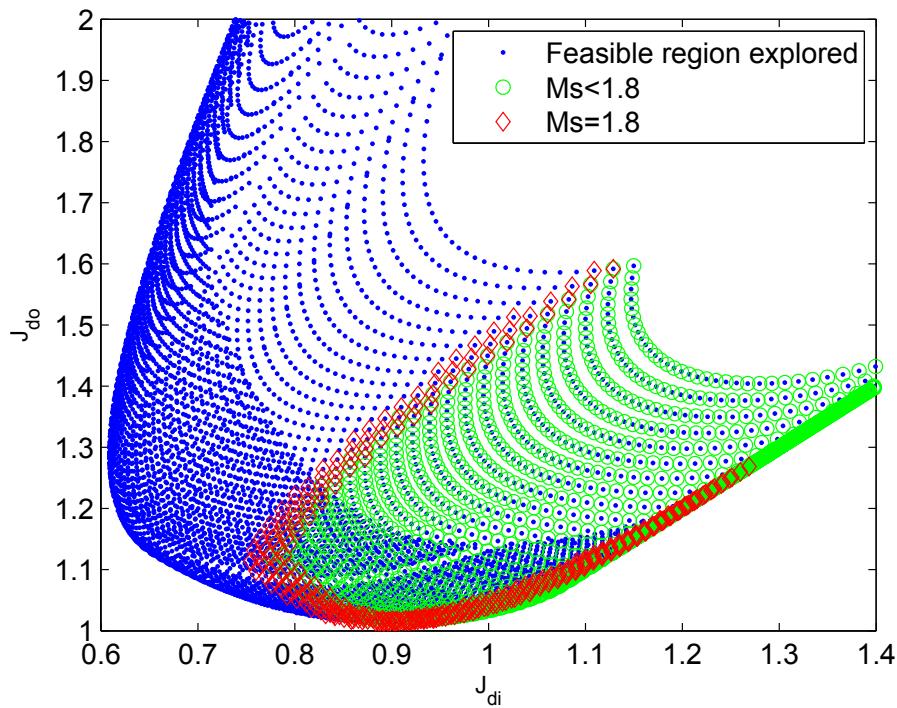
(AH in the figure), yield in the Pareto Front, and therefore can be considered also as Pareto optimal. The case of Grimholt and Skogestad (2012) is also interesting because it is almost optimal in the J_{d0} sense, but due to its consideration of robustness, it is far from the Pareto front given in this particular case. As expected, the method by Ziegler and Nichols (Ziegler and Nichols, 1942) (ZN in the figure) is far from optimal, but it is included in this analysis for comparison purposes only.

In this particular section, the obtained Paretos did not consider the robustness of the controlled system. If the M_s is computed as a function of the degradation of J_{di} , the result is shown in 6.15.

shrink

on

Of course, the robustness of the controlled system is not good for a real application of the controller. It is generally accepted that a value of two or lower is a desirable, and in this case, almost all of them has a value of M_s greater than 2. However, it is possible to consider the robustness inside the optimization problem, either as a cost function or as constraint. As an example, consider the case of $\tau_0 = 0.5$ presented in Figure 6.16. When the case for $M_s \leq 1.6$ is considered, the feasible region is shrink considerably. In this case, the Pareto front for the optimal tuning is outside the new feasible region and therefore, a completely different Pareto front would be obtained when the optimization methods are run with this new constraint. Depending on the desired value of M_s and the given value of τ_0 , it may be possible that the optimal frontier also satisfy the robustness constraint. Therefore, since the robustness can be considered just as a constraint in the optimization problem, a tuning methodology that considers both multi-objective optimality and robustness can be obtained using the Pareto front framework and will be the choice in the rest of this book.

Fig. 6.15: Sensitivity function wrt α , varying τ_0 .Fig. 6.16: Complete feasible region for $\tau_0 = 0.5$ and the subregion where $M_S \leq 1.8$.

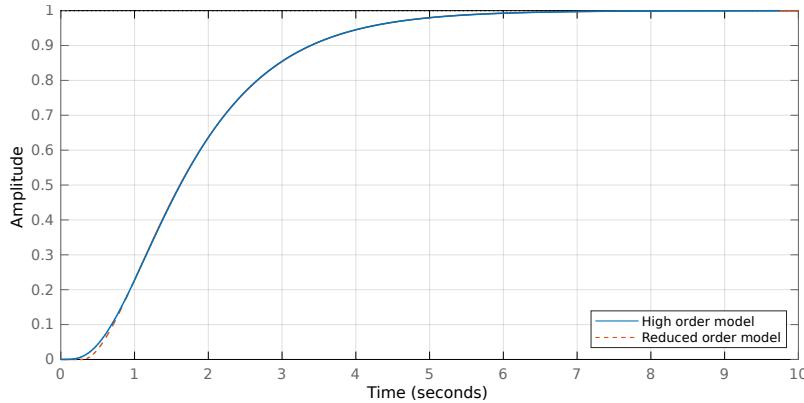


Fig. 6.17: Comparison between the high and reduced order models.

6.2 High-Order Benchmark Plant

First the enhanced normalized normal constraint (ENNC) method is going to be tested in high order benchmark plant (Åström and Hägglund, 2000). The model of the plant is given by a fourth order transfer function:

$$P(s) = \frac{1}{\prod_{n=0}^{n=3} (0.5^n s + 1)}. \quad (6.7)$$

The first step is to obtain a low order model that is able to reflect the main dynamics of the plant. In general, the tuning of proportional integral derivative (PID) controllers starts with a first or second order model (Alfaro, 2006). In this particular case, using a step change as the input signal, the low order model that can be found from this experiment is given by:

$$F(s) = \frac{e^{-0.297s}}{(0.9477s + 1)(0.6346s + 1)}, \quad (6.8)$$

alternatively, if it is supposed that the “real” model of the plant is known, an order reduction procedure, for example, the half-rule method may be used (Skogestad, 2003). The comparison between the high order model and the reduced order model in the time domain is presented in Figure 6.17. As it can be seen, the model represent S accurately the dynamics of the original plant and therefore it is considered to be a good approximation of the original model.

The next step is to find the Pareto front for this particular plant. The followed methodology was as presented in Chapter 7. For this particular case, only J_{di} and J_r , where considered as the cost functions with a 2DoF PID controller. In Figure 6.18, the obtained Pareto front is presented.

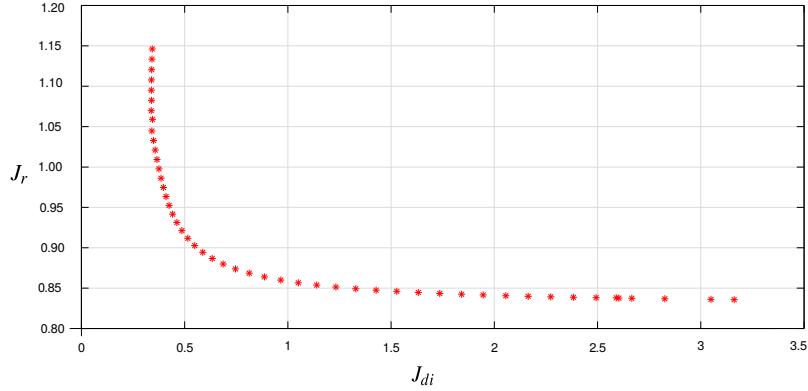


Fig. 6.18: The Pareto front for the benchmark process.

Table 6.2: PID controller parameters using two degrees of freedom.

Tuning	K_c	T_i	T_d	β
optimum J_{di}	3.3750	1.0812	0.3095	0.5466
optimum J_r	3.0572	8.4419	0.3986	1.2329
ART_2	3.3657	1.7636	0.4884	0.2971
$uSORT_2$	3.1708	0.8997	0.3945	0.4731

The curve has a typical form, with a ~~high~~ slope for low values of J_{di} and an almost flat slope for higher values. This shape has a particular physical meaning: to improve the response of the J_{di} cost function, the J_r value has to be augmented (worsening the servo response), however the degradation is not as much as the improvement in the J_{di} function. This is a clear example of one of the many advantages of using a multi-objective framework for controller tuning and the main reason why is the chosen framework in this book, it gives the decision taker more tools to select the more appropriate tuning for the controllers.

In order to compare the response of the optimal controllers, the tuning for the anchor points are presented along the responses of the ART_2 method (Vilanova et al, 2011) and the $uSORT_2$ method(Alfaro and Vilanova, 2012a), in order to compare the performance of the closed loop response. It is important to clarify that these two tunings ~~are~~ just the extreme points of the Pareto front, thanks to the ENNC method and that there is a practically and infinite amount of possible parameter tuning to select. The obtained parameters are listed in Table 6.2 for reference. It is important to note that in all cases, the Maximum Sensibility was set to be around $M_s = 2.0$ to ensure a minimum level of robustness.

In Figure 6.19, the closed-loop responses of all the four controllers are presented for the case of a step change in the setpoint. it was found that precisely the controller in the anchor point of the Pareto front that give the minimum value of J_r is in fact the one that give the best result of all the controllers. However, it has to be noticed

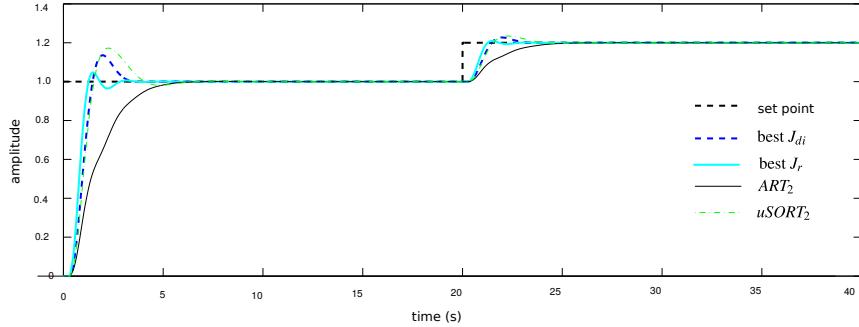
Fig. 6.19: Optimal response of the control system J_r

Table 6.3: Servo response for the benchmark system.

Tuning	IAE	M_s
optimum J_r	1.004	2
optimum J_{di}	1.297	2
$uSORT_2$	1.522	2
ART_2	2.121	2

that both ART_2 and $uSORT$ methods are intended for regulator response mainly, and therefore, it was not expected to have a *low* J_r . The obtained values are given in Table 6.3 where both the IAE and M_s are presented.

On the other hand, the optimal controllers in the regulator mode are presented in Figure 6.20. Again, as expected, the controller in the anchor point that has the lowest value of J_{di} , is the one with the fastest response. Also, it is clear that the other anchor point (the one with the lowest value of J_r) has the worst response for disturbance rejection as it was expected. In table 6.4, the corresponding values of IAE for the curves in Figure 6.20 are presented.

The optimal controller for disturbance rejection is presented in figure 6.20. Again, the tuning given by the ENNC method is the fastest response to reach the desired value. As it can be seen, the ART_2 and the $uSORT_2$ methods fall between these two optimal responses. However, it does not necessarily means that these methods are optimal because they could be dominated by other controllers that are exactly in the front. Only the tuning found with the ENNC method can be considered to be Pareto optimal using the IAE as the metric. In Table 6.4, the IAE values of the responses presented in Figure 6.20 are stated, and again the controller that achieves the lower error value is the one that correspond to the anchor point.

It is important to note that in these figures, only two possible points (in fact, the two extreme cases) were considered, but in reality, there are much more options to select for intermediate values of the parameters between these two cases. It has to be noticed that the second order overdamped model is well suited to approximate high order models. Therefore, having done the computation for this particular model as

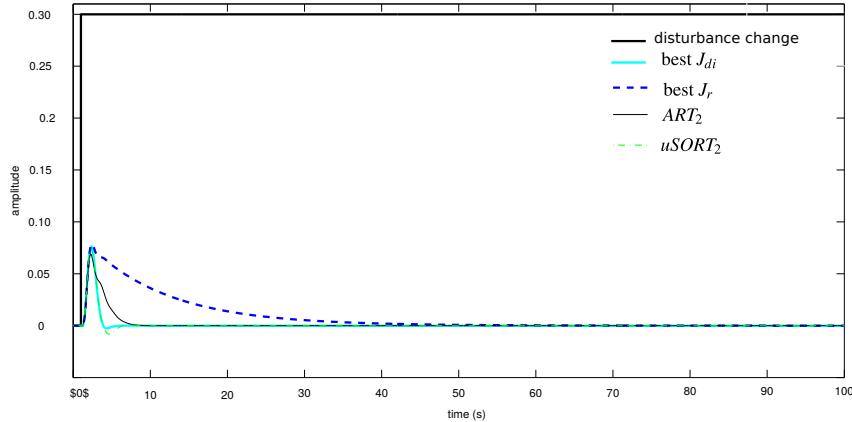
Fig. 6.20: Optimal response of the control system J_{di}

Table 6.4: Regulator response for the benchmark system.

Tuning	IAE	M_s
optimum J_{di}	0.1017	2
$uSORT_2$	0.1095	2
ART_2	0.1574	2
optimum J_r	0.8283	2

presented in Section 7.1 allows to tackle almost any real-case of overdamped plants that can be found in the industry.

6.3 LiTaO₃ Thin Film Deposition Process

Temperature control is a very important factor in the deposition process of lithium tantalate (LiTaO₃) by means of metal organic chemical vapor deposition (MOCVD) (Zhang et al, 2004).

The dynamics of the reactor chamber are characterized by a large lag and time-delay. It is important for the quality of the final product, that the controller follows a predefined temperature profile accurately (servo control) while being able to reject other disturbances (regulatory control).

The model of the MOCVD chamber can be given by:

$$G(s) = \frac{Ke^{-Ls}}{Ts + 1}, \quad (6.9)$$

where the gain $K = 3.2$, the time constant $T = 200$ s and the time-delay $L = 150$ s.

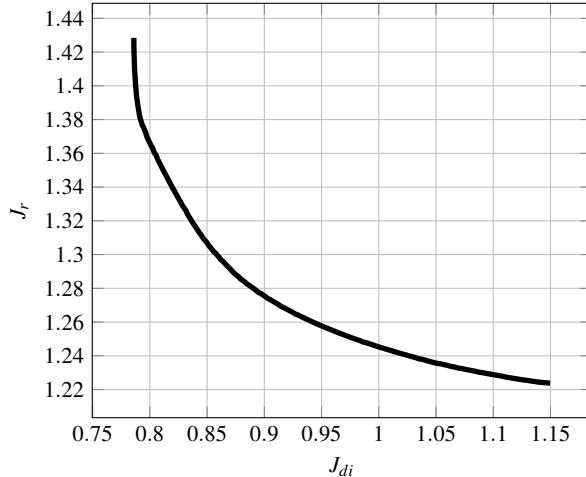


Fig. 6.21: Pareto front for the LiTaO₃ thin film deposition process.

constraint

For this case, a two function MOOP is considered with J_{di} and J_r as cost functions and a robustness ~~restriction~~ of $M_S = 2.0$. When solving the optimization using the ENNC method, the obtained Pareto front is as given in figure 6.21.

Again, the Pareto front let the decision maker to choose between multiple possible solutions. In this particular case, taking the anchor point for minimum value of J_{di} from Figure 6.21, it can be seen that, from this point, if J_{di} is degraded by 1.8%, it means an improvement of 4.37% for J_r . This information could only be possible if the Pareto is available in some way, either as a graph or as a set of raw data.

In order to help the control engineer to understand the tuning of the controller, it could be useful to plot the variation of the controller parameters as a function of the degradation of one of the cost functions. Given that the LiTaO₃ Thin Film Deposition Process requires to follow a given temperature profile, the control engineer may surely consider J_r as the main function.

In Figure 6.22 the values of all the controller parameters are plotted against m , where m is defined as the normalized degradation of J_r ($m = 0$ represents the anchor point where J_r has its lowest value). Looking the behaviour of K_p in Figure 6.22a, it is clear that the value of K_p is kept fairly constant for all values of m . This almost negligible variation may be associated to the fact that, for all controller tunings, the maximum sensitivity is set at $M_S = 2$. When this constraint is not taken into account, the value of K_p may have large variations as in the example in Section 8.1.

For the case of the integral time T_i , the behavior of this parameter is plotted in Figure 6.22b. Note that, contrary to the case of K_p , the variation is highly dependent of J_r and fairly smooth, which is desirable if the interest is to find a tuning rule. However, for the derivative time T_d in Figure 6.22c, it can be seen that the variation is important but the behavior is not as near as smooth as in the case of T_i . Finally and the setpoint weight factor β , also has a piece-wise behavior with respect to J_r .

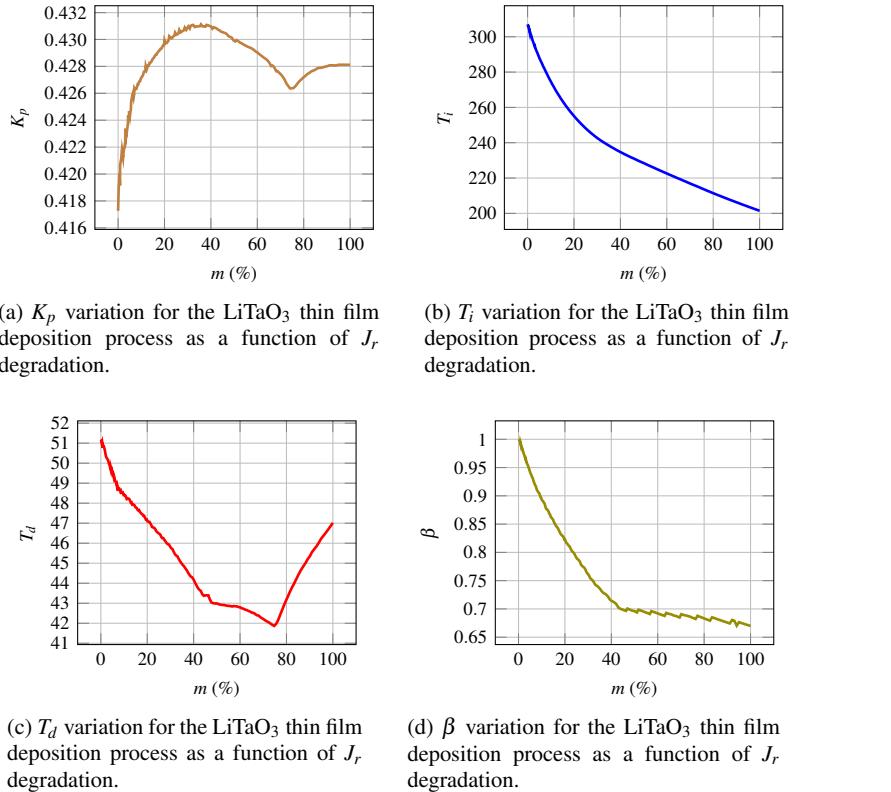


Fig. 6.22: Variation of the parameters vs J_r for the LiTaO₃ thin film deposition process.

the task is just to decide and pick

and also the variation of its value is important. Nevertheless, once computed the corresponding tunings of all the controllers of the Pareto, ~~is just to pick and decide~~ which one is more appropriate for the task.

The response of the controlled system to a setpoint step change is presented in figure 6.23 and for a step signal in the input disturbance in figure 6.24. For both cases, the anchor points controllers where compared against the uSORT₂ tuning rule (Alfaro and Vilanova, 2012b), since both uses a 2DoF PID controller structure and both attempt to minimize an IAE cost function.

Take into account the response presented in Figure 6.23. The response of the controlled system is compared against the uSORT₂ method as in Section 6.2 given that in this particular example the value of M_s is also considered. Given that the responses taken from the Pareto are the extreme cases, it is to be expected that all other responses in the Pareto are going to be between these two. Therefore, all the controllers in the Pareto surpass the servo response of the uSORT₂, which is not a

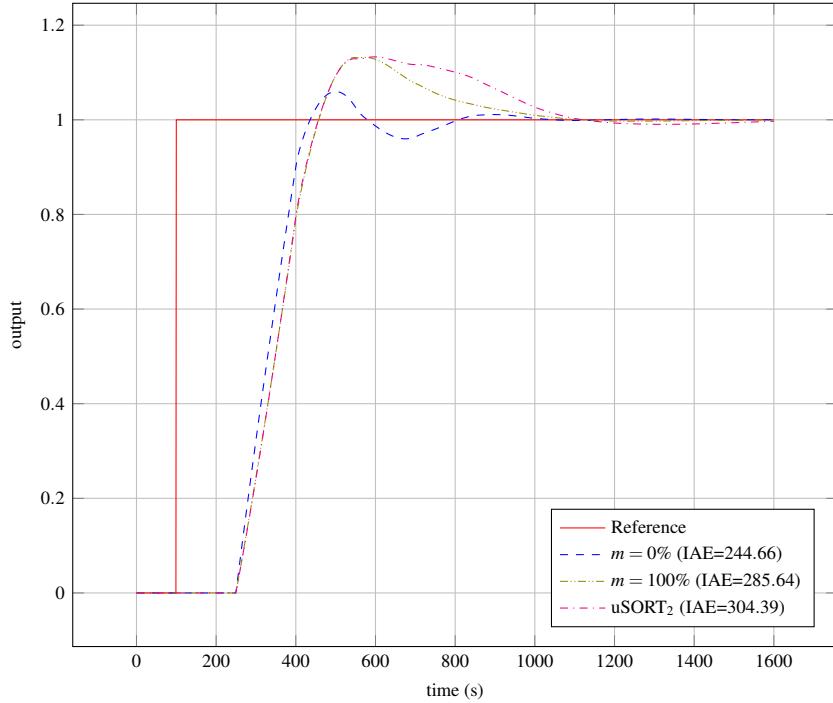


Fig. 6.23: Servo response of the LiTaO_3 thin film deposition process with three different tuning.

surprising result given that the uSORT₂ method is said to be suboptimal with respect to the servo response, as it is primarily optimized for regulation.

For this same reason, when examining the responses in Figure 6.24 for the disturbance rejection case, it can be seen that the uSORT₂ response is close to the case of $m = 100\%$ and this is exactly what it was expected since the uSORT₂ method was intended for regulation. Of course, the tuning of the other anchor point has a much worse response, and therefore, the uSORT₂ response lies between these two extremes. However, this fact does not mean that the uSORT₂ method is optimal in the Pareto sense, because, it may be possible to find a better servo response with a similar regulation IAE.

Finally to consciously be aware of the advantage of computing the Pareto front, in figure 6.25, five points of the Pareto front where selected ranging from $m = 100\%$ to $m = 0\%$ and the reference tracking response was plot in the same axis. As before, the front was found with the constraint $M_{S,\max} \leq 2.0$. The fact that, among all possible controllers computed, at the end only one of them is going to be selected, may be seen as a “waste” of resources. And this surely may be true if the Pareto is computed for every single case. However, if a general case is computed beforehand and a tool is used to select one of the many controllers, the Pareto front then can be seen more

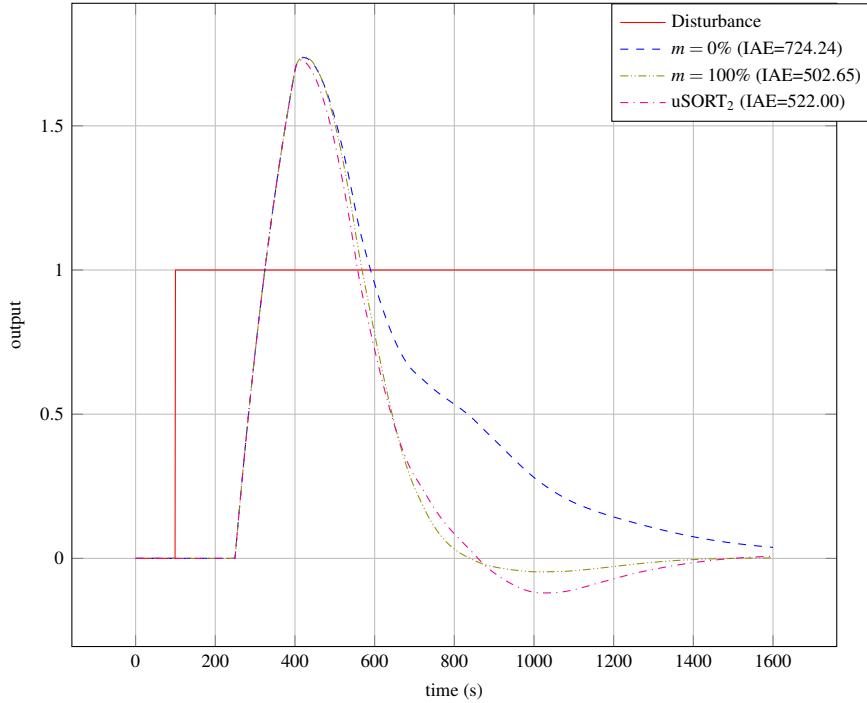


Fig. 6.24: Regulation response of the LiTaO₃ thin film deposition process with three different tuning.

advantageous and certainly more useful for the decision maker. An example of such case is presented in Section 8.1.3.

References

- Alfaro VM (2006) Identificación de Modelos de orden reducido a partir de la curva de reacción del Proceso. Ciencia y Tecnología (UCR) 24(2):2
- Alfaro VM, Vilanova R (2012a) Optimal Robust Tuning for 1DoF PI/PID Control Unifying FOPDT/SOPDT Models. IFAC Proceedings Volumes 45(3):572–577, DOI <http://dx.doi.org/10.3182/20120328-3-IT-3014.00097>, URL <http://www.sciencedirect.com/science/article/pii/S1474667016310874>
- Alfaro VM, Vilanova R (2012b) Set-point weight selection for robustly tuned PI/PID regulators for over damped processes. In: Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on, pp 1–7, DOI 10.1109/ETFA.2012.6489607
- Åström KJ, Hägglund T (1995) PID Controllers: Theory, Design and Tuning. Instrument Society of America
- Åström KJ, Hägglund T (2000) Benchmark Systems for PID Control. In: Proceedings IFAC Workshop Digital Control: Past, Present and Future of PID Control

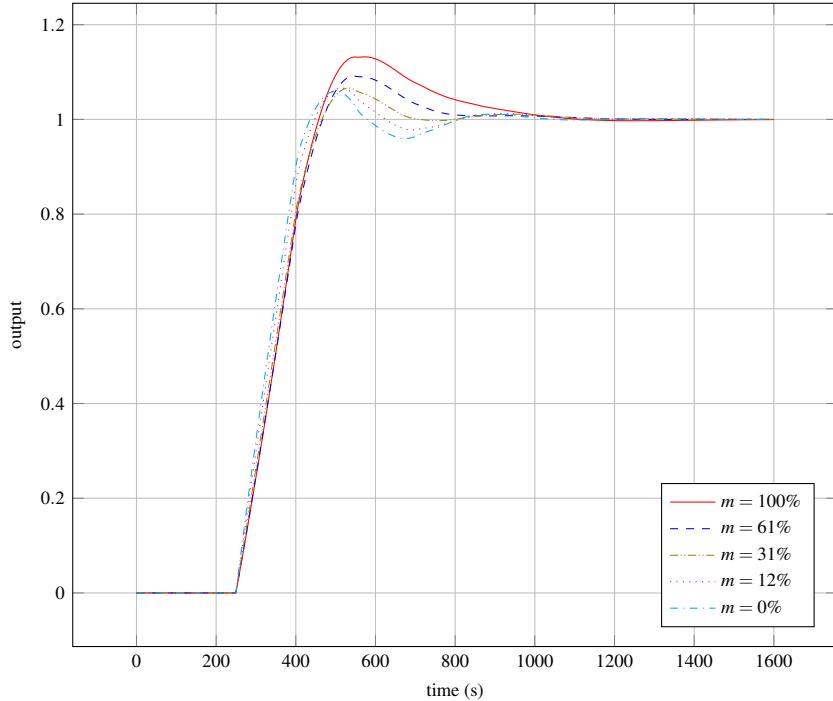


Fig. 6.25: Servo response of the LiTaO_3 thin film deposition process varying the tuning across the Pareto front with different degradations.

- Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization* 14(1):63–69, DOI 10.1007/BF01197559
- Grimholt C, Skogestad S (2012) Optimal PI-Control and Verification of the SIMC Tuning Rule. In: IFAC Conference on Advances in PID Control PID’12, Brescia (Italy)
- Messac A, Ismail-Yahaya A, Mattson C (2003) The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization* 25(2):86–98, DOI 10.1007/s00158-002-0276-1, URL <http://link.springer.com/10.1007/s00158-002-0276-1>
- Murril PW (1967) Automatic control of processes. International Textbook Co.
- O'Dwyer A (2000) A summary of PI and PID controller tuning rules for processes with time delay. Part 1: PI controller tuning rules. In: Proceedings of PID'00: IFAC Workshop on Digital Control, Terrassa, Spain, pp 175–180
- Rovira A, Murrill P, Smith C (1969) Tuning controllers for setpoint changes. *Instruments and Control Systems* 42:67–69
- Skogestad S (2003) Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13(4):291–309, DOI 10.1016/S0959-1524(02)00062-8, URL <https://www.sciencedirect.com/science/article/pii/S0959152402000628> <https://linkinghub.elsevier.com/retrieve/pii/S0959152402000628>
- Smith CA, Corripio AB (1985) Principles and practice of automatic process control. John Wiley and Sons, New York
- Vilanova R, Alfaro VM, Arrieta O (2011) Analytical Robust Tuning Approach for Two-Degree-of-Freedom {PI/PID} Controllers. *Engineering Letters (IAENG)* 19:204–214

- Zhang D, Huang D, He Y (2004) Intelligent temperature control strate for LiTaO₃ thin film deposition process. In: Intelligent Mechatronics and Automation, 2004. Proceedings. 2004 International Conference on, pp 339–344, DOI 10.1109/ICIMA.2004.1384216
- Ziegler J, Nichols N (1942) Optimum Settings for automatic controllers. ASME Transactions 64:759–768

Chapter 7

PID tuning as a multi-objective optimization problem

Abstract The proportional integral derivative (PID) tuning problem stated in Chapter 4 is solved using the enhanced normalized normal constraint (ENNC) methodology presented in Chapter 5. First the problem is solved using a MATLAB script that can be found in the appendix of this chapter and also downloaded as a companion software. The result of this script is a set of files that defines 2200 Pareto fronts with the optimal solutions of the problem of finding the tuning of two degrees of freedom (2DoF) PID controller for overdamped second order plus time delay (ODSOPTD) plant families. Then two possible approaches are presented to use this results: first an attempt to find a tuning rule based on this data is presented. This approach was found to be very difficult to apply given the complexity of the data. Then the data was used as a data base and a GUI was created to serve as the bridge between the user and the results. This GUI was encapsulated as a MATLAB app and is also included as the companion software for this book.

7.1 Solution of the multi-objective optimization tuning

When solving the multiobjective optimization problem (MOOP) presented in (4.14) for different normalized plants, one is able to find a family of Pareto fronts.

Defining the normalized variable $\hat{s} = Ts$, the time delay for the normalized plant, τ_0 , becomes:

$$\tau_0 = \frac{L}{T}, \quad (7.1)$$

Then, one is able to find the corresponding Pareto front for the normalized plant, that represents many combinations of lag time and dead time. The gain of the plant is considered to be included in the controller gain for the sake of the normalization.

The steps that are required to find the front for each Pareto is presented in Algorithm 1.

The Pareto front was found for each normalized plant with approximately 1000 points for each front. Using the steps of Algorithm 1, a total of 220 different nor-

Algorithm 1 Script for finding all Pareto fronts.

```

 $M_{svec} \leftarrow [10, 2, 1.8, 1.6, 1.4]$ 
 $\tau_{0vec} \leftarrow (0.1 : 0.1 : 2)$ 
 $a_{vec} \leftarrow (0 : 0.1 : 1)$ 
for all permutations of  $M_s \in M_{svec}$ ,  $\tau_0 \in \tau_{0vec}$  and  $a \in a_{vec}$  do
    Define the plant  $P(s)$  with  $\tau_0$  and  $a$ 
    Find initial tuning using Usort2 method
    Create cost function  $J(\theta) = [J_{di}(\theta, P(s), t), J_{do}(\theta, P(s), t), J_r(\theta, P(s), t)]$ 
    Create constraint function  $MCalc(\theta, P(s)) \leq M_s$ 
    Apply ENNC method to find the Pareto front
    Apply Pareto filter
    Compute actual  $M_s$  for each controller
    Save to file
end for

```

malized plants were considered. Also, five different values of desired maximum M_s were considered. Therefore a total of 1100 different cases were computed. For each case, a Pareto was found using the ENNC method with approximately 500 points for each front. Notice that each of these points represents a different tuning (and since the optimization was done for 2DoF controllers, each point has a different value for κ_p , τ_i , τ_d and β), the total possible Pareto optimal PID controllers found, reaches approximately 500 000 different controllers for all the totality of the plants and M_s . All these controller tunings, can also be found in the companion software of this book as comma separated values files.

In regard of the robustness constraint, it has to be noticed that the constraint is of a inequality kind, that is:

$$M_s(\theta, P(s)) \leq M_{smax},$$

in this particular case, the maximum sensitivity is set as 2.0, 1.8, 1.6 or 1.4. A maximum sensitivity of $M_s = 10.0$ was used as a way to relax the constraint such a manner that practically the optimization was done without constraints but the same script was used for the computation.

The function to compute the Pareto front with the ENNC method transform the multiobjective cost function into a single function using the scalarization method presented in Section 5.2.4. The function uses a standard optimization procedure¹ to find each of the points of the pareto. The ENNC function was based on the work of Houska et al (2011) and Logist et al (2012), the Pareto filter used was as in Cao (2020).

A very important part of the computation is the implementation of the cost function. It is desirable to have a convex function to minimize, since a global minimum is most likely to be found. However, when using the integral of the absolute value of the error (IAE) as a measure of the performance of the closed-loop response, the resulting cost function is not convex. In fact, the implementation of the cost function implies to do a simulation of the dynamic model and compute the integral of the absolute value of the error.

of

¹ The function `fmincon` of the MATLAB optimization toolbox was applied

functions and

It is common to implement this cost function using Simulink, and in general it is a straightforward way to compute the cost function. However, it implies to load all the power of Simulink with features that may be not be used to compute the IAE. Consider now that to compute one single point of the Pareto, it implies to use an optimization method with dozens of iterations and possibly hundreds of calls to the cost **functions**. And each of these call functions implies to call Simulink.

Remember that, to solve the problem presented in this section, it is expected to find around 1000 points for each of the 1100 different cases. Potentially, to find all the Paretos that are intended, it may be needed millions of calls to the Simulink implementation of the cost function. With this panorama, it is practically mandatory to find a faster implementation of the cost function.

In order to solve the problem in (4.14) the computation of the IAE was implemented with a hybrid approach between MATLAB and the C language. The idea is to avoid the call to Simulink. Instead, the differential equations of the closed-loop response were solved using an implementation of the fourth order Runge-Kutta numerical method. The simulation was implemented in C with the interface API provided by MATLAB. Using the `mex` instruction, the function is compiled and then called all the times needed. It was found that using this approach, the simulation time was reduced 97.7% while the average error with respect to the simulation in Simulink was 9.118×10^{-8} . The IAE is computed later in a MATLAB function with the results of the simulation. In all cases, a step input was used for all the sources of disturbance.

Finding all this data is time-consuming, even with the C implementation. It would be impractical to find the Pareto front every time it is needed, therefore, the best practice is to find the complete set of Pareto fronts once and save the results for later use.

However, there are two possible ways to use the data. In one hand, the data can be used to find a tuning rule that is as simple as possible that approximates the results of the optimizations. The other option is to create a software tool capable to access the files and interpolate the final tuning. In both cases, a way to incorporate the user preferences need to be addressed. Because, as stated in Section 5.3, finding the Pareto fronts is not the end of the optimization problem because a final single solution needs to be selected.

In the following, these two possible routes are considered with two examples on how it may be done. First, an example on how to find a possible tuning rule using the data for dead-time dominated processes is presented. The obtained results are good, however, given the complexity of the data, the tuning rule turned out not to be as simple as desired. Later, a computer-aided design (CAD) program is presented that is capable of finding a controller tuning based on the idea of “Maximum allowed degradation” of the cost function. In short, this tool let the user to navigate into the Pareto front, without the need to visualize the Pareto, at the same time that let the user to select the final tuning.

7.2 Viability for tuning rules

In this section, two examples on how to use the gathered data from the Pareto front is presented in order to find a tuning rule.

The first example use a reduced set of data where only J_{di} and J_{do} cost functions are considered and a proportional integral (PI) controller is tuned. The decision variable is the allowed degradation of the J_{di} cost function. These results were first documented in Contreras Leiva and Rojas (2015).

The second example shows the results of a PID controller tuning that takes into account all three proposed sources of disturbances (J_{di} , J_{do} and J_r) for plants that are delay dominant. These results where first documented in Moya et al (2017).

7.2.1 Tuning of a PI controller with two cost functions

In this case, the Pareto front was conformed just by the cost functions J_{di} and J_{do} and with a simple first order model given by:

$$P(s) = \frac{Ke^{-Ls}}{Ts + 1}. \quad (7.2)$$

However, only PI controllers where considered. The set of data is not equal to the presented in Section 7.1, instead, this data was found using the LCR scalarization Rojas et al (2015), but the idea behind the tuning is the same.

In this case, only the feedback controller $C_y(s, \theta)$ of Figure 4.2 is considered. After analyzing the data, it was found that one possible function that represents the variation of κ_p with respect to the degradation variable α is given by:

$$\kappa_p = a(\alpha) + b(\alpha)\tau_0^{c(\alpha)}, \quad (7.3)$$

where the parameters are given by:

$$\begin{aligned} a(\alpha) &= a_1\alpha + a_2\alpha^2, \\ b(\alpha) &= b_0 + b_1\alpha, \\ c(\alpha) &= c_0 + c_1\alpha^2. \end{aligned}$$

For the case of the integral time, the proposed rule is given by:

$$\tau_i = d(\alpha) + e(\alpha)\tau_0^{f(\alpha)}, \quad (7.4)$$

where the parameters can be computed as:

Table 7.1: Constants associated with κ_p in (7.3).

	$M_s \leq 2$	$M_s \leq 1.8$	$M_s \leq 1.6$
$0.1 \leq \tau_0 \leq 0.5$			
a_1	0.2887	0.18960	0.08484
a_2	-0.8579	-0.3039	-0.1811
b_0	0.6053	0.5356	0.4500
b_1	0.3205	0.1592	0.1161
c_0	-0.9917	-0.9928	-0.9897
c_1	0.1341	0.04427	0.03272
$0.6 \leq \tau_0 \leq 2.0$			
a_1	0.08305	0.001233	0.003224
a_2	-0.05774	-0.02813	-0.01945
b_0	0.8470	0.6964	0.53560
b_1	-0.06829	0.07708	0.05128
c_0	-0.6934	-0.6921	-0.7940
c_1	0	0	0

$$\begin{aligned} d(\alpha) &= d_0 + d_1\alpha + d_2\alpha^2, \\ e(\alpha) &= e_0 + e_1\alpha, \\ f(\alpha) &= f_0 + f_2\alpha^2. \end{aligned}$$

As it can be seen, to compute κ_p and τ_i , it is necessary to know 13 different constants. These constants are found using a curve fitting procedure that depends on the normalized dead-time and the maximum sensitivity required. The values found for these constant^s are presented in Table 7.1 for the constants associated with κ_p and in Table 7.2 for the constants associated with τ_i . It is important to note that, in order to found these equations, a series of curve fitting procedures were required. This is not a trivial task and even for a simple model like the one used in this example and a reduced controller, the quantity of computation needed is very high.

This is an example on how the Pareto front can be used as the basis to create tools that permit to extract the results without using the Pareto directly. Of course, the functions that are proposed need to follow closely the results obtained with the optimizations. In Table 7.3 the mean R-squared value is presented for the case of the regression on the κ_p variable and for the case of τ_i in Table 7.4. As it can be seen, the selected function and parameters fit the data well, which should give the user confidence that using (7.3) and (7.4) will most likely give a Pareto optimal results.

The proposed tuning rule is comparable to other optimal tuning rules as in Murril (1967) and Alfaro and Vilanova (2012), because in all cases, an integral cost function is optimized. In the case of the Murrill tuning (Murril, 1967), the IAE is minimized for disturbance rejection, while in the case of the Usort method (Alfaro and Vilanova, 2012), the disturbance rejection response is also minimized, but the maximum sensitivity is taken into account as a measure of robustness. *vsORT*

Table 7.2: Constants associated with τ_i in (7.4).

	$M_s \leq 2$	$M_s \leq 1.8$	$M_s \leq 1.6$
$0.1 \leq \tau_0 \leq 0.4$			
d_0	0	0	0
d_1	1.3840	0.9996	0.9685
d_2	-0.4286	-0.07516	-0.05125
e_0	1.4440	1.3590	1.3110
e_1	-1.179	-1.0500	-1.0970
f_0	0.6197	0.5474	0.4775
f_2	0	0	0
$0.5 \leq \tau_0 \leq 2.0$			
d_0	0.4483	0.5623	0.6746
d_1	0.5070	0.3530	0.3607
d_2	0	0	0
e_0	0.8717	0.6029	0.3793
e_1	-0.418	-0.1828	-0.2590
f_0	0.6532	0.7480	0.7055
f_2	0.2085	0.0114	0.4632

Table 7.3: Approximation indexes for κ_p in (7.3).

M_s	τ_0	R-squared
2.0	$\tau_0 \leq 0.5$	0.9982
	$\tau_0 \geq 0.6$	0.9929
1.8	$\tau_0 \leq 0.5$	0.9994
	$\tau_0 \geq 0.6$	0.9945
1.6	$\tau_0 \leq 0.5$	0.9996
	$\tau_0 \geq 0.6$	0.9971

Table 7.4: Approximation indexes for τ_i in (7.4).

M_s	τ_0	R-squared
2.0	$\tau_0 \leq 0.4$	0.9925
	$\tau_0 \geq 0.5$	0.9947
1.8	$\tau_0 \leq 0.4$	0.9978
	$\tau_0 \geq 0.5$	0.9986
1.6	$\tau_0 \leq 0.4$	0.9960
	$\tau_0 \geq 0.5$	0.9936

For these two methods, only the input disturbance is taken into account, while in the proposed method of this example, the user can vary the response between being optimal to the input disturbance or optimal to the output disturbance, given more flexibility to the tuning. A controller was tuned with these three methods and

/ Seen
compared in Table 7.5. As it can be from this table, for $\tau_0 = 0.1$ and $\tau_0 = 0.5$,

Table 7.5: Comparison of the performance of the closed-loop J_{di} .

τ_0	Tuning	IAE	M_s
0.1	Usort	0.0615	2.0016
	Murrill	0.0425	3.8956
	Proposed tuning	0.1081	2.0399
0.5	Usort	0.7388	2.0099
	Murrill	0.6186	2.9757
	Proposed tuning	0.8330	1.7757
1.0	Usort	1.6760	2.0047
	Murrill	1.6710	2.1831
	Proposed tuning	1.5730	2.0366

Murril presents the best performance, but in both cases, the value of M_s is rather high, which is undesirable for real control loops. The method presented here gives results very similar to the ones obtained with Usort which is expected, since both methods minimize an integral cost function with M_s as the robustness constraint. The advantage of the method shown in this example is that the user has an extra variable to fine tune the response.

7.2.2 Tuning for a Three-objective PID controller

no se ha dicho que se refiere al modelo de 2º orden.

For this, the variable a takes values from 0 to 1 in 0.1 steps and τ_0 takes values from 1 to 2 in 0.1 steps. However, for this next example, the maximum sensitivity is constraint to $M_{s,max} = 2$.

The methodology after finding the data is to perform a curve fitting procedure to find equations useful to compute the value of κ_p , τ_i , τ_d and *beta* as a function of a and τ_0 and a factor of degradation of the cost functions.

Notice that the idea is that, knowing the model of the plant, the values of the controller parameters can be computed without needing to perform all the optimization, and also the decision maker can also select the weight for each cost function in order to find a single set of parameters.

This idea of "allowed degradation" is now introduced. Considered that J_{di} and J_{do} are normalized as:

$$\delta = \frac{J_{di}(\theta) - J_{di,min}(\theta)}{J_{di,max}(\theta) - J_{di,min}(\theta)}, \quad (7.5)$$

$$\gamma = \frac{J_{do}(\theta) - J_{do,min}(\theta)}{J_{do,max}(\theta) - J_{do,min}(\theta)}, \quad (7.6)$$

where both $0 \leq \delta \leq 1$ and $0 \leq \gamma \leq 1$. Then, these variables can be understood as the degradation of the function, considering the minimum value of the cost function as its optimal. Then a value of $\delta = 1$ represents a degradation of 100% of the J_{di} cost function. It is important to notice that the Pareto front is constructed from three different cost functions. Therefore, if one selects the value of the allowed degradation for two functions (in this case, δ and γ), the logical step is to choose the lowest value of J_r that complies with the maximum degradation of the other two functions. Then for example, if $\delta = \gamma = 1$, which means that the decision maker is willing to ~~allowed~~ allow a complete degradation of J_{di} and J_{do} , the resulting tuning is expected to represent the optimal tuning for servo control.

Now, it is important to understand that the “degraded” tuning, is also optimal in the Pareto sense, because all found tunings are optimal. Therefore, in these frame, a degraded tuning does not mean a “bad” tuning, it is just the result of a choice decision when selecting the final optimal controller. In all Pareto decisions, a compromise has to be made when selecting the final solution.

The work done to find the tuning rules, summed up to almost two hundred and twenty regressions for all values of a and τ_0 in order to find the complete set of parameters $\boldsymbol{\theta}$.

After different heuristic tests, the regression analysis showed that a second order fit gave the best results for κ_p , τ_i and τ_d , while a first order fit for β was enough to model the variation of this parameter.

The tuning rule for all controller parameters are proposed to be as:

$$\begin{aligned} \kappa_p &= p_{00} + p_{01} \cdot \gamma + p_{02} \cdot \delta \\ &\quad + p_{03} \cdot \gamma^2 + p_{04} \cdot \gamma \cdot \delta + p_{05} \cdot \delta^2, \end{aligned} \quad (7.7)$$

$$\begin{aligned} \tau_i &= p_{10} + p_{11} \cdot \gamma + p_{12} \cdot \delta \\ &\quad + p_{13} \cdot \gamma^2 + p_{14} \cdot \gamma \cdot \delta + p_{15} \cdot \delta^2, \end{aligned} \quad (7.8)$$

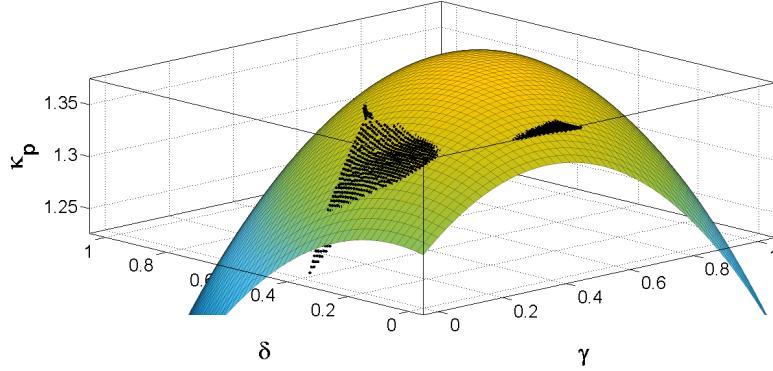
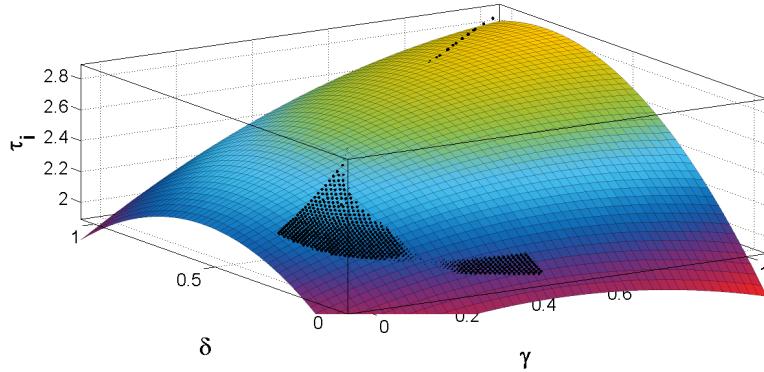
$$\begin{aligned} \tau_d &= p_{20} + p_{21} \cdot \gamma + p_{22} \cdot \delta \\ &\quad + p_{23} \cdot \gamma^2 + p_{24} \cdot \gamma \cdot \delta + p_{25} \cdot \delta^2, \end{aligned} \quad (7.9)$$

$$\beta = p_{30} + p_{31} \cdot \gamma + p_{32} \cdot \delta, \quad (7.10)$$

The coefficients p_{ij} , where $i = \{0, 1, 2, 3\}$ and $j = \{0, 1, 2, 3, 4, 5\}$, depend on a and τ_0 . The corresponding fits of κ_p , τ_i , τ_d and β , are shown in Fig. 7.1, 7.2, 7.3 and 7.4 for $a = 0.1$ and $\tau_0 = 1$.

It is important to notice an important caveat. In those figures, the values that belong to the computed Pareto are shown as dots, while the corresponding regression is plotted as a 3D surface. It can be noticed that the domain of the regressions is larger than the actual results of the Pareto. Even though the fitting is good (around $R = 0.9$), the regression represents interpolation and extrapolation from the real data. Therefore, it is important to check how well the regression works and to not exceed the limits where it yields good results.

Going back to the p_{ij} , it has to be noticed that the value of these parameters, depends on the model of the plant. Therefore, it is required to find another set of

Fig. 7.1: Second order fit for κ_p when $a = 0.1$ and $\tau_0 = 1$ Fig. 7.2: Second order fit for τ_i when $a = 0.1$ and $\tau_0 = 1$

regressions over these parameters in terms of a and τ_0 . Therefore, a curve fitting procedure is also required for each p_{ij} . As an example of these regressions, Fig. 7.5 shows the result for p_{00} parameter as a function of a and τ_0 . The selected fit for every coefficient in the range of $1 \leq \tau_0 \leq 2$, was also a second order polynomial. The equation that is considered has the form:

$$p_{ij} = b_{j0} + b_{j1}a + b_{j2}\tau_0 + b_{j3}a^2 + b_{j4}a\tau_0 + b_{j5}\tau_0^2. \quad (7.11)$$

Another two hundred and twenty regressions were made for all p_{ij} . The results for every coefficient are shown in Table 7.6 for κ_p , Table 7.7 for the integral time, Table 7.8 for the derivative time and Table 7.9 for β .

K

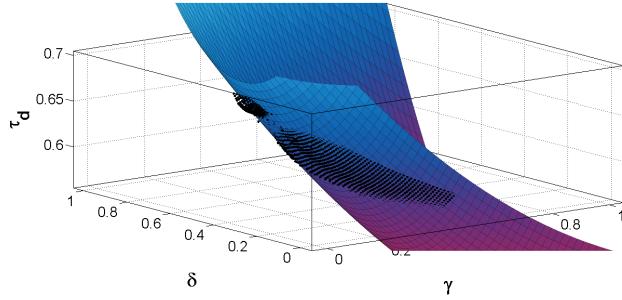


Fig. 7.3: Second order fit for τ_d when $a = 0.1$ and $\tau_0 = 1$

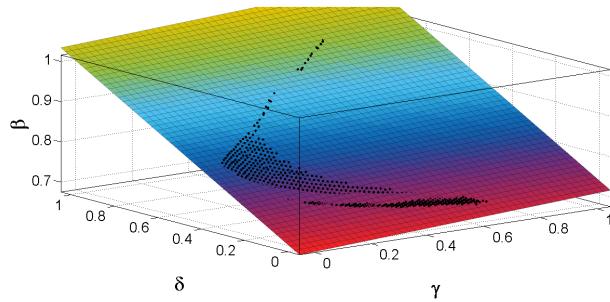


Fig. 7.4: First order fit for β when $a = 0.1$ and $\tau_0 = 1$

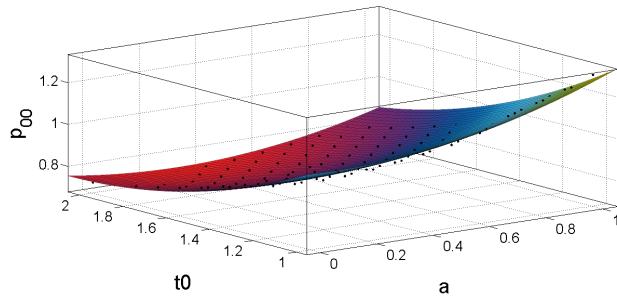


Fig. 7.5: Second order fit for p_{00} in κ_p

7.2.2.1 Comparison of regression against Pareto data

To compare the results from the tuning rule, some simulations were done to compare the original data against the results. The plant model is:

Table 7.6: Coefficients for κ_p .

p_{ij}	b_{ik}	p_{ij}	b_{ik}
p_{00}	b_{00} 1.820	p_{10}	b_{10} 0.328
	b_{01} 0.128	p_{11}	b_{11} 0.224
	b_{02} -1.048	p_{12}	b_{12} -0.268
	b_{03} 0.270	p_{01}	b_{13} -0.022
	b_{04} -0.151	p_{14}	b_{14} -0.069
	b_{05} 0.255	p_{15}	b_{15} 0.076
p_{02}	b_{20} 0.291	p_{30}	b_{30} 0.043
	b_{21} -0.129	p_{31}	b_{31} -0.520
	b_{22} -0.250	p_{32}	b_{32} -0.254
	b_{23} 0.105	p_{03}	b_{33} 0.473
	b_{24} 0.005	p_{34}	b_{34} -0.111
	b_{25} 0.059	p_{35}	b_{35} 0.079
p_{04}	b_{40} -0.077	p_{50}	b_{50} -0.412
	b_{41} 0.611	p_{51}	b_{51} -0.247
	b_{42} 0.249	p_{52}	b_{52} 0.296
	b_{43} -0.603	p_{05}	b_{53} 0.080
	b_{44} 0.197	p_{54}	b_{54} 0.013
	b_{45} -0.071	p_{55}	b_{55} -0.091

Table 7.7: Coefficients for τ_i .

p_{ij}	b_{ik}	p_{ij}	b_{ik}
p_{10}	b_{00} 0.591	p_{10}	b_{10} -0.408
	b_{01} 0.559	p_{11}	b_{11} 0.640
	b_{02} 0.545	p_{12}	b_{12} 0.855
	b_{03} 0.017	p_{13}	b_{13} -0.238
	b_{04} 0.045	p_{14}	b_{14} -0.0024
	b_{05} -0.028	p_{15}	b_{15} -0.193
p_{12}	b_{20} 1.718	p_{30}	b_{30} 1.297
	b_{21} 0.652	p_{31}	b_{31} -0.423
	b_{22} -1.160	p_{32}	b_{32} -2.095
	b_{23} -0.855	p_{13}	b_{33} 1.226
	b_{24} -0.719	p_{34}	b_{34} -1.041
	b_{25} 0.363	p_{35}	b_{35} 0.649
p_{14}	b_{40} -0.077	p_{50}	b_{50} -1.346
	b_{41} 0.621	p_{51}	b_{51} -1.148
	b_{42} 0.277	p_{52}	b_{52} 1.224
	b_{43} -1.193	p_{15}	b_{53} -0.218
	b_{44} 1.030	p_{54}	b_{54} 0.512
	b_{45} -0.025	p_{55}	b_{55} -0.572

$$P_1(s) = \frac{e^{-1.5\hat{s}}}{(\hat{s}+1)(0.5\hat{s}+1)} \quad (7.12)$$

Where $K = 1$, $T = 1$ s, $L = 1.5$ s and $a = 0.5$. Table 7.10 compares the results of the optimization against the results of using the proposed tuning rule. Arbitrarily, the values for δ and γ were chosen as $\delta = 1$ and $\gamma = 1$.

Table 7.8: Coefficients for τ_d .

p_{ij}	b_{ik}	p_{ij}	b_{ik}
p_{20}	b_{00} 0.111	p_{21}	b_{10} -0.0076
	b_{01} 0.450		b_{11} -0.163
	b_{02} 0.274		b_{12} -0.212
	b_{03} -0.025	p_{23}	b_{13} 0.154
	b_{04} -0.069		b_{14} -0.074
	b_{05} 0.003		b_{15} 0.0026
p_{22}	b_{20} -0.238	p_{23}	b_{30} -0.237
	b_{21} 0.105		b_{31} -0.938
	b_{22} -0.016		b_{32} 1.121
	b_{23} -0.234	p_{24}	b_{33} 0.496
	b_{24} 0.094		b_{34} 0.331
	b_{25} -0.0254		b_{35} -0.641
p_{24}	b_{40} 0.379	p_{25}	b_{50} -0.224
	b_{41} 0.908		b_{51} 0.109
	b_{42} -1.330		b_{52} 0.805
	b_{43} -1.203	p_{25}	b_{53} 0.669
	b_{44} 0.215		b_{54} -0.527
	b_{45} 0.683		b_{55} -0.112

Table 7.9: Coefficients for β .

p_{ij}	b_{ik}
p_{30}	b_{00} 0.538
	b_{01} 0.023
	b_{02} 0.179
	b_{03} -0.114
	b_{04} 0.047
	b_{05} -0.034
p_{31}	b_{10} -0.152
	b_{11} 0.065
	b_{12} 0.277
	b_{13} 0.017
	b_{14} -0.052
	b_{15} -0.082
p_{32}	b_{20} 0.585
	b_{21} -0.082
	b_{22} -0.280
	b_{23} 0.116
	b_{24} 0.011
	b_{25} 0.044

From Table 7.10, can be seen that the results obtained from the tuning rule are close to those obtained directly from the Pareto. Plots for each method were drawn as shown in Fig. 7.6. The step response of the control signal is shown in Figure 7.7 while the comparison for an input-disturbance is presented in Figure 7.8 and for the output disturbance is in Figure 7.9. The figures show the IAE between both signals as a measure of how good the tuning rule approximates the optimization. As

the control signal for a reference step response

Table 7.10: Result comparative of the Pareto data against the fitted data, with $\delta = 1$ and $\gamma = 1$.

θ and cost functions	From Pareto	From Tuning rule
κ_p	0.810	0.793
τ_i	2.176 s	2.113 s
τ_d	0.644 s	0.720 s
β	1.000	1.000
J_r	2.689	2.691
J_{di}	2.687	2.673
J_{do}	2.689	2.691
M_s	1.9174	1.9449

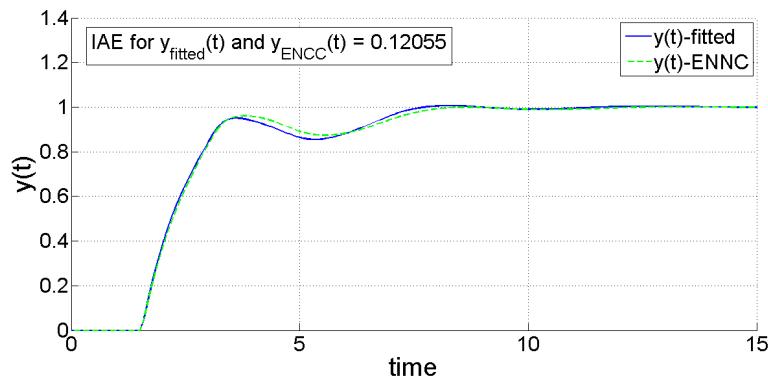


Fig. 7.6: Servo response for the Pareto results found with the normalized normal constraint (NNC) method and the tuning results.

it can be seen, the responses are almost identical, showing that this methodology is feasible.

The tuning rule is also used for the extreme cases of τ_0 , that is, $\tau_0 = 1$ and $\tau_0 = 2$, using the following models:

$$P_F(\hat{s}) = \frac{e^{-\hat{s}}}{(\hat{s}+1)(0.5\hat{s}+1)}, \quad (7.13)$$

$$P_S(\hat{s}) = \frac{e^{-2\hat{s}}}{(\hat{s}+1)(0.5\hat{s}+1)}, \quad (7.14)$$

where $P_F(\hat{s})$ and $P_S(\hat{s})$ stand for the minimum and maximum dead time considered in this study, respectively.

The allowed degradation was set to $\delta = 0.5$ and $\gamma = 0.5$. As before, one would expect to have the best servo response that complies with the allowed degradation.

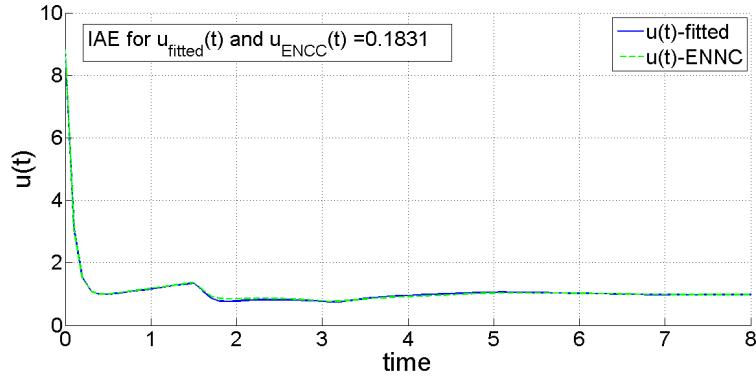


Fig. 7.7: Comparison of the control action signal for a setpoint step change using the data from the Pareto found with the ENNC method and the tuning rule.

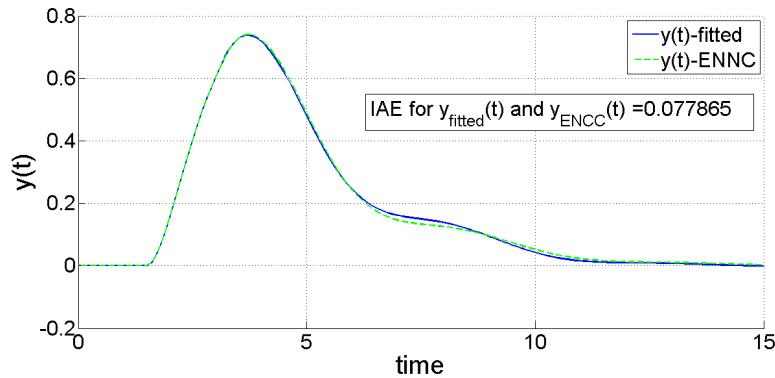


Fig. 7.8: Step input disturbance response for tuning from Pareto (ENNC method) and regressions results.

The comparison between the Pareto optimizations and the tuning rule for P_F and P_S are shown in Table 7.11 and Table 7.12 respectively.

It is clear that the tuning rule is able to produce near Pareto-optimal controllers (in both cases, the maximum error is in τ_d).

One of the interesting features is that the decision maker is able to choose the final solution by given a suitable value to δ and γ as he or she considers appropriate. Since the data used to find the tuning rule had the ~~constraint~~ to have a maximum sensitivity of $M_s = 2.0$ it is expected to have a stable closed-loop. Another interesting characteristic of this tuning rule is its ability to select the appropriate parameters taking into account three different sources of disturbances, unlike other PID tuning rules.

constrained

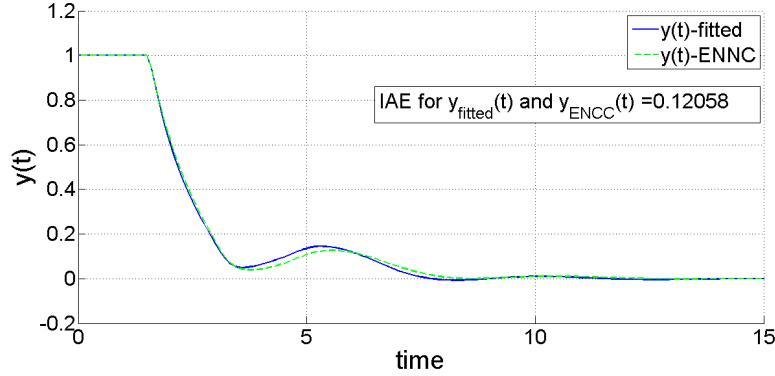


Fig. 7.9: Step output disturbance response for ENNC results and regressions results.

Table 7.11: Results for J_{di} , J_{do} and J_r , using $\delta = 0.5$ and $\gamma = 0.5$ for $P_F(s)$.

θ and IAE	From tuning rule	From Pareto	Difference (%)
κ_p	1.150	1.120	2.67
τ_i	1.987 s	1.900 s	4.58
τ_d	0.425 s	0.495 s	-14.14
β	0.887	0.889	-0.23
J_r	1.955	2.1446	8.84
J_{di}	1.729	1.6947	2.02
J_{do}	1.874	1.803	3.94
M_s	2.024	2.013	0.55

Table 7.12: Results for J_{di} , J_{do} and J_r , using $\delta = 0.5$ and $\gamma = 0.5$ for $P_S(s)$.

θ and IAE	From tuning rule	From Pareto	Difference (%)
κ_p	0.742	0.744	-0.270
τ_i	2.345 s	2.264	3.578
τ_d	0.629 s	0.712	11.657
β	0.919	0.927	-0.863
J_r	3.360	3.662	-8.247
J_{di}	3.162	3.065	3.165
J_{do}	3.237	3.158	2.502
M_s	1.976	2.000	-0.012

From Table 7.11 and 7.12 it can be deduced that the tuning rule finds a controller that has a better servo response than using the data directly, but compromising the response to the input and output disturbance rejection.

7.2.3 Comments on creating tuning rules from Pareto fronts

With the example⁹ presented above, it was clear that it is feasible to find tuning rules from Pareto fronts. However, there are several points that need to be addressed:

- The tuning rule was intended to be as simple as possible. However, it needed 126 coefficients to find the four parameters of the controller. Compared with other tuning rules O'Dwyer (2009), this tuning rule is complex.
- The idea of the degradation factor is interesting and directly related to the Pareto front, however, is not as intuitive as setting something more measurable, as the time constant of the closed-loop system for example.
- The tuning rule is restricted to values of $1 \leq \tau_0 \leq 2$. The data for other values of τ_0 exist, in fact, the reader can download the complete set of data from the companion software for this book. However, the complexity of the data made unfeasible to find a good simple tuning rule for all possible values of τ_0 .
- The tuning rule is also restricted to PID controllers. But it is very common to use PI controllers in the industry. However, it is not possible to just discard the derivative time from the obtained tuning.
- The data obtained from the optimizations has the constraint to have certain Maximum Sensitivity M_s . However, the presented rule takes into account only the value of $M_s = 2$. In order to have tuning rules for other values of M_s , possibly another set of 126 coefficients needs to be found for each desired value, which requires a lot of effort that may be not worth it.

All these points raise an important question, is it useful to find tuning rules that becomes too cumbersome for setting a PID controller? The literature about PID tuning (see O'Dwyer (2000) for a list of PID tuning rules) generally shows simple tuning rules that need only a few decision parameters (for example Skogestad (2003)) or even no decision parameters, since they minimize a single cost function (as in the MoReRT tuning rule (Alfaro and Vilanova, 2016)), and therefore, only one tuning is possible. However, using the multiobjective approach presented here, the relationships between the controller gains, the tuning parameters and the model parameters become so complex, that a simple tuning rule that compasses all cases is impossible to find.

In these cases⁹ a more direct approach may be more suitable for the task of finding the best controller tuning. It is true that the Pareto front is not the final solution for the tuning problem, however only a selection is needed to ultimately find the desired solution. Therefore, a *database* approach may be more sensible to the task at hand. This option is explored in the next section.

7.3 Database approach for the final tuning

The information of the Pareto front is very valuable, each point represents an optimal controller tuning that also complies with the robustness criterion. However, without

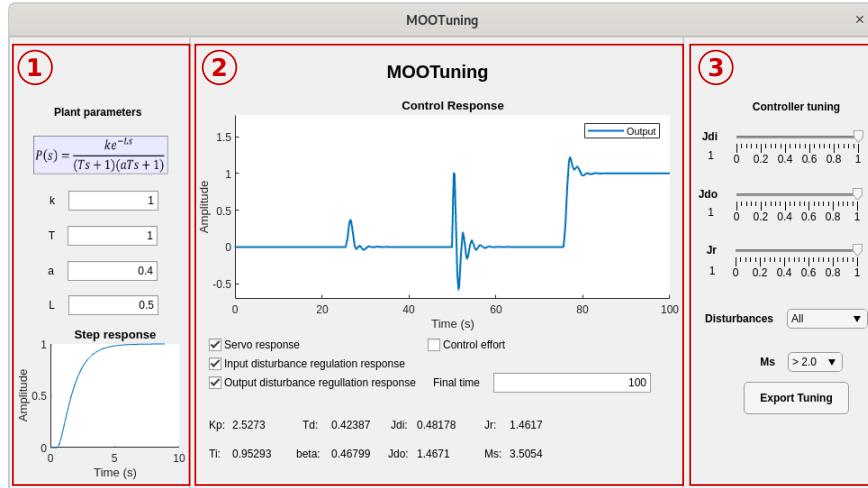


Fig. 7.10: Interface of MOOTuning for PID parameter selection.

any guidance about how to choose the final tuning, the data becomes useless. The idea of using the Pareto as the basis for a tuning rule was explored in Section 7.2 for two degrees of freedom PID controllers taking into account three different sources of disturbances and a robustness constraint for a ODSOPTD plant. However it was found that a simple rule is very difficult to find, given the complexity of the relationship between the different parameters. For simpler cases, it may be possible to find suitable tuning rules (as in Contreras Leiva and Rojas (2015)) but for a more realistic case, the final tuning becomes cumbersome.

In this scenario, the other approach to take advantage of the information in the Pareto is to actually use the data directly. Visualize the Pareto is also a difficult task, specially for more than two cost functions. Therefore, the approach that is presented here is to use a CAD that ~~lets~~ allows the user to select the desired closed-loop performance according to their needs without the need to plot the front.

The proposed CAD tool that accompanies this book is named MOOTuning, and is available as a MATLAB app. An screenshot of the tool is presented in Figure 7.10. It has three main components:

1. Plant parameters input section
2. Results section
3. Tuning section

The plant parameters input section is used to enter the parameters of the model of the plant. The expected model is an ODSOPTD, but the value of a can be set to zero, given the option to have also a first order plus time delay (FOPTD) model. However, it is necessary to have a normalized ~~delay time~~ time delay larger than 0.1. The tool warns the user when an invalid value is entered (as a negative time constant for

time delay

example). When any of these parameters are changed, the step response plot at the bottom is automatically updated.

The second section shows the user the results of the selected tuning. The main component is a figure where the closed-loop response is plotted for the selected sources of disturbance: *Servo response* is the closed-loop response to a setpoint step change, *Input disturbance regulation response* plots the closed-loop response to a step change at the input of the plant and *Output disturbance regulation response* plots the closed-loop response to step change at the output of the plant. The control effort can be plotted as well in the same graph.

At the bottom of the second section, the results of the controller parameters, the value of the cost function and the value of the maximum sensitivity are presented with the given tuning. These are updated every time the user makes a change in the third section of the tool.

The last section of the tool is the Tuning section. Here, the user is presented with a series of “decision choices” that defines which of the points of the front is selected as the final tuning. The sliders at the top represents the *allowed degradation* of the function with respect to the optimal point. A value of zero means that the lowest value of the cost function is desired, while a value of 1 represents that the function can have any value (even its maximum value).

The tool select^s the function with the lowest allowed degradation as the main cost function. Then, it searches for a set of parameters that comply with all the degradation limits set by the user, that also has the lowest possible value of the main function. As it can be seen, this tool let^s the user select^s the desired value from the Pareto, without the need to plot the front.²

The user has also the ability to select if all the cost function need^s to be used for the tuning, or if only the input disturbance and the setpoint change^s are considered. Finally, the user can select different values of Maximum Sensitivity, to set the robustness of the closed-loop system. The *Export Tuning* button let the user to copy the values of the tuning and export them as an structure to the MATLAB workspace.

Of course, it may be possible that the user selects a set of values for J_{di} , J_{do} and J_r that do not correspond to a Pareto point. In that case, the tool shows a window indicating that the current selection is not feasible. This window is shown in Figure 7.11. The user then needs to relax the allowed degradation of the cost function (i.e. increase the allowed degradation of one of the functions) to be able to find the appropriate tuning. Once the user is satisfied with the design, the tuning can be exported to the MATLAB workspace and the figure can be saved using the standard MATLAB methods².

² The tool was created using MATLAB 20019b, earlier versions may not work as expected.

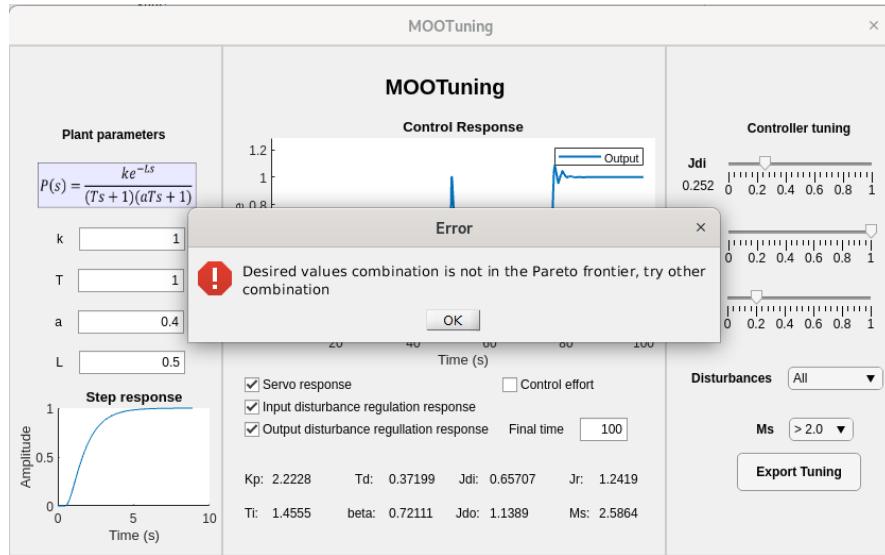


Fig. 7.11: Error raised if the desired point lies outside the Pareto front.

7.3.1 Example using MOOTuning

In the following, a typical workflow example is presented using the proposed tool. Suppose that the model of the plant to be controlled is given by:

$$H(s) = \frac{0.6e^{-5s}}{(10s + 1)(2s + 1)}. \quad (7.15)$$

From this transfer functions, it can be seen that the parameters are given by $k = 0.6$, $T = 10$, $L = 5$ and $a = 0.2$. The first step then consists of introducing these parameters into the tool, as shown in Figure 7.12. Everytime a parameter is introduced, the tool updates the step response of at the bottom of the screen. In this case, it can be seen that the plant takes approximately 50s to reach steady state. It has to be noticed that the tuning of the controller is not going to be updated until the allowed degradation of any functions is changed.

For this particular example, only two cost functions are going to be considered (J_{di} and J_r) with the option of $M_z > 2.0$. Originally, the allowed degradation is one for both functions, which means that the response presented has the lowest IAE for servo response without any ~~restriction~~ on the input disturbance response. With this characterization, the initial closed loop response is as presented in Figure 7.13. The response to an input disturbance and to a change in the setpoint is presented in the Tool. However, it can be seen that 100 s are not enough to see the complete response. With a 200 s time window, the closed-loop response is given as presented in Figure 7.14. As it can be seen the, disturbance rejection response is rather slow,

constraint

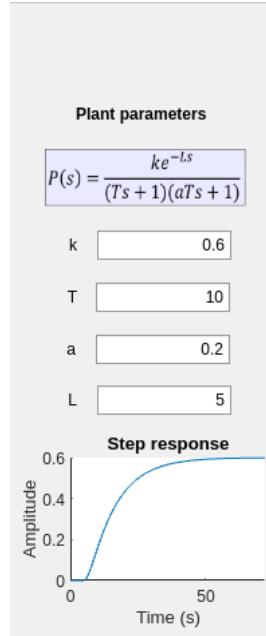


Fig. 7.12: Input of the parameters of the model into MOOTuning.

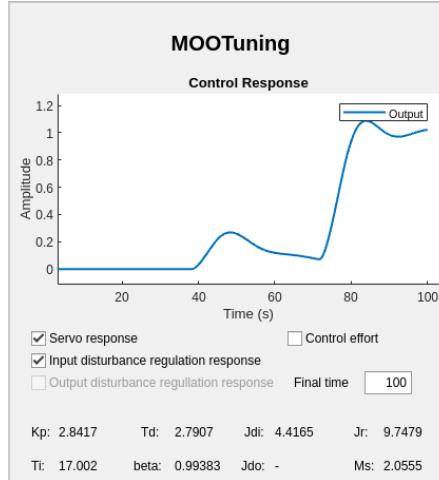


Fig. 7.13: Initial closed-loop response.

and the servo response is appropriate. However, it is possible to find a compromise between both. Assume that a degradation of 20% in the servo response *can* be tolerated. The tool can help to answer which is the best disturbance rejection response *can*

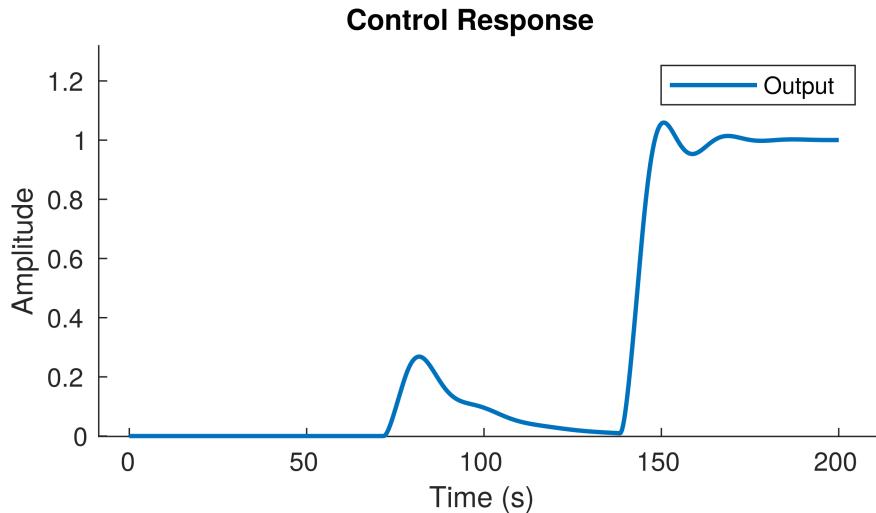


Fig. 7.14: Closed-loop response with a 200 s time window.

Case	J_{di}	J_r	M_s
Original Setting	5.77	10.56	2.05
J_r with allowed 20% degradation	3.81 (-34.0%)	11.10 (5.11%)	2.44

Table 7.13: Comparison between the original setting for the closed-loop response and the degraded J_r tuning.

that can be achieved with this constraint. After trying different values, it was found that the best disturbance rejection response that can be achieved that, at the same time, has a servo response with a degradation no greater than 20% from its optimal value, has an allowed degradation of 32.6% for J_{di} .

In Table 7.13 the comparison between the original and the final settings are presented. As it can be seen, the tool let the user to find an optimal controller that complies with the maximum allowed degradation for both functions at the same time. In fact, it was possible to find a degradation that reduces the J_{di} IAE by 34% while worsening the J_r only by 5.11%. However, it has to be noticed that the robustness value was also increased, which means that the new tuning is less robust than the original. J_{di}

If it is necessary to maintain the same degree of robustness, the tool allows the user to set the robustness for three different levels (2.0, 1.8 and 1.6). With the M_s constraint set to 2.0, the same exercise was performed: the best J_{di} cost function was found while keeping the J_r allowed degradation under 20% and the M_s value near 2.0. The results are given in Table 7.14. With the extra constraint of keeping the same robustness value, it is necessary to allow a 48.1% degradation on J_{di} . It is not possible to find a better response, because it would yield outside the feasible region.

Case	J_{di}	J_r	M_s
Original Setting	5.65	10.62	2.00
J_r with allowed 20% degradation	4.68 (-17.17%)	10.99 (3.37%)	2.01

Table 7.14: Comparison between the original setting for the closed-loop response and the degraded J_r tuning with an extra constraint of $M_s = 2.0$.

Also, it is possible to improve the J_{di} response by 17.17% (compared with the 34% obtained above).

This example shows the usefulness of the tool to explore the Pareto, without the need to plot the Fronts and without optimizing every time a change is made in the allowed degradation parameters. Of course, this is possible thanks to the fact that all the Paretos were obtained beforehand and that they are available as a series of files included in the tool.

References

- Alfaro VM, Vilanova R (2012) Optimal Robust Tuning for 1DoF PI/PID Control Unifying FOPDT/SOPDT Models. IFAC Proceedings Volumes 45(3):572–577, DOI <http://dx.doi.org/10.3182/20120328-3-IT-3014.00097>, URL <http://www.sciencedirect.com/science/article/pii/S1474667016310874>
- Alfaro VM, Vilanova R (2016) Model-Reference Robust Tuning of PID Controllers. Advances in Industrial Control, Springer International Publishing, Cham, DOI 10.1007/978-3-319-28213-8, URL <http://link.springer.com/10.1007/978-3-319-28213-8>
- Cao Y (2020) Pareto Set. URL <https://www.mathworks.com/matlabcentral/fileexchange/15181-pareto-set>
- Contreras Leiva M, Rojas JD (2015) New Tuning Method for PI Controllers based on Pareto-Optimal Criterion with Robustness Constraint. IEEE Latin America Transactions 13(2):434–440, DOI 10.1109/TLA.2015.7055561
- Houska B, Ferreau HJ, Diehl M (2011) ACADO toolkit-An open-source framework for automatic control and dynamic optimization. Optimal Control Applications and Methods 32(3):298–312, DOI 10.1002/oca.939, URL <http://doi.wiley.com/10.1002/oca.939>
- Logist F, Vallerio M, Houska B, Diehl M, Van Impe J (2012) Multi-objective optimal control of chemical processes using ACADO toolkit. Computers & Chemical Engineering 37(0):191–199, DOI 10.1016/j.compchemeng.2011.11.002, URL <http://www.sciencedirect.com/science/article/pii/S0098135411003231> <https://linkinghub.elsevier.com/retrieve/pii/S0098135411003231>
- Moya F, Rojas JD, Arrieta O (2017) Pareto-based polynomial tuning rule for 2DoF PID controllers for time-delayed dominant processes with robustness consideration. In: 2017 American Control Conference (ACC), IEEE, Seattle, pp 5282–5287, DOI 10.23919/ACC.2017.7963775, URL <http://ieeexplore.ieee.org/document/7963775/>
- Murril PW (1967) Automatic control of processes. International Textbook Co.
- O'Dwyer A (2000) A summary of PI and PID controller tuning rules for processes with time delay. Part 1: PI controller tuning rules. In: Proceedings of PID'00: IFAC Workshop on Digital Control, Terrassa, Spain, pp 175–180
- O'Dwyer A (2009) Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London, UK

- Rojas JD, Valverde-Mendez D, Alfaro VM, Arrieta O, Vilanova R (2015) Comparison of multi-objective optimization methods for PI controllers tuning. In: 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), IEEE, pp 1–8, DOI 10.1109/ETFA.2015.7301410, URL <http://ieeexplore.ieee.org/document/7301410/>
- Skogestad S (2003) Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13(4):291–309, DOI 10.1016/S0959-1524(02)00062-8, URL <https://www.sciencedirect.com/science/article/pii/S0959152402000628> <https://linkinghub.elsevier.com/retrieve/pii/S0959152402000628>

Chapter 8

Industrial application examples

Abstract In this chapter, different examples are provided to apply the tool presented in this book. The MOOTuning software is used to analyze the temperature control in a Continuous Stirred Tank Heater (CSTH) in Section 8.1 using two a three cost functions. Also a Continuos Stirred Tank Reactor (CSTR) is considered for the control of the concentration of the product in an isothermal case in Section 8.2

8.1 Continuously Stirred Tank Heater

8.1.1 Description of the process

The control of a continuously stirred tank heater (CSTH) is a common task in industrial processes. In this section, the control of the temperature of the CSTH will be solved as a multiobjective optimization problem (MOOP) using a two degrees of freedom (2DoF) proportional integral derivative (PID) controller. The diagram of the process is presented in Figure 8.1. A heat exchanger is installed inside the tank to heat the fluid. The flow rate inside the heat exchanger is controlled with a valve with input variable U_T and the liquid inside the heat exchanger enters with temperature T_{ci} and leaves with temperature T_{co} , the average temperature inside the heat exchanger is T_{ca} . The volume inside the tank is variable, the input flow rate is Q_i with temperature T_i . The output flow rate is Q with temperature T . The output flow rate is controlled with a valve with input variable U_L . The tank is covered with a jacket that prevents any heat loss to the atmosphere.

inside

According to Alfaro and Vilanova (2016), a possible model for this process is given by the following set of algebraic-differential equations:

- Tank mass balance:

$$A \frac{dH(t)}{dt} = Q_i(t) - Q(t),$$

where A is the transversal area of the tank and $H(t)$ is the liquid level.

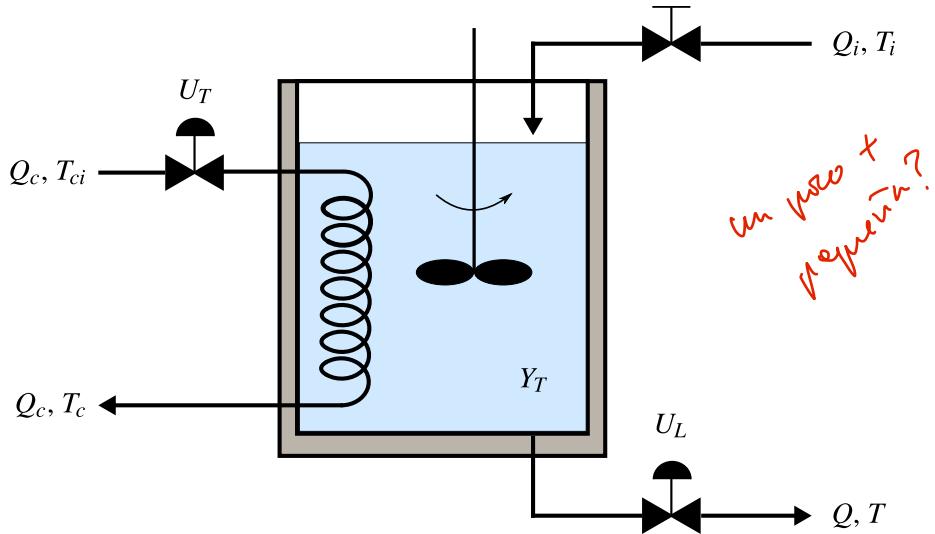


Fig. 8.1: Simplified diagram of a continuously stirred tank heater to be controlled.

- Tank energy balance:

$$\rho C_p A H(t) \frac{dT(t)}{dt} = \rho C_p Q_i(t) (T_i(t) - T(t)) + W(t),$$

where C_p is the heat capacity of the fluid and $W(t)$ is the rate of heat transfer from the heat exchanger to the tank.

- Heat exchanger energy balance:

$$\rho_c C_{pc} V_c \frac{T_{ca}(t)}{dt} = \rho_c C_{pc} Q_c(t) (T_{ci}(t) - T_{co}(t)) - W(t),$$

where ρ_c is the density of the fluid inside the heat exchanger, C_{pc} is the heat capacity of the fluid inside the heat exchanger and V_c is the volume of the heat exchanger.

- Heat transfer between the heat exchanger and the fluid in the tank:

$$W(t) = UA_c (T_{ca}(t) - T(t)),$$

where U is overall heat-transfer coefficient, A_c is the area of the heat exchanger, $T_{ca}(t)$ is the average temperature inside the heat exchanger which is related to $T_{co}(t)$ and $T_{ci}(t)$ as:

$$T_{ca}(t) = \frac{T_{ci}(t) + T_{co}(t)}{2}$$

Also, in Alfaro and Vilanova (2016), the transmitters and the valves are modeled as:

- Level transmitter: it is supposed that the level transmitter is a capacitive type electronic transmitter that has a first order dynamics:

$$T_L \frac{dY_L(t)}{dt} + Y_L(t) = K_L H(t),$$

where T_L is its time constant, Y_L is the level signal and K_L is the transmitter gain.

- Temperature transmitter: It is supposed that a Pt₁₀₀ RTD electronic sensor is installed in a thermowell at the tank outlet pipe. It is supposed that it has a second order dynamic:

$$T_T^2 \frac{d^2 Y_T(t)}{dt^2} + 2T_T \frac{dY_T(t)}{dt} + Y_T(t) = K_T T(t),$$

where T_T is its time constant and K_T is its gain.

- Level control valve: it is supposed that a ball valve with an electroneumatic actuator is used. The valve inherent flow characteristics is nearly quadratic and the relationship between the flow $Q(t)$ and the input variable U_L is given by:

$$\begin{aligned} T_{vL} \frac{dX_L(t)}{dt} + X_L(t) &= K_{xL} U_L(t), \\ Q(t) &= K_{vL} X_L^2(t) \sqrt{\rho g H(t)}, \end{aligned}$$

where T_{vL} is the level control valve time constant, K_{xL} level control valve stem constant K_{vL} level control valve constant and $X_L(t)$ is the level control valve stem normalized travel.

- Temperature control valve it is also supposed to be a ball valve with an electroneumatic actuator, however, it is supposed that the valve has an equal-percentage inherent flow characteristics given by:

$$\begin{aligned} T_{vT} \frac{dX_T(t)}{dt} + X_T(t) &= K_{xT} U_T(t), \\ Q_c(t) &= K_{vT} R_{vT}^{(X_T(t)-1)} \sqrt{P_{cp} - (R_c Q_c^2(t) + P_{cr})}, \end{aligned}$$

where T_{vT} is the temperature control valve time constant, K_{xT} the temperature control valve stem constant, K_{vT} is the temperature control valve constant P_{cp} is the heating fluid pump discharge pressure, P_{cr} is heating fluid system return pressure and $X_T(t)$ is the temperature control valve stem normalized travel.

Taking this model in consideration, it can be said that, from the point of view of the controller, the controlled variables are give by the signals $Y_L(t)$ (which represents the level) and $Y_T(t)$ (which represents the temperature of the fluid of the tank). The manipulated variables are given by $U_L(t)$ (which directly affects Q) and $U_T(t)$ (that directly affects Q_c). $Q_i(t)$, T_i and T_{ci} are considered as disturbances. The state variables of the system are given by $H(t)$, $T_T(t)$, T_{co} , $Y_L(t)$, Y_T , $X_L(t)$ and $X_T(t)$, therefore, this model comprises a seventh order non-linear system for a two-input two-output industrial process. The parameters of the model can be found in

Table 8.1. This model was implemented in Simulink and can be found with the

Table 8.1: Parameters for the CSTM process

Symbol	Value	Description
<i>Tank parameters</i>		
ρ	1200 kg m^{-3}	tank fluid density
A	0.0707 m^2	tank inside section area
C_p	$4190 \text{ J kg}^{-1} \text{ }^\circ\text{C}^{-1}$	tank fluid heat capacity
g	9.8 m s^{-2}	gravity acceleration
K_T	$2 \% / ^\circ\text{C}$	temperature transmitter gain
K_{vL}	1.25×10^{-5}	level control valve constant
K_{vT}	3×10^{-6}	temperature control valve constant
K_{xL}	$0.01 \% /$	level control valve stem constant
Q_i	$7 \times 10^{-4} \text{ m}^3 \text{ s}^{-1}$	normal tank inlet fluid flow rate
T_i	$24 ^\circ\text{C}$	fluid inlet temperature
T_L	2 s	level transmitter time constant
T_T	15 s	temperature transmitter time constant
T_{vL}	3 s	level control valve time constant
T_{vT}	5 s	temperature control valve time constant
<i>Heat exchanger parameters</i>		
ρ_c	800 kg m^{-3}	heating fluid density
A_c	0.6362 m^2	heat exchanger transfer area
C_{pc}	$2400 \text{ J kg}^{-1} \text{ }^\circ\text{C}^{-1}$	heating fluid heat capacity
K_L	$125 \% / \text{m}$	level transmitter gain
K_{xT}	$0.01 \% /$	temperature control valve stem constant
P_{cp}	$4.14 \times 10^5 \text{ Pa}$	heating fluid pump discharge pressure
P_{cr}	$1.38 \times 10^5 \text{ Pa}$	heating fluid system return pressure
R_c	$5.5 \times 10^{10} \text{ Pa}/(\text{m}^3/\text{s})^2$	heating system pipe nominal flow resistance
R_{vT}	50	temperature control valve rangeability
T_{ci}	$320 ^\circ\text{C}$	heating fluid inlet temperature
U	$440 \text{ J s}^{-1} \text{ m}^{-2} \text{ }^\circ\text{C}^{-1}$	overall heat-transfer coefficient
V_c	0.0139 m^3	heat exchanger volume

companion software. In Figure 8.2. Each equation of the model was implemented in a subsystem for clarity. For example in Figure 8.3 the Simulink implementation of the heat exchanger energy balance is presented. The result of this submodel is the computation of the state variable T_{ca} , which represents the average temperature of the heating fluid. As it can be seen, the parameters of the model are not hard-coded in the Simulink blocks, instead a parameter initialization script is called before the simulation starts. If the user desires to change any value of the parameters, it can be done globally in the script and then automatically called during the simulation.

Called

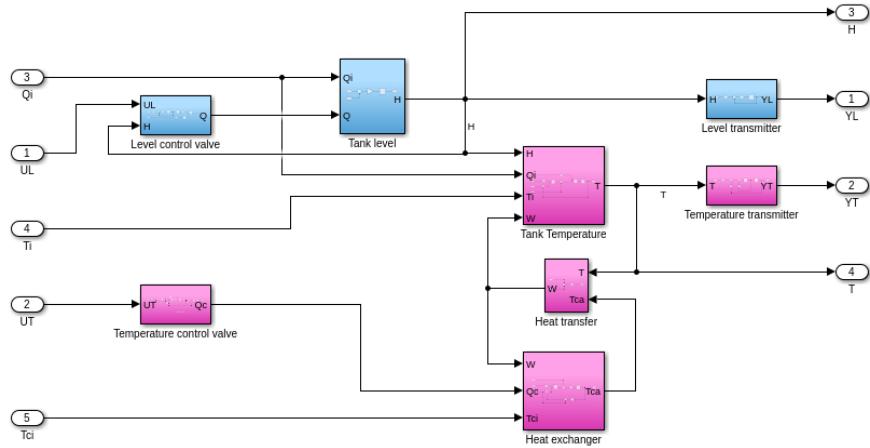


Fig. 8.2: Simulink implementation of the model of the heater.

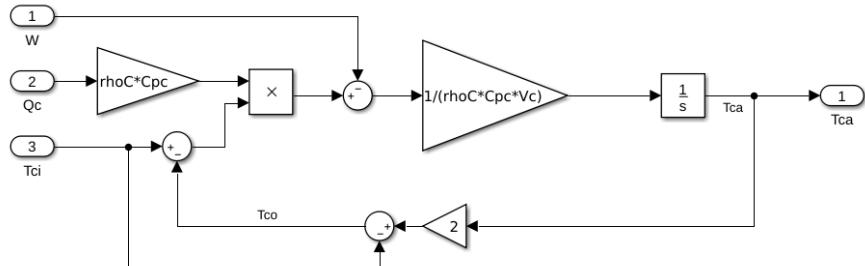


Fig. 8.3: Example of the implementation of the heat exchanger energy balance.

8.1.2 Simplified linear model

In order to find a PID controller using the MOOTuning app, it is necessary to find a linear model of the plant in the operation point. An identification procedure was performed with a change of 10% in the value of U_T to find the transfer function between Y_T and U_T . The response to this change is depicted in Figure 8.4. As it can be seen, the response is overdamped and takes approximately 500 s to reach a new steady state. A change in 10% on the input signal produces a variation of approximately 3.5% in the output signal. It has to be noticed that the presented signals are normalized between 0 and 100% representing the full span of the transmitter and actuators.

In order to find the model, the process was supposed to have two poles, no zeros and a pure time-delay (also known as dead-time). Of course, if a linearization procedure were performed using the nonlinear model, a seventh order model would be

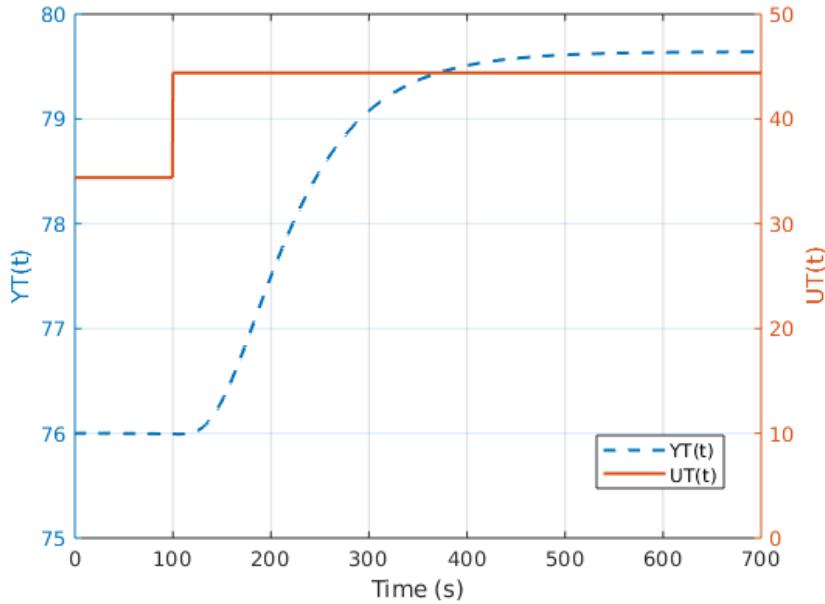


Fig. 8.4: Response of the process to a change of 10% in the $U_T(t)$ input.

obtained. However, for PID tuning, a first or second order model is usually expected to tune the controller.

Considering the experiment performed with the data as depicted in Figure 8.4, the resulting simplified model is given by:

$$\frac{Y_T(s)}{U_T(s)} = \frac{0.3658e^{-24.736s}}{(52.861s+1)(52.805s+1)}. \quad (8.1)$$

From this transfer function, it can be deduced that the gain is equal to $K = 0.3658$, the main time constant is given by $T = 52.861$, the ratio between the two time constant is given by $a = 0.9989$ and the dead-time is given by $L = 24.736$, therefore the normalized dead-time is given by $\tau = 0.4679$.

To test the validity of this simplified model, the response of the transfer function is compared against the response of the non-linear model. It was found that the transfer function response is very similar to the response of the non-linear model, as can be seen in Figure 8.5. It is clear that the non-linear model is a good representation of the dynamical response of the process. This is the first step in order to find a suitable PID controller to control the plant. In Alfaro and Vilanova (2016) the level of the tank is also controlled, however the dynamic of the level is simpler (its model can be approximated with a first order model without delay) and in this

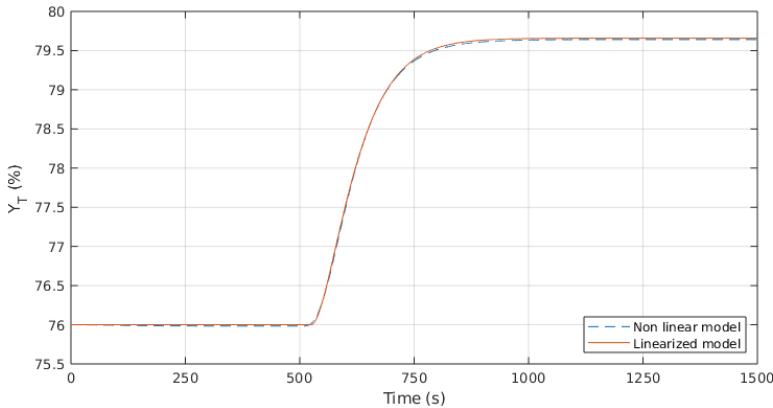


Fig. 8.5: Comparison between the linear and no linear models for the CSTH.

particular example, only the temperature is going to be controlled, while the level is considered to be constant.

8.1.3 PID control of the CSTH considering two integral cost functions

In this section, the process will be controlled using a PID controller with different tuning methods and compared with the multiobjective optimization (MOO) framework used in the book.

Two different tuning rules were considered: the method by Rovira et al (1969) and the method by Murril (1967). For the case of the Rovira and Murril method, the model in (8.1) was reduced to a first order model using the Half-Rule in Skogestad (2003):

$$\frac{Y_T(s)}{U_T(s)} = \frac{0.3658e^{-24.7360s}}{79.2635s + 1}. \quad (8.2)$$

The equations were implemented as presented in O'Dwyer (2009):

- Supposing a first order plus time delay (FOPTD) model given by:

$$P(s) = \frac{Ke^{-Ls}}{Ts + 1}$$

The Murrill tuning is given by:

Table 8.2: Comparison of different PID tunings for the CSTH process.

Tuning	K_p	T_i	T_d	β	J_r	J_{di}
Murril	5.87	65.02	23.21	1	82.51	15.34
Rovira	4.34	120.81	18.48	1	76.00	27.79
MOO01	11.3	58.82	29.48	0.43	85.65	7.06
MOO02	9.26	123.85	26.68	0.80	69.96	13.38
MOO03	8.20	185.21	28.14	0.99	67.93	22.54

$$K_p = \frac{1.435}{K} \left(\frac{T}{L} \right)^{0.921}$$

$$T_i = \frac{T}{0.878} \left(\frac{L}{T} \right)^{0.749}$$

$$T_d = 0.482T \left(\frac{L}{T} \right)^{1.137}$$

- Again, supposing a FOPTD as above, the Rovira tuning is given by:

$$K_p = \frac{1.086}{K} \left(\frac{T}{L} \right)^{0.869}$$

$$T_i = \frac{T}{0.740 - 0.13 \frac{L}{T}}$$

$$T_d = 0.384T \left(\frac{L}{T} \right)^{0.914}$$

The values of the computed values can be found on Table 8.2, along with its associated values of J_{di} and J_r . The Murril and Rovira methods presented in the table were selected because they are intended to minimize the integral of the absolute value of the error (IAE). In all cases, the PID tuning is for a one degree of freedom controller (that is the reason why β is equal to one).

The PID that can be found using the data and the framework presented in Chapter 7 is an overdamped second order plus time delay (ODSOPTD), which may do the comparison somehow unfair. However, the idea now is to compare methods that tries to minimize the IAE. Using the MOOTuning Tool that accompanies this book, the Pareto front that was found is given as in Figure 8.6. As it was expected, all the controllers found using the MOO tool present lower values for J_{di} and J_r . If all controllers had the same topology, most certainly both controller would be close to the anchor points. However, what it is important here is the fact that, using the tool, the user has the ability to chose between practically an infinity of possible controllers. From the Pareto front, three different tunings were selected in order to compare the

from
an

process model

S

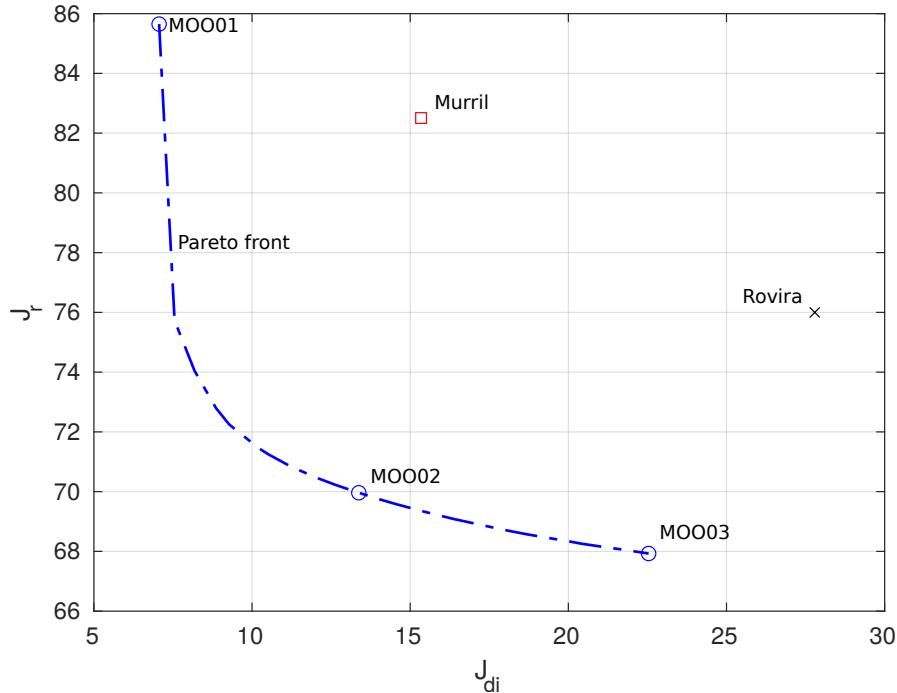


Fig. 8.6: MOOTuning compared to the Murril and Rovira methods that also minimizes IAE.

responses using the non-linear plant. The values of the parameters are presented also in Table 8.2 and depicted as circles in the Pareto front in Figure 8.6.

In Figures 8.7 and 8.8 the responses to a step change in the setpoint and in the disturbance are presented. The corresponding values of IAE are also presented in the graph. In all cases, the robustness was not considered as a constraint, but it is possible to include it within the MOOTuning software. From Figure 8.6 given the steep slope of the curve for lower values of J_{di} that a small change in J_{di} may improve substantially the performance for J_r . Therefore, one may be more prone to select a controller that may have a little degradation in J_{di} and for this reason, controller MOO01 may not be a good selection as a final solution unless having the minimum value possible of J_{di} is the final goal.

The controller MOO02 may be seen as an intermediate solution between MOO1 and MOO03 in case both J_{di} and J_r are equally important for the decision maker. The power of the multiobjective framework is evident, and giving that the computational power is done offline, it becomes a good tool for the tuning of PIDs in an industrial setting.

To check how the tuning performs with the nonlinear model, a simulation was performed using the tuning of the controller MOO02. The setpoint was increased by 5% at $t = 100$ s and the temperature of the steam was increased by 10°C at $t = 700$ s.

Survive?

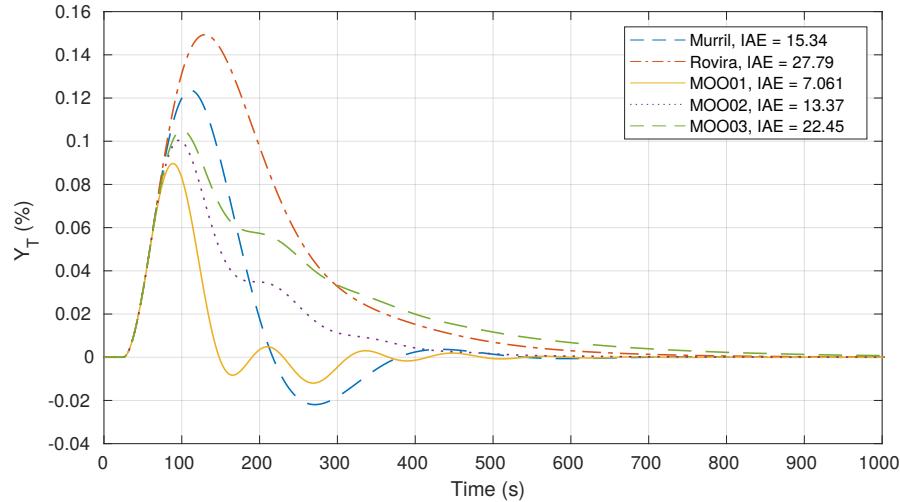


Fig. 8.7: Regulator response comparison for minimum IAE.

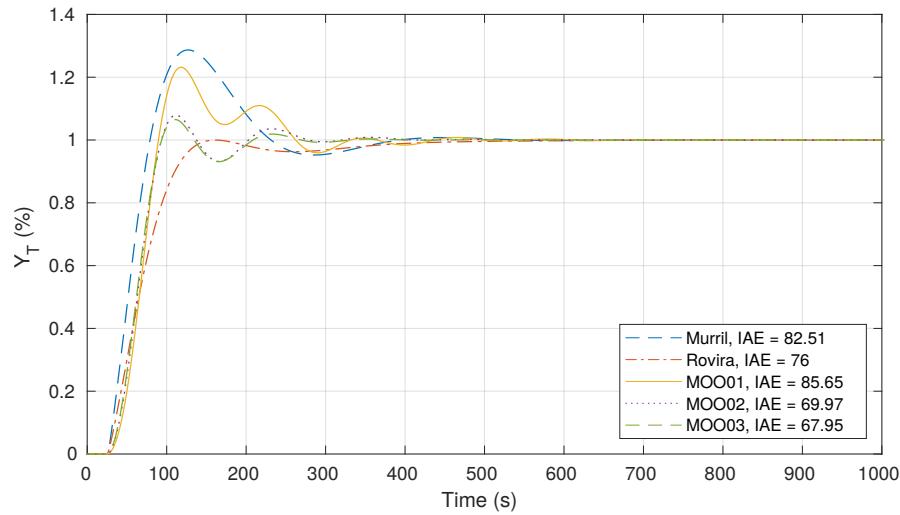


Fig. 8.8: Servo response comparison for minimum IAE.

The response is presented in Figure 8.9. As it can be seen, the servo response is very close to the one presented in Figure 8.8, which is a clear indicator that the linear model was a good approximation of the plant at the given operation point. The response to the change in the stem temperature cannot be compared with the regulation presented in Figure 8.7, because the disturbance was not applied directly at the input of the plant. However, it is interesting to note that the controller was able to respond with a good dynamic even though it was not optimized for this case.

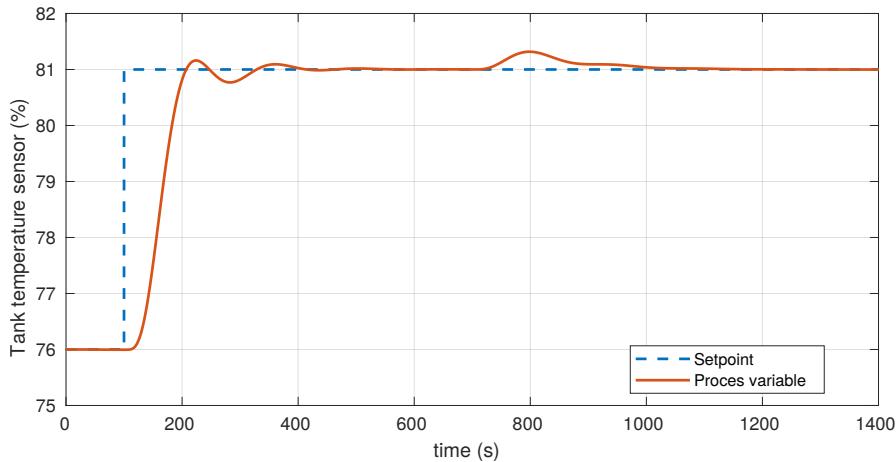


Fig. 8.9: Response of the controlled system using the nonlinear model for the CSTH.

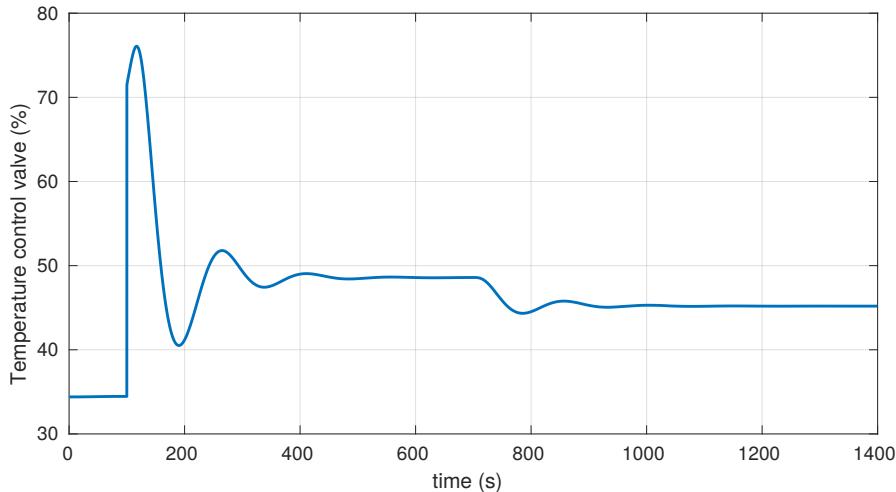


Fig. 8.10: Response of the controlled system using the nonlinear model for the CSTH.

The controlled variable is presented in Figure 8.10. When the setpoint changes, the response of the controller is abrupt (more than double its original value), but then the value rapidly reaches the new setpoint. The change produced by the disturbance has a milder response, and in less than 200 s reaches again a new steady state.

The case presented here tried to show the steps to use the Pareto front as the methodology to find the controller tuning more appropriate to the task. The example is a simple plant, but very representative of the dynamics that can be found in an industry environment. Many of the plants can be modeled as a second order over-

was intended *200s*

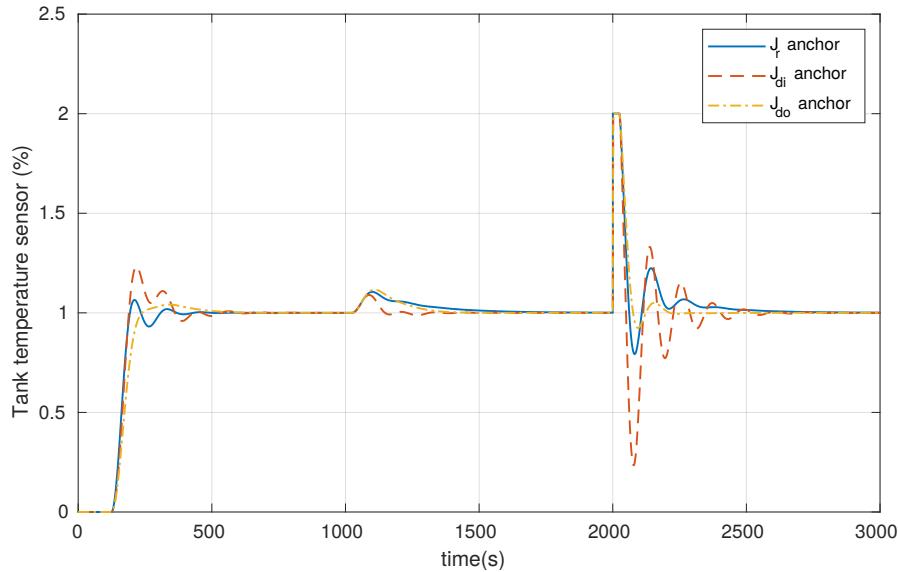


Fig. 8.11: Response of the CSTH process in the three anchor points of the pareto front for $M_s \leq 2.0$.

damped process, and therefore, the tool and the data used in this book are readily applied in many cases.

8.1.4 PID control of the CSTH considering three integral cost functions

is also able

The MOOTuning software ~~also~~ is able to find the optimal parameters of a PID controllers considering three cost functions as presented in Section 7.2.2. Of course, it is not necessary to use this software since the data base with all the values is also part of the companion software, but the MATLAB app has the advantage to be a simple interface between the user and the data.

As an example, the tuning tool is used to find the parameters of the controller that has the lowest J_{do} value. This case is interesting because the anchor point where J_{do} has the lowest value, neither the value of J_r nor J_{di} have their maximum value. On the other hand, when J_r is set to be the lowest possible value, J_{do} becomes the function that has an intermediate value but J_{di} has its maximum value from the Pareto. Only for the anchor point where J_{di} is minimum, both J_r and J_{do} get their maximum value. The responses for the three anchor points are depicted in Figure 8.11 for the case where $M_s \leq 2.0$. As it can be seen, the response is quite different among the three anchor points. First a change in the reference value is

performed at $t = 100$ s, an input step disturbance is present at $t = 1000$ s and an output step disturbance is introduced in the system at $t = 2000$ s. The values of the cost functions are presented in Table 8.3. The percentage increment is reported for

Table 8.3: Cost functions for the three cost functions case scenario.

	J_r anchor point	J_{di} anchor point	J_{do} anchor point
J_r	67.95 (minimal)	85.63 (+26.02%)	82.46(+21.35%)
J_{di}	22.46 (+218.13%)	7.06 (minimal)	17.63 (+149.72%)
J_{do}	74.25 (+38.01%)	102.43 (+90.39%)	53.80 (minimal)
Parameters	$K_p = 8.19$ $T_i = 185.18$ $T_d = 28.14$ $\beta = 0.99$	$K_p = 11.29$ $T_i = 58.8306$ $T_d = 29.48$ $\beta = 0.43$	$K_p = 6.18$ $T_i = 108.76$ $T_d = 29.93$ $\beta = 0.82$

each cost function in each case.

Using Figure 8.11 and Table 8.3, it can be confirmed that the tuning with the best parameters for J_{di} produces the worst responses for the other two functions. With these results one may be prone to select the tuning for the lowest value of J_{di} as the final because it does not have the worst values of the other cost functions and it has much better response for the output disturbance case. Observe for example the response for the J_{di} anchor point where the response for the output disturbance is bad (it has a J_{do} increment of +90.39% with respect to its lowest value). In some sense, it could be seen as the best compromise between the cost functions. Only if the control engineer is heavily invested in minimize the J_{di} cost function, it may select the J_{di} anchor point as the final response, however, using the multiobjective framework presented here, he or she has to be fully aware that is selecting the worst response for the other functions.

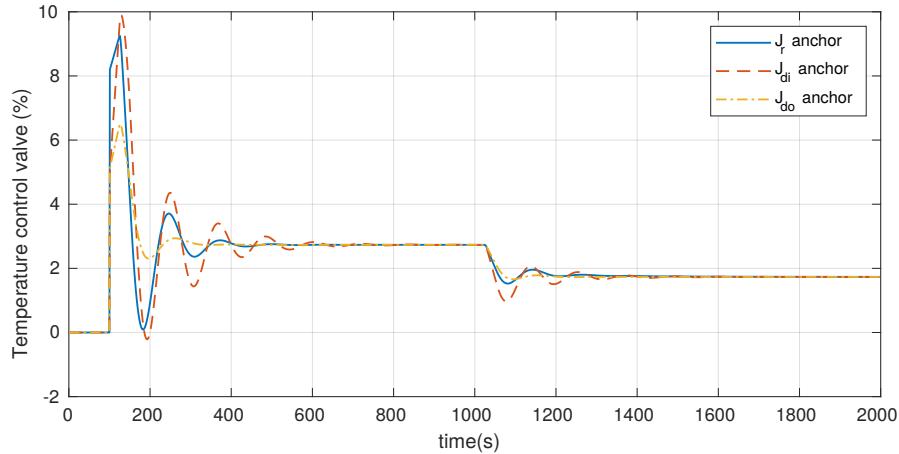
Another reason to select the J_{do} anchor point is related to the control effort. When the Total Variation computed as:

$$(TV) \quad TV = \sum_{i=0}^{N-1} (u(i+1) - u(i)),$$

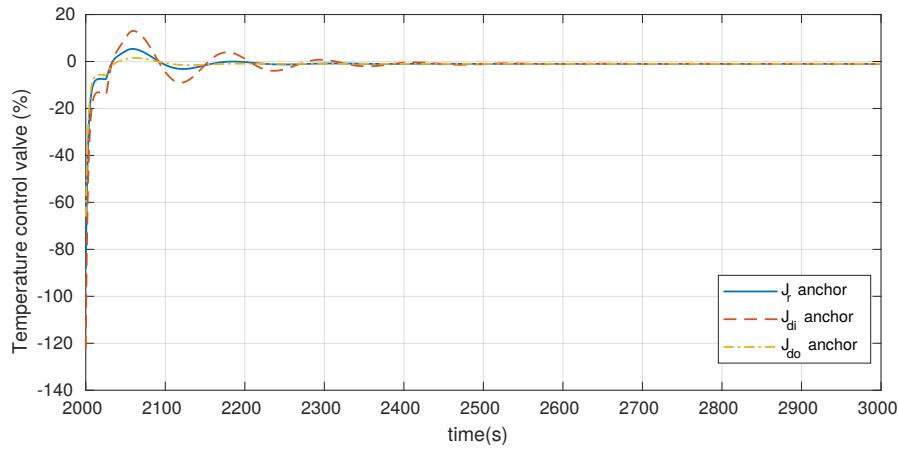
is compared between the three responses, it is found that $TV_{J_r} = 223.82$, $TV_{J_{di}} = 352.94$ and $TV_{J_{do}} = 152.81$ using a step size of 0.01s. It is clear that the response given by the J_{do} anchor point is a very good choice among all the possible values. The control signal is plotted on Figure 8.12. Since the control signal has a larger magnitude for an output disturbance rejection, it was plotted in different axis.

As it can be seen from the response, effectively the response of the J_{do} anchor point is smoother than the other two. Even if the control engineer is not looking for the best response to the output disturbance, the obtained tuning may be a good compromise between servo and regulation responses with a mild control signal.

In section 8.1.3, the system was controlled considering only two sources of disturbances. When adding another dimension to the problem, certainly the selection



(a) Reference and input disturbance step changes.



(b) Output disturbance step change.

Fig. 8.12: Control signal for the CSTH using three cost functions.

of the final controller may be more difficult because another degree of freedom is added. However, the insight that was gathered from rethinking the problem from this other point of view can be seen as beneficial, because the tuning found using the 3 dimensional Pareto could be a better solution (from a physical point of view) that may not be part of the front with only two functions.

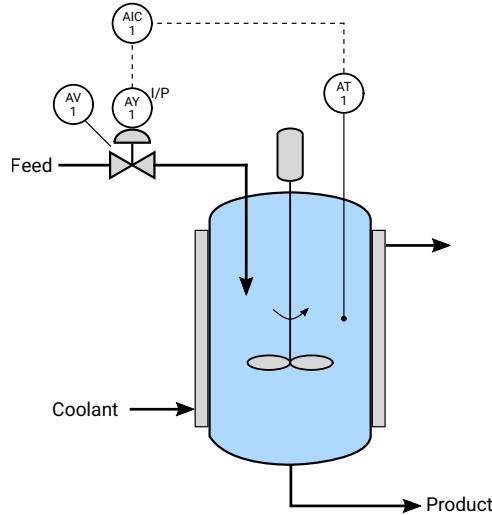


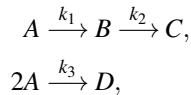
Fig. 8.13: CSTR process using the Van de Vusse reaction model.

8.2 Continuously Stirred Tank Reactor

8.2.1 Description of the process

The continuously stirred tank reactor (CSTR) with the Van de Vusse reaction (van de Vusse, 1964) is a common benchmark plant for testing control algorithms given its different dynamics depending on the operating point and is depicted in Figure 8.13. For this particular case, the isothermal process is considered and both the concentration sensor as the valve actuator will be modeled as well. The objective is to control the feed flow to obtain the desired concentration of a product.

The Van de Vusse reaction models a process where desired product B is obtained from A, but at the same time, both A and B are degraded to D and C respectively. The chemical equation that represent this reaction is given by:



where k_i are the rate constants of the formation rates of A and product B as given by (van de Vusse, 1964):

$$\begin{aligned} r_A &= -k_1A - k_3A^2, \\ r_B &= k_1A - k_2B. \end{aligned} \tag{8.3}$$

When performing a mass balance, the model becomes (Arrieta et al, 2010):

Table 8.4: Parameters values for the CSTR model.

$$\begin{aligned} k_1 &= 0.833 \text{ min}^{-1} & k_2 &= 1.667 \text{ min}^{-1} \\ k_3 &= 0.167 \text{ min}^{-1} & C_{Ai} &= 10 \text{ mol L}^{-1} \\ V &= 700 \text{ L} \end{aligned}$$

$$\begin{aligned} \frac{dC_A(t)}{dt} &= \frac{F_r(t)}{V} (C_{Ai} - C_A(t)) - k_1 C_A(t) - k_3 C_A^2(t) \\ \frac{dC_B(t)}{dt} &= -\frac{F_r(t)}{V} C_B(t) + k_1 C_A(t) - k_2 C_B(t) \end{aligned} \quad (8.4)$$

where C_A and C_B represents the reactants concentrations in mol L^{-1} , C_{Ai} is the concentration of A in the feed flow in mol L^{-1} , F_r is the input flow in L min^{-1} , and V is the volume of the CSTR in L. The nominal values of the parameters are presented in

The range of the sensor for the product is supposed to be in the range 0 to $1.5714 \text{ mol L}^{-1}$ and the maximum flow that is allowed by the valve is given by $634.1719 \text{ L min}^{-1}$. Given these values, the model for the sensor-transmitter is given by:

$$y(t) = \left(\frac{100}{1.5714} \right) C_B(t), \quad (8.5)$$

while the transmitter is modeled as:

$$F_r(t) = \left(\frac{634.1719}{100} \right) u(t), \quad (8.6)$$

where $u(t)$ and $y(t)$ are the normalized input and output signal, respectively. The operation point of the plant is given by $u_0 = 60\%$ and $y_0 = 70\%$ which represents concentration of $C_{A0} = 2.9175 \text{ mol L}^{-1}$ and $C_{B0} = 1.10 \text{ mol L}^{-1}$ with the input concentration given by $C_{Ai0} = 10 \text{ mol L}^{-1}$.

These parameters give the system a wide range of operation. In Figure 8.14 it can be seen that with the value of C_{Ai0} given and the input at 60%, the output yield 70%. In the figure the dashed lines represents the variation due to $\pm 10\%$ variation of C_{Ai} . It can be seen that, with a 100% value for u and a variation of 10% of C_{Ai} , the output is less than 100% of the capacity of the sensor.

C_A *C_B*
The concentration of C_a and C_b are both plotted in Figure 8.15. The C_a concentration is on the horizontal axis while the C_b concentration is on the vertical axis. The selected operation point is represented with a circle in the curve and also the curves with the variation on the value of C_{Ai} are also presented. As it can be seen, in the operation point selected, the curves start to diverge, which means that the model has a larger dependency on the variation of the input concentration. This has to be taken into account when designing the feedback controller.

When considering the transient response to a change in the input flow u , the response is as given in Figure 8.16. It is interesting to note that the system has an inverse response. This characteristic limits the possible performance of the con-

i u? i flow?

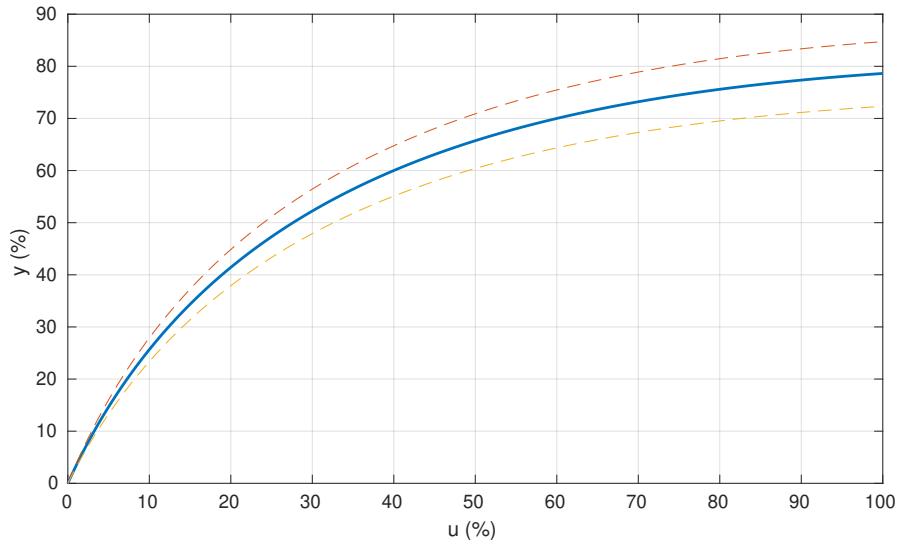


Fig. 8.14: Operation points of the CSTR process.

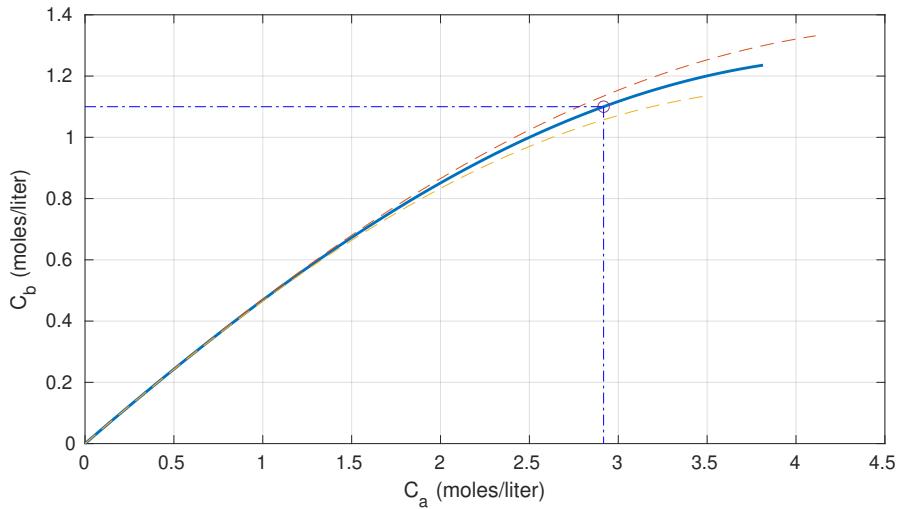


Fig. 8.15: Concentration of the reactants for all possible operating points.

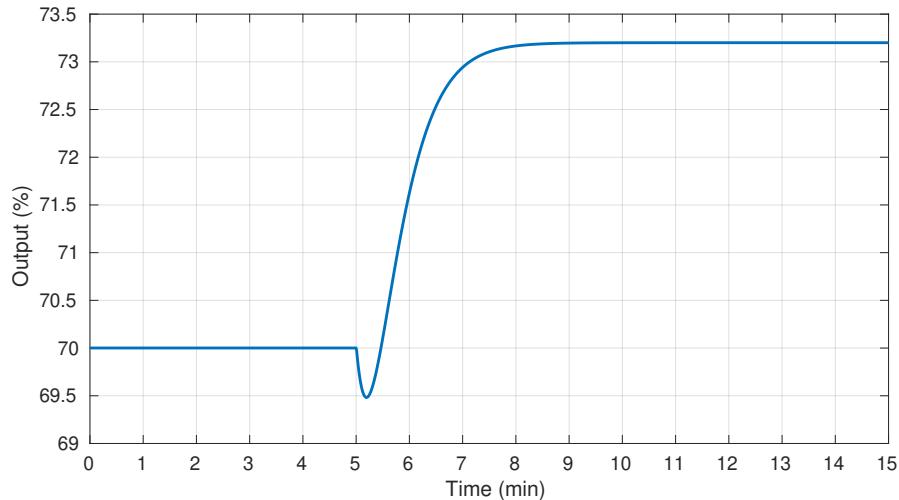


Fig. 8.16: Open loop response to a step change in the input signal of the CSTR.

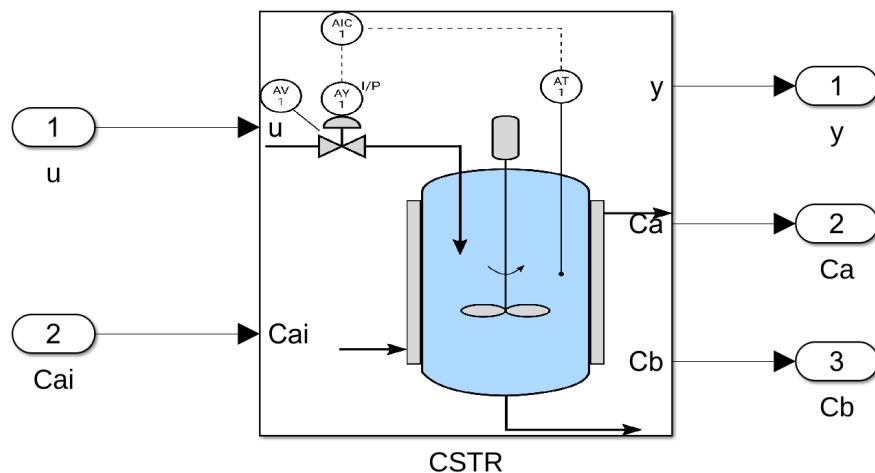


Fig. 8.17: CSTR implemented as an S-function in Simulink .

trolled loop, making unfeasible to increase the gain of the controller to achieve a faster response without instability. This is also another point that has to be considered when designing the controller.

This model was implemented as an S-Function in MATLAB / Simulink and can be found in the companion software. In Figure 8.17, the basic block with the model is presented. It has u and C_{ai} as inputs and y , C_a and C_b as outputs. With the intention to facilitate the characterization of the model, a mask was designed to enter the parameters and the initial value of the states. This mask can be found in Figure 8.18.

*en el modelo aparecen los
parámetros*

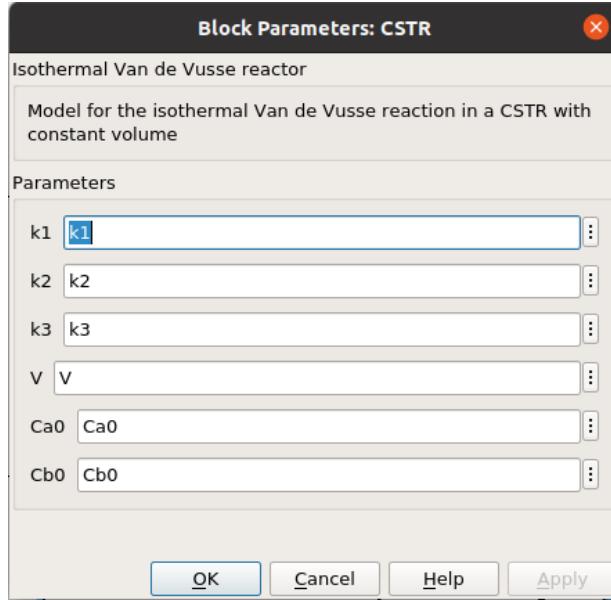


Fig. 8.18: Simulink mask to enter parameters and initial values for the CSTR system.

All the values of the parameters can be set on the MATLAB workspace and use directly on the mask. This is an advantage in case the user desires to use another set of parameters or compare the response of the system with different values.

The objective of this example is to control the reactor using the multiobjective approach. In order to use the framework and the MATLAB app, it is necessary to find a suitable linear second order model. Next, the procedure to find this model is presented.

8.2.2 Linearization

In order to use the MOOTuning software, it is necessary to have a linear model of the process. In this case, the linearization is done by taking the first order approximation of the model in (8.4) near the operation point.

When defining the incremental variables $\delta\mathbf{x}$, $\delta\mathbf{u}$ and δy as:

$$\delta\mathbf{x} = \begin{bmatrix} C_A - C_{A0} \\ C_B - C_{B0} \end{bmatrix},$$

$$\delta\mathbf{u} = \begin{bmatrix} u - u_0 \\ C_{Ai} - C_{Ai0} \end{bmatrix},$$

$$\delta y = y - y_0.$$

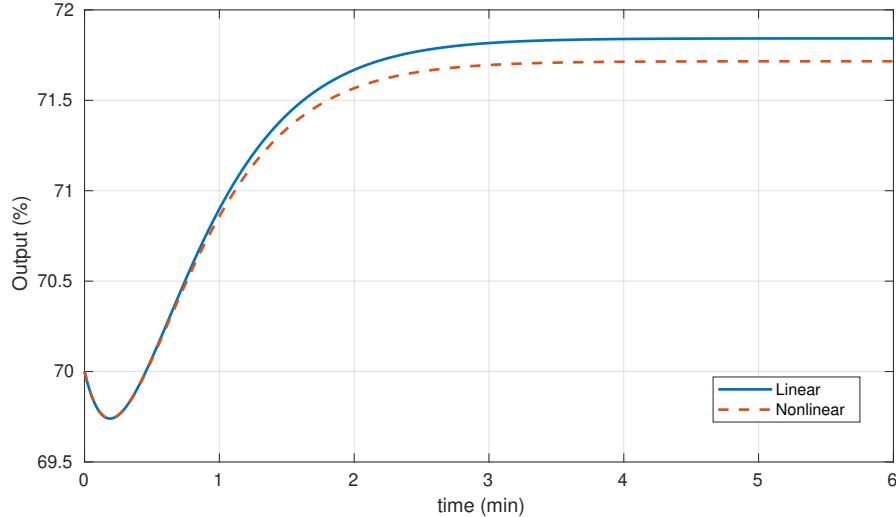


Fig. 8.19: Comparison of the nonlinear and linear model for the CSTR for a 5% change in the input.

The process dynamics can be approximated by the first order model given by:

$$\delta \dot{\mathbf{x}} = \mathbf{A} \delta \mathbf{x} + \mathbf{B} \delta \mathbf{u}, \quad (8.7)$$

$$\delta y = \mathbf{C} \delta \mathbf{x}, \quad (8.8)$$

where,

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \frac{6.341719u_0}{V} - k_1 - 2k_3C_{A0} & 0 \\ k_1 & \frac{-6.341719u_0}{V} - k_2 \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} \frac{6.341719(C_{A0}-C_{A0})}{V} & \frac{6.341719u_0}{V} \\ \frac{-6.341719C_{B0}}{V} & 0 \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} 0 & \frac{100}{1.5714} \end{bmatrix}. \end{aligned}$$

Using the parameters of Table 8.4 and the operation point values, the resulting transfer function between input u and output y is found to be:

$$F(s) = \frac{-0.6342s + 1.913}{s^2 + 4.56s + 5.193} \quad (8.9)$$

It has to be clear that this transfer function is only an approximation of the model, and therefore it is valid only around the operating point. When a step change of 5% is entered in the system, the response of the nonlinear model and the response of the transfer function start to differ as presented in Figure 8.19. As it can be seen,

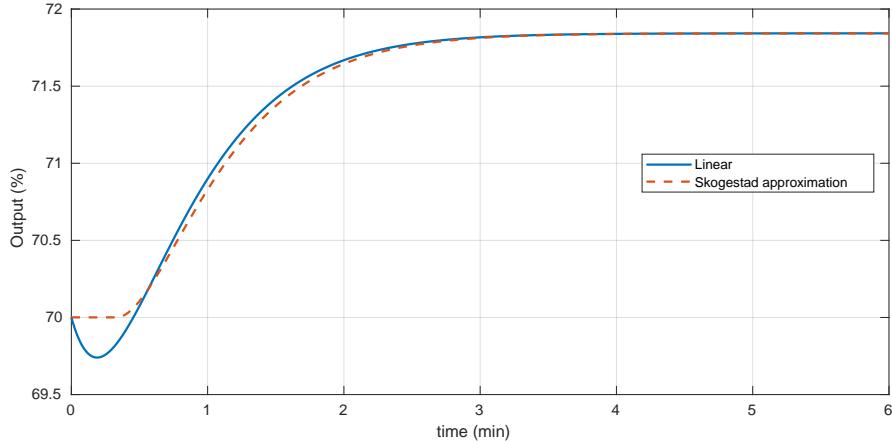


Fig. 8.20: Approximation of the non-minimum phase zero with a time delay for the linear model of the CSTR.

the linear model represents the transient response well, however it fails to predict the steady state. Of course, the smaller the change in the input, the better is the the approximation for both the transient and the steady state response.

However, the model that is expected in the MATLAB app does not contemplate a non-minimum phase zero in the model. Using the model reduction rules of Skogestad (2003), the ODSOPTD model ends as:

$$F_{aprox} = \frac{0.368e^{-0.331}}{0.193s^2 + 0.878s + 1}. \quad (8.10)$$

which corresponds to approximate the non-minimum phase zero by a time delay. The gain is given by $K = 0.3684$, the time constant is given by $\tau = 0.4256$ min, the time delay is $L = 0.3315$ min and $a = 0.9408$. This new transfer function gives the response presented in Figure 8.20. Of course, the dynamic model is quite different from the original nonlinear model, both in the transient and steady states. Since the controller is going to be designed with the linearized time-delayed transfer function, it is necessary to consider some ~~restriction~~ constraint on the performance in order to give certain robustness to the design.

8.2.3 Controller design

For this particular example, the design of the controller parameters will contemplate three different robustness cases: without any ~~restriction~~ constraint to the value of M_s , $M_s = 2.0$ and $M_s = 1.8$. The MOOTuning app is able to introduce the robustness as a constraint

Table 8.5: Different tunings for the CSTR obtained with MOOTuning.

Design	K_p	T_i	T_d	β	J_{di}	J_r
$M_s \geq 2.0$						
C1	6.00	0.61	0.29	0.49	0.13	0.98
C2	4.54	1.24	0.26	0.99	0.27	0.80
C3	5.54	0.85	0.26	0.66	0.14	0.85
C4	5.33	0.91	0.26	0.73	0.17	0.84
$M_s = 2.0$						
C1	4.43	0.76	0.21	0.66	0.17	0.92
C2	4.29	1.19	0.25	0.99	0.28	0.80
C3	4.44	0.86	0.22	0.73	0.19	0.86
C4	4.40	1.00	0.24	0.87	0.23	0.83
$M_s = 1.8$						
C1	3.88	0.72	0.22	0.72	0.20	0.98
C2	3.86	1.13	0.24	0.99	0.29	0.82
C3	3.91	0.85	0.21	0.75	0.22	0.89
C4	3.92	0.96	0.22	0.87	0.24	0.85

constraint

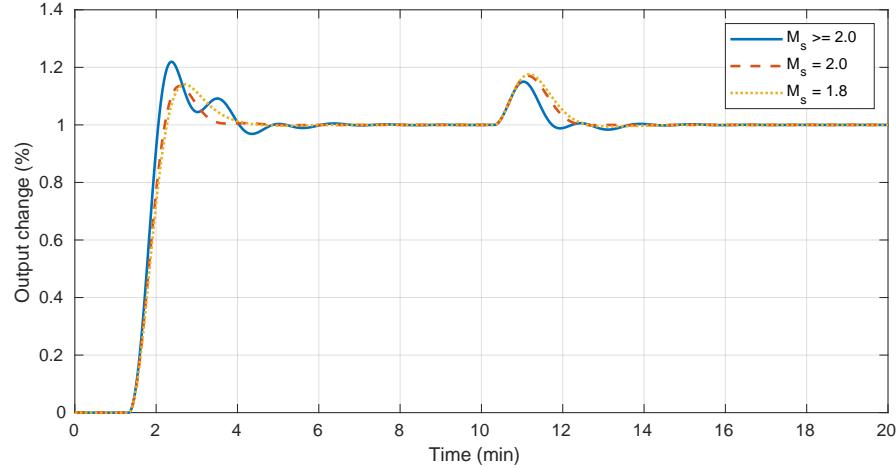
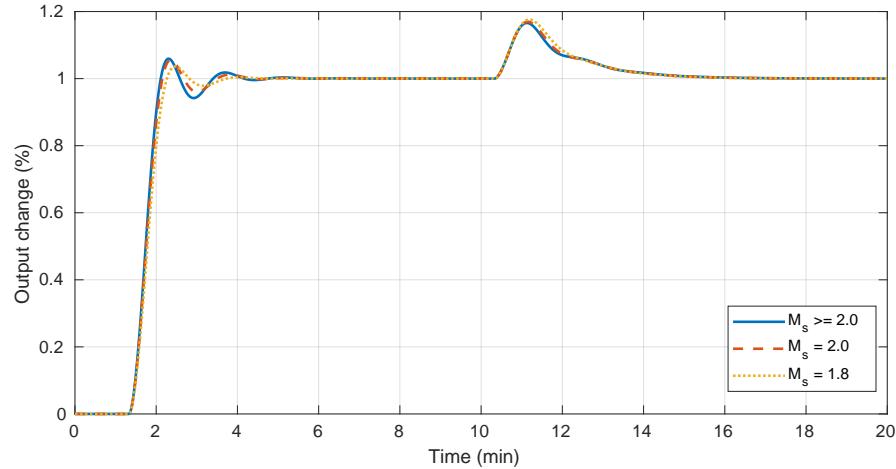
~~restriction~~ in the design. However, it is limited to certain values of M_s which are considered to be reasonable for an industrial control applications.

In this particular example, only J_r and J_{di} are going to be considered. The design is based on the model in (8.10) and will be tested on the nonlinear model as well. Several points of the Pareto are going to be tested: the two anchor points (C1 for best regulator and C2 for best servo), the best regulator possible with a 20% degradation on J_r (C3) and the best servo possible with a 20% degradation on J_{di} (C4).

All the tuning values for the cases studied are presented in Table 8.5. In general, it can be seen that the proportional gain is heavily dependent of the robustness value (more robustness implies lower values of K_p). On the other hand it seems that the value of T_i is more dependent of the degradation of J_{di} while the variation of T_d is small across all cases. The value of β seems like a combination of the degradation of J_{di} and the value of the robustness.

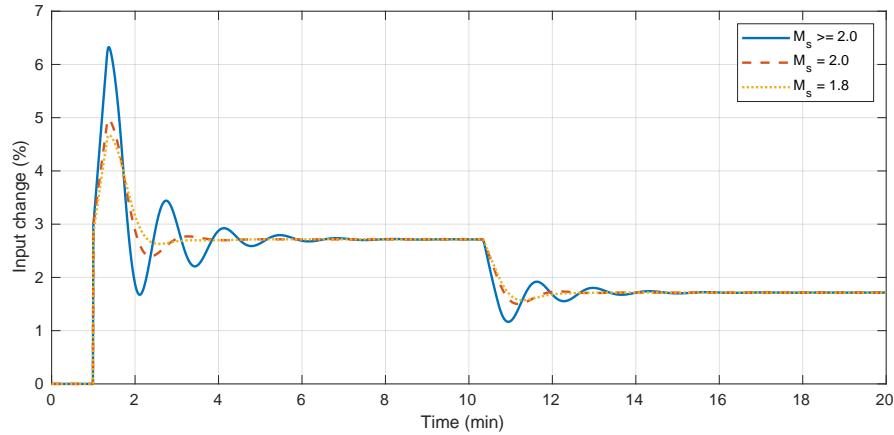
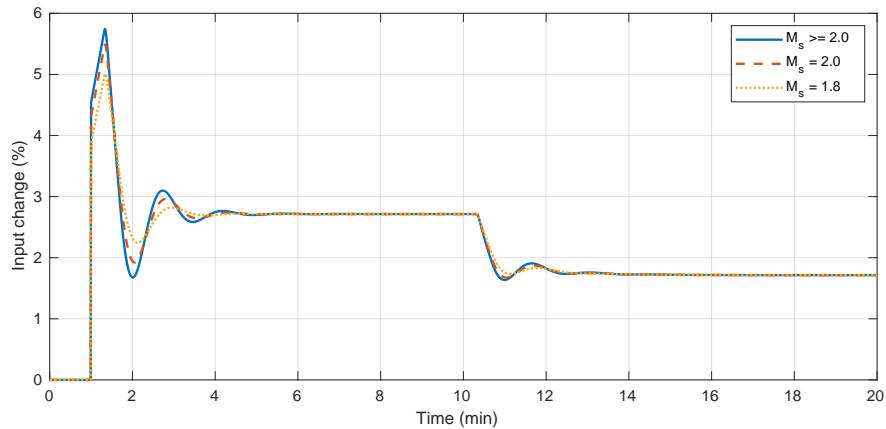
To compare the responses, it is useful to plot the step response of C1 controllers across all robustness values as presented in Figure 8.21 and for C2 controllers as in Figure 8.22. It is interesting to note that C1 controllers presents more variability in the response with respect to the controllers of the C2 family. In both figures it is clear that there is a compromise between the servo and the regulator responses, but this compromise becomes less important when the robustness is considered. Consider Figure 8.21, the response for $M_s \geq 2.0$ does not take into account any constraint on the robustness and as it can be seen this response is very different than the cases where $M_s = 2.0$ and $M_s = 1.8$ are forced. It has to be noticed that adding the robustness constraint greatly limits the possible values of the controllers.

On the other hand, it was found that for the C2 controllers family the responses are very similar among all robustness, as can be seen in Figure 8.22. The reason

Fig. 8.21: Response for C1 controllers for all M_s values.Fig. 8.22: Response for C2 controllers for all M_s values.

for this is the parameter β . This parameter does not affect the robustness value of the controlled system nor the regulator response. Therefore, the optimization tends to find a low value for K_p , which gives better robustness, but then compensates with a high value of β .

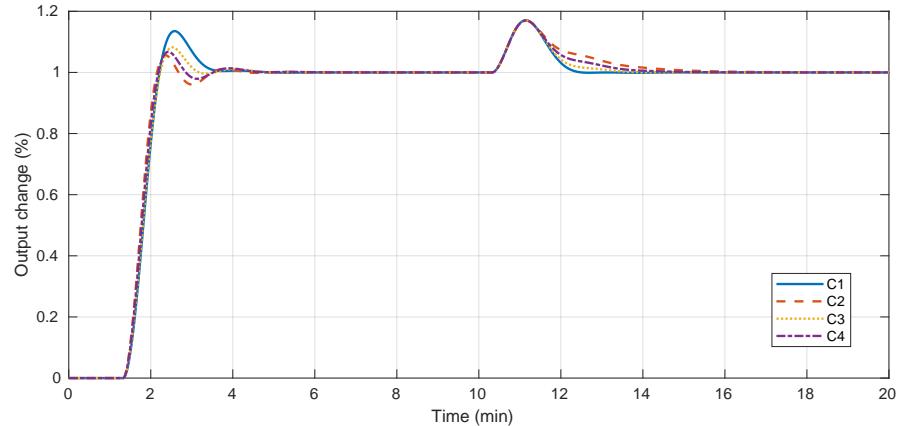
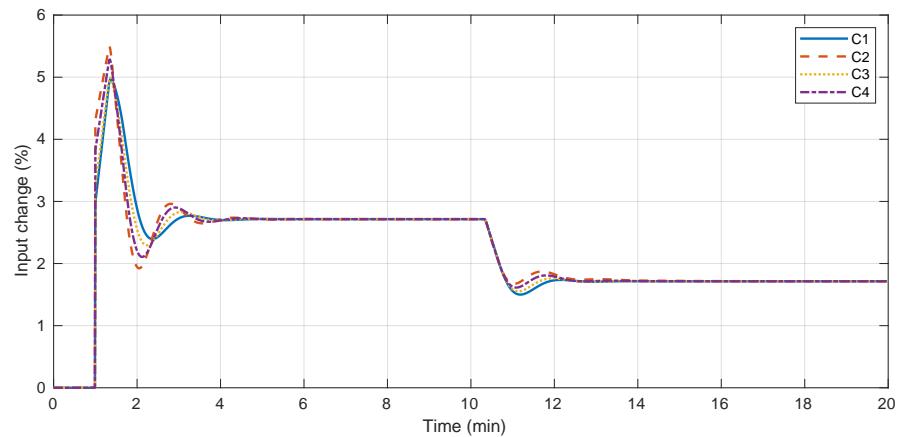
The control signals for the controllers are presented in Figures 8.23 and 8.24. For the case of C1 controllers, the control signal varies significantly according to the robustness value. The case without any constraint has a larger peak and more oscillatory response than any of the other controllers. It is interesting, however, to note that the control signal for the C2 controllers are very similar and again, this

Fig. 8.23: Control signal for C1 controllers for all M_s values.Fig. 8.24: Control signal for C2 controllers for all M_s values.

is due to the presence of the β parameter. The fact that an unconstrained controller may be applied to the controlled system has to be taken with caution, because in this design, the model used for the controller tuning is known to be different than the system. Therefore, applying an aggressive control signal may lead to unwanted oscillatory behavior and even instability.

Now, the responses of the different controllers families can be compared for the same given robustness, for example $M_s = 2.0$ as in Figure 8.25. For all this controllers, the robustness is near $M_s = 2.0$, but the performance is varied from J_r and J_{di} . Here, the compromise between both responses is clearer since the best servo is at the same time the worst regulator and the best regulator is the worst servo.

However, given that all controllers are constrained to fulfill $M_s = 2.0$, the difference between them are not very notorious. It can be seen on Figure 8.26 that the constrained to

Fig. 8.25: Response for all controller families with $M_s = 2.0$.Fig. 8.26: Control signal for all controller families with $M_s = 2.0$.

peaks and oscillatory behavior of all controllers are practically the same. When the robustness became too ~~thing~~ (for example values below 1.4) the controllers for servo and regulator practically become the same and the degree of freedom to select the dynamic behavior is practically non-existent.

is this?

8.2.4 Validation of the controller designs

The designed controllers were tested using the nonlinear model of the CSTR. It is important to note that the controllers were tuned for a plant model that is different than the “real” plant. In this particular case, several approximations where made

Table 8.6: IAE values for the controllers applied to the nonlinear model as servo controllers

Robustness	Controller			
	C1	C2	C3	C4
$M_s = 1.8$	0.94	0.83	0.87	0.84
$M_s = 2.0$	0.86	0.79	0.83	0.81
$M_s \geq 2.0$	11.20	0.78	1.02	0.83

Table 8.7: TV values for the controllers applied to the nonlinear model as servo controllers

Robustness	Controller			
	C1	C2	C3	C4
$M_s = 1.8$	9.40	13.20	9.14	11.10
$M_s = 2.0$	11.34	20.01	12.75	16.37
$M_s \geq 2.0$	1936.10	29.10	128.30	73.20

Uncertainty, therefore instability

which were necessary in order to use the framework presented in the other chapters. Because of this, it is important to look for a solution that takes into account this possible sources of instability. In Table 8.6, the IAE for the servo response to a step change in the reference at $t = 1\text{ s}$ is presented, with the response plotted in Figure 8.27. On the other hand the total variation is presented in Table 8.7 and the control signal in Figure 8.28. All controllers where tested for all possible robustness measures.

The first thing to notice is that for the case of $M_s \geq 2.0$, presented in Figure 8.27c, the response becomes practically unstable for the C1 tuning. The PID controller that was implemented in Simulink have and antiwindup loop that prevents the integral part to become infinite. For the other cases the response presents important oscillations. The problem with C1 is that the gain is relatively high, which produces the controller to saturate as can be seen in Figure 8.28c.

For this particular case, controller C2 (which was expected to be the “best” servo) yields a better IAE when comparing the $M_s \geq 2.0$ against $M_s = 1.8$ and $M_s = 2.0$ as it was expected knowing Table 8.5. However this is not the case for C4 controllers, because the nonlinearity of the plant and the approximations made start to take a toll on the design of the controllers.

Now consider the controllers that took into account the robustness as a constraint in the optimization. For all cases the controller are able to control the plant without oscillation (except for controller C1). As expected, controllers C2 and C4 had the best performance, but also the were more expensive (higher values of TV). However, an interesting option is controller C3. This controller was designed by allowing a 20% degradation of J_{di} . Its servo response may not be the best in terms of performance but it has an interesting compromise between the control effort and the

yields

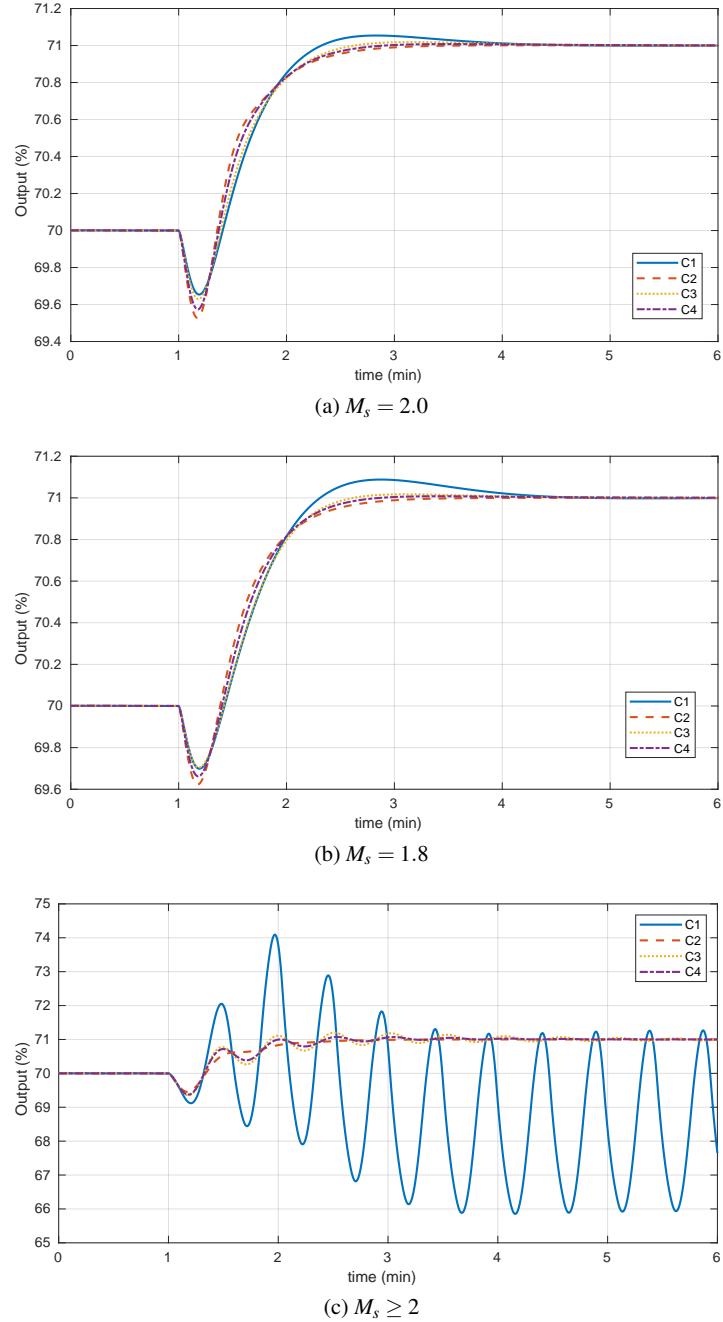


Fig. 8.27: Response of the controlled system with the nonlinear model for several robustness levels serving as servo.

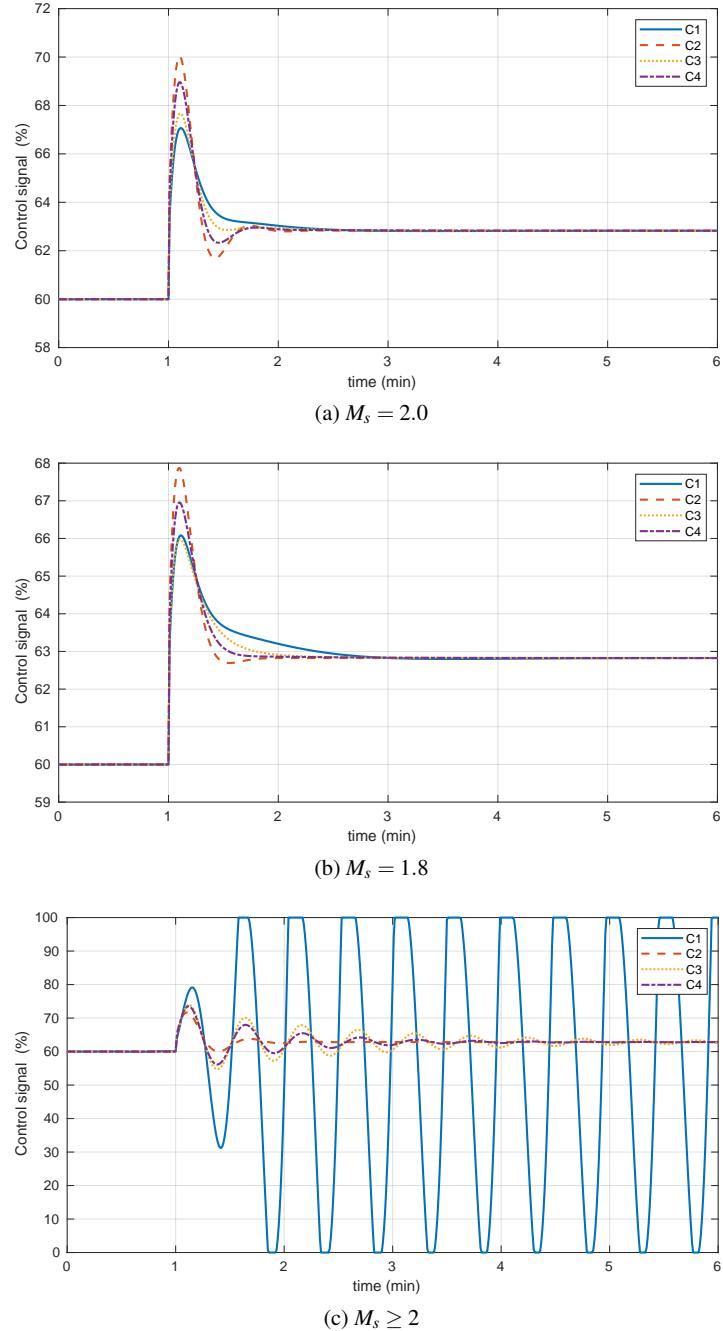


Fig. 8.28: Control signal of the controlled system with the nonlinear model for several robustness levels serving as servo.

Table 8.8: IAE values for the controllers applied to the nonlinear model as regulator controllers

Robustness	Controller			
	C1	C2	C3	C4
$M_s = 1.8$	2.29	3.40	2.54	2.85
$M_s = 2.0$	2.03	3.24	2.26	2.65
$M_s \geq 2.0$	11.62	3.18	1.97	2.00

Table 8.9: TV values for the controllers applied to the nonlinear model as regulator controllers

Robustness	Controller			
	C1	C2	C3	C4
$M_s = 1.8$	13.30	11.85	12.50	11.65
$M_s = 2.0$	13.68	15.54	13.16	14.10
$M_s \geq 2.0$	3034.50	22.00	183.90	84.00

IAE value. If this degradation does not affect the regulator response too much, this controller may be considered as the final tuning. *In this*

The next step is then to validate the response as a regulator. ~~Table~~ 8.8 the values of IAE are presented for a step change of 1 mol L^{-1} in C_{ai} for all controller cases and robustness values. The corresponding values of TV are presented in Table 8.9. The plots of the responses and the control signal are shown in Figure 8.29 and Figure 8.30 respectively. As in the case of the servo response, the design for $M_s \geq 2.0$ is not useful in the C1 case, because the response is practically unstable again as depicted in Figure 8.29c. It has to be noticed that the response of the plant between C_{ai} and the output does not have a non-minimum phase zero, and therefore an inverse response is not present.

For the other cases ($M_s = 2.0$ and $M_s = 1.8$) the best controllers were the C1 family, followed by C3. This is expected since these controllers were found as the best regulators. Controller C2 has the worst regulator response but at the same time, it is also the most expensive cost signal for the case $M_s = 2.0$. *these*

Let's examine C3 controllers. The performance of ~~the~~ controllers are worse than C1, but just by approximately 11%, and with a less aggressive control signal for the regulator response (Table 8.9). Taking into account that C3 was also a family of controllers that had a relatively good performance for servo control, so far it represents a good candidate to become the final controller.

As a final validation test, let the setpoint change to be 5%. In that case the response is as given in Figure 8.31 for all controllers families and $M_s = 2.0$.

The values of IAE and TV are presented in Table 8.10. The PID was implemented to be limited in the range between 0 and 100%, with the corresponding antiwindup. Since controllers C2 and C4 have a control signal which is more aggressive for set-

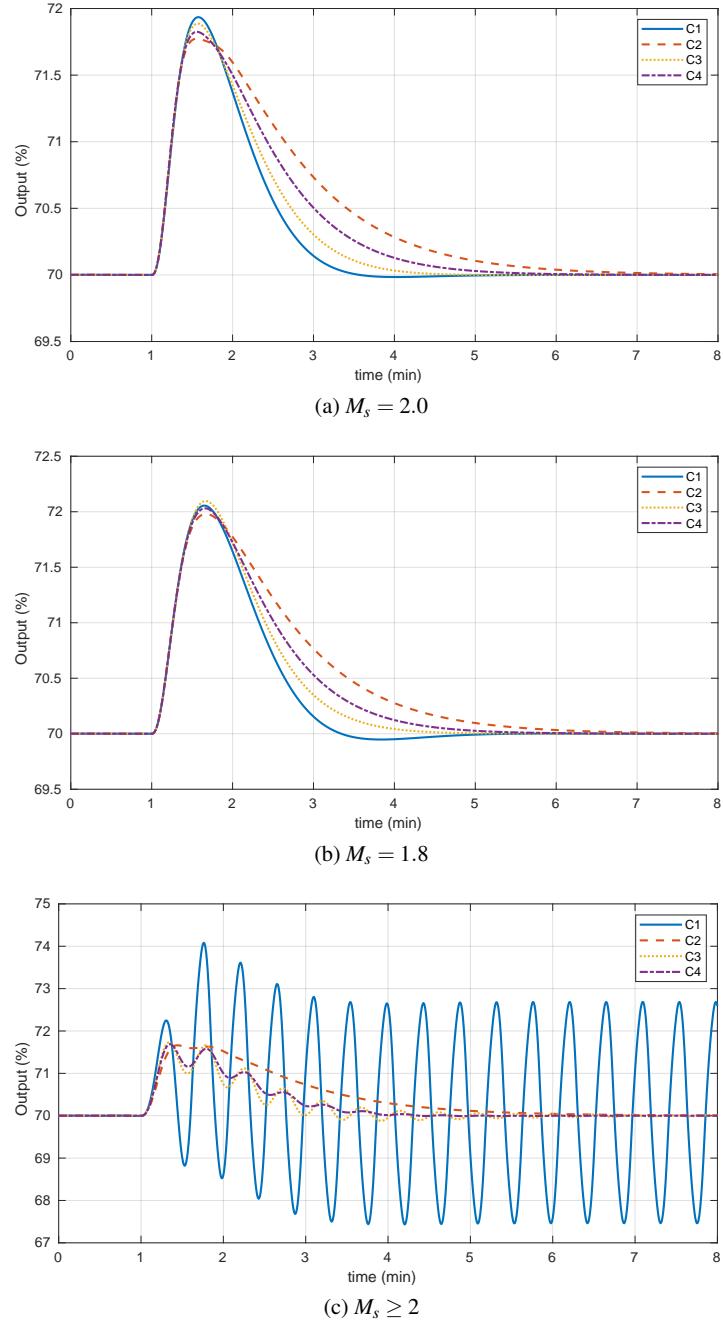


Fig. 8.29: Response of the controlled system with the nonlinear model for several robustness levels serving as regulator.

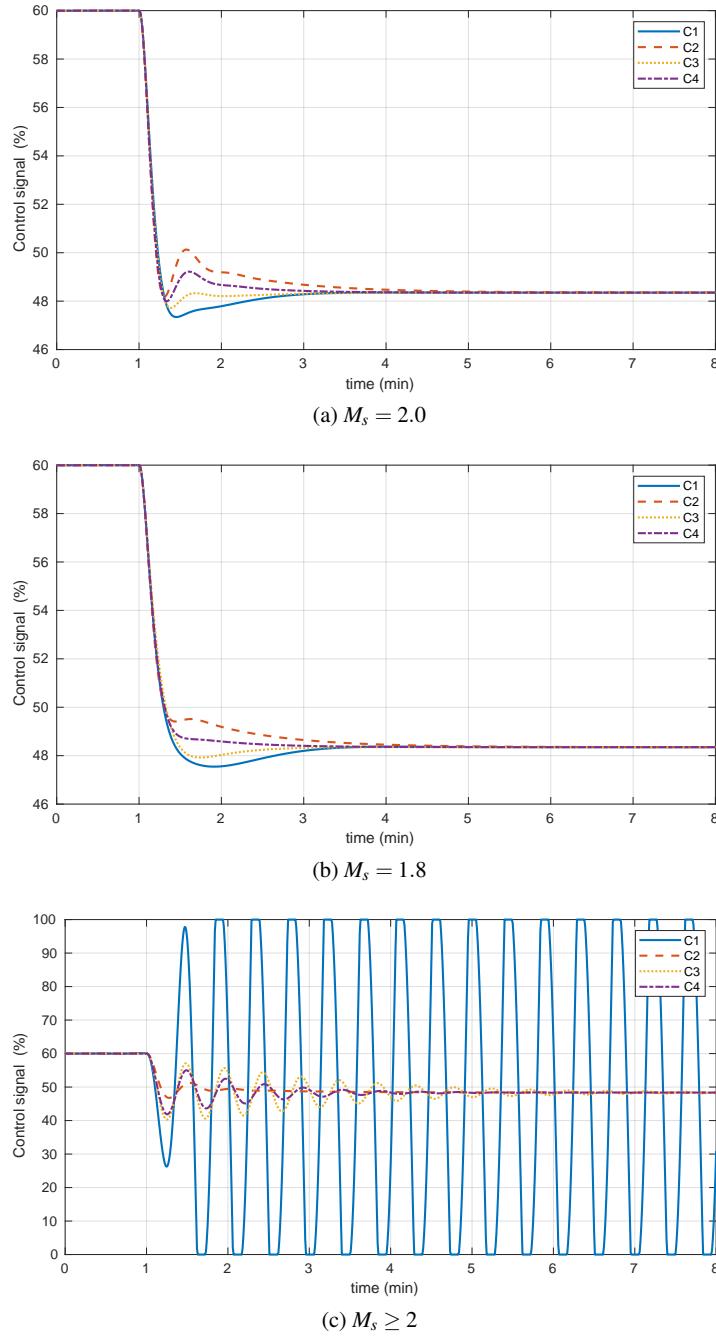


Fig. 8.30: Control signal of the controlled system with the nonlinear model for several robustness levels serving as regulator.

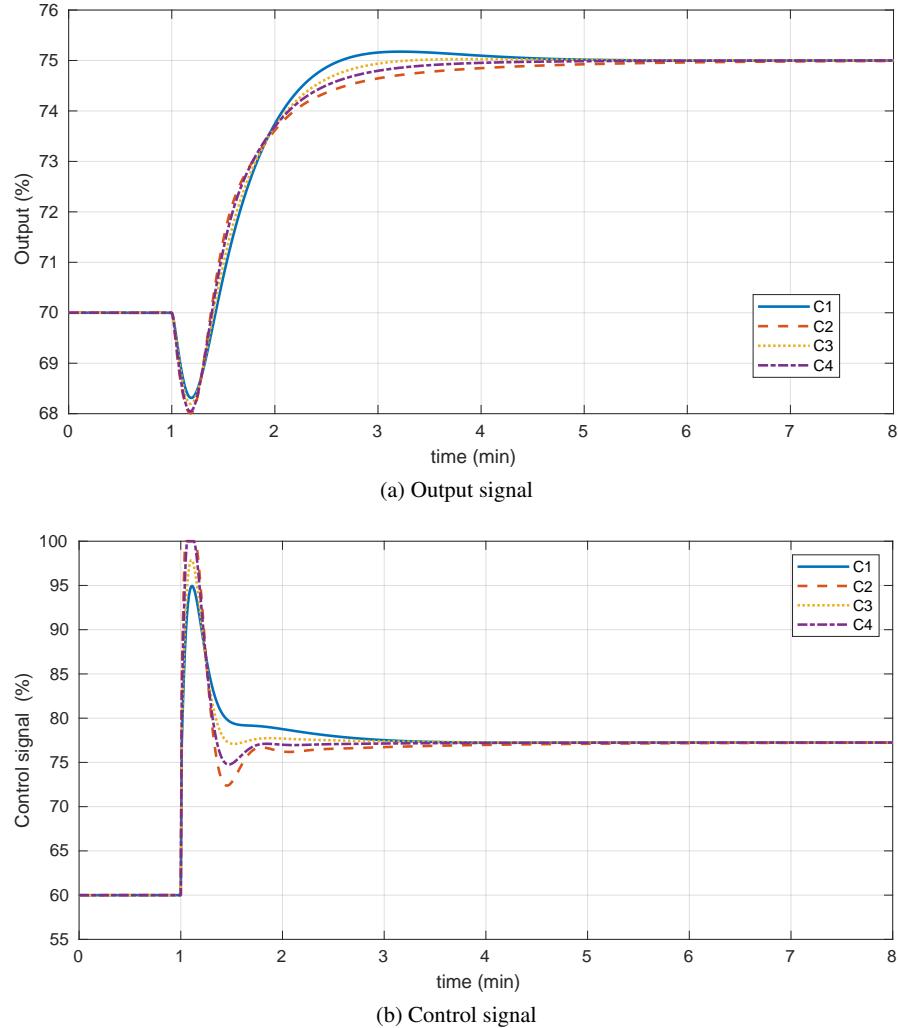


Fig. 8.31: Response to a 5% step change in the setpoint signal.

point changes, they saturate and produces larger values of IAE than the controllers intended for regulation. Interestingly, C3 controller has a lower value of IAE than any other controller while its TV value lies in between C1 and C4 controllers. Again, this test indicates that C3 controller can be a good candidate for the final tuning of the PID.

Lastly, lets again set the change in the setpoint to 5%, but lets compare the difference when C3 is selected as the controller and the M_s is varied. In Figure 8.32 the output of the controlled loop and the control signal is presented and in Table 8.11, the values of IAE and TV are presented for both $M_s = 1.8$ and $M_s = 2.0$. As it

Table 8.10: IAE values for the controllers applied to the nonlinear model as servo controller for a step change of 5% and $M_s = 2.0$

Cost function	Controller			
	C1	C2	C3	C4
IAE	4.69	5.07	4.60	4.71
TV	52.70	73.44	59.81	68.01

Table 8.11: IAE values for the C3 controllers applied to the nonlinear model as servo controller for a step change of 5%

Cost function	M_s	
	1.8	2.0
IAE	4.83	4.61
TV	42.11	59.81

was expected, since neither of those controllers saturates, the response with a less restrictive robustness constraint has a better IAE. However it certainly has a more aggressive control signal. The IAE for the case $M_s = 2.0$ is 4.55% lower than the $M_s = 1.8$ case however it comes to a cost of having a 42.03% higher TV. Considering all the analysis done to this point and having into account that the tuning were made with several approximations from the original model, it seems that the C3 tuning (best servo allowing a 20% degradation on J_{di}) with a robustness constraint of $M_s = 1.8$ is the most sensible choice as the final tuning.

8.3 Final remarks

The motivation of the examples presented in this chapter was to show all the advantages that can be derived from using a multiobjective approach when tuning a PID controller for industrial applications. As has been shown, the final selection of the parameters was defined not only by its optimal value, but also, according to the robustness needs and the level of compromise between the cost functions.

However, it has to be noticed that the cost functions selected for these studies are totally arbitrary and other authors may choose to optimize the tuning of the parameters with other criteria. For example, in Section 5.1.2, the total variation was selected as one of the cost functions and the Pareto found was considerably different from the ones found using J_{di} , J_{do} and J_r . But it is known that the IAE is a practical measure of the optimality of the control in industry (Shinskey, 2002), and for this reason was the selected cost function for the tool presented in this book and consequently the examples examined in this chapter.

respect to

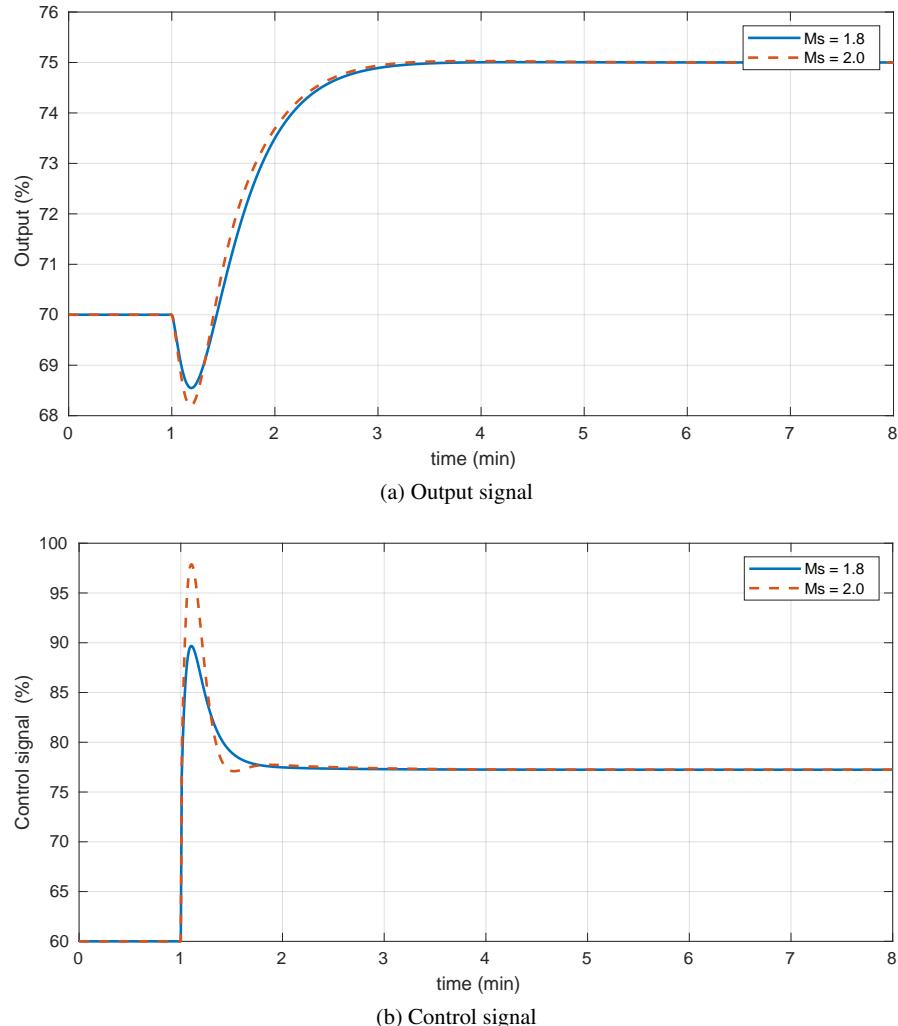


Fig. 8.32: Response to a 5% step change in the setpoint signal for C3 controllers.

Apart from the theoretical contribution of this book, the MATLAB tool that is included along with the complete set of data, represents an interesting starting point for further studies. Of course the methodology presented in 7.1 is completely general and can (and is encouraged⁴) to be changed to the needs of the decision maker.

One of the most interesting characteristics of control systems is that some kind of compromise is always involved. The relationship between servo and regulation control, or between performance and aggressiveness is always something that has to be taken into account along with the need to have a robust control system that is able to keep working despite the difference between the model and the actual

plant. It is the desire of the authors to have helped in the development of a deeper understanding on this issues, and to open the door to more research in the field of optimization applied to industrial control.

References

- Alfaro VM, Vilanova R (2016) Model-Reference Robust Tuning of PID Controllers. *Advances in Industrial Control*, Springer International Publishing, Cham, DOI 10.1007/978-3-319-28213-8, URL <http://link.springer.com/10.1007/978-3-319-28213-8>
- Arrieta O, Vissoli A, Vilanova R (2010) PID autotuning for weighted servo/regulation control operation. *Journal of Process Control* 20(4):472–480, DOI 10.1016/j.jprocont.2010.01.002
- Murril PW (1967) Automatic control of processes. International Textbook Co.
- O'Dwyer A (2009) Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London, UK
- Rovira AJA, Murrill PW, Smith CL (1969) Tuning controllers for setpoint changes. *Instruments and Control Systems* 42:67–69
- Shinskey FG (2002) Process Control: As Taught vs as Practiced. *Industrial & Engineering Chemistry Research* 41(16):3745–3750, DOI 10.1021/ie010645n, URL <https://pubs.acs.org/doi/10.1021/ie010645n>
- Skogestad S (2003) Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13(4):291–309, DOI 10.1016/S0959-1524(02)00062-8, URL <https://www.sciencedirect.com/science/article/pii/S0959152402000628> <https://linkinghub.elsevier.com/retrieve/pii/S0959152402000628>
- van de Vusse J (1964) Plug-flow type reactor versus tank reactor. *Chemical Engineering Science* 19(12):994–996, DOI 10.1016/0009-2509(64)85109-5, URL <https://linkinghub.elsevier.com/retrieve/pii/0009250964851095>

References

- Abbas A, Sawyer P (1995) A multiobjective design algorithm: Application to the design of isiso control systems. *Computers & Chemical Engineering* 19(2):241–248, DOI 10.1016/0098-1354(94)00044-O, URL <https://linkinghub.elsevier.com/retrieve/pii/009813549400044O>
- Alcantara S, Pedret C, Vilanova R, Zhang WD (2010) Simple Analytical min-max Model Matching Approach to Robust Proportional-Integrative-Derivative Tuning with Smooth Set-Point Response. *Industrial & Engineering Chemistry Research* 49(2):690–700
- Alcantara S, Vilanova R, Pedret C (2013) PID control in terms of robustness/performance and servo/regulator trade-offs: A unifying approach to balanced autotuning. *Journal of Process Control* 23(4):527–542
- Alfaro VM (2006a) Identificación de Modelos de orden reducido a partir de la curva de reacción del Proceso. *Ciencia y Tecnología (UCR)* 24(2):2
- Alfaro VM (2006b) Low-Order Models Identification from the Process Reaction Curve. *Ciencia y Tecnología (Costa Rica)* 24(2):197–216
- Alfaro VM, Vilanova R (2012a) Conversion Formulae and Performance Capabilities of Two-Degree-of-Freedom PID Control Algorithms. In: 17th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA 2012)
- Alfaro VM, Vilanova R (2012b) 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, DOI 10.1016/J.JPROCONT.2012.01.001, URL <https://www.sciencedirect.com/science/article/pii/S0959152412000042>
- Alfaro VM, Vilanova R (2012c) Optimal Robust Tuning for 1DoF PI/PID Control Unifying FOPDT/SOPDT Models. *IFAC Proceedings Volumes* 45(3):572–577, DOI <http://dx.doi.org/10.3182/20120328-3-IT-3014.00097>, URL <http://www.sciencedirect.com/science/article/pii/S1474667016310874>
- Alfaro VM, Vilanova R (2012d) Set-point weight selection for robustly tuned PI/PID regulators for over damped processes. In: Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on, pp 1–7, DOI 10.1109/ETFA.2012.6489607
- Alfaro VM, Vilanova R (2013a) Robust tuning of 2DoF five-parameter PID controllers for inverse response controlled processes. *Journal of Process Control* 23(4):453–462, DOI 10.1016/j.jprocont.2013.01.005, URL <https://www.sciencedirect.com/science/article/pii/S0959152413000152> <https://linkinghub.elsevier.com/retrieve/pii/S0959152413000152>
- Alfaro VM, Vilanova R (2013b) Simple Robust Tuning of 2DoF PID Controllers From A Performance/Robustness Trade-off Analysis. *Asian Journal of Control* 15(6):1700–1713, DOI 10.1002/asjc.653, URL <http://dx.doi.org/10.1002/asjc.653>
- Alfaro VM, Vilanova R (2016) Model-Reference Robust Tuning of PID Controllers. *Advances in Industrial Control*, Springer International Publishing, Cham, DOI 10.1007/978-3-319-28213-8, URL <http://link.springer.com/10.1007/978-3-319-28213-8>
- Arrieta O, Vilanova R (2010) Performance degradation analysis of controller tuning modes: application to an optimal PID tuning. *Int J Innov Comput Inf Control* 6(10):4719–4729
- Arrieta O, Vilanova R (2012) Simple Servo/Regulation Proportional-Integral-Derivative (PID) Tuning Rules for Arbitrary Ms-Based Robustness Achievement. *Industrial & Engineering Chemistry Research* 51(6):2666–2674
- Arrieta O, Vilanova R, Alfaro VM, Moreno R (2008) Considerations on PID Controller Operation: Application to a Continuous Stirred Tank Reactor. In: 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA08), September 15-18, Hamburg-Germany
- Arrieta O, Visioli A, Vilanova R (2010) PID autotuning for weighted servo/regulation control operation. *Journal of Process Control* 20(4):472–480, DOI 10.1016/j.jprocont.2010.01.002

- Åström K, Hägglund T (1984) Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica* 20(5):645–651, DOI 10.1016/0005-1098(84)90014-1, URL <https://linkinghub.elsevier.com/retrieve/pii/0005109884900141>
- Åström KJ, Hägglund T (1995a) PID Controllers: Theory, Design and Tuning. Instrument Society of America, Research Triangle Park, NC 27709, USA
- Åström KJ, Hägglund T (1995b) PID Controllers: Theory, Design and Tuning. Instrument Society of America
- Åström KJ, Hägglund T (2000) Benchmark Systems for PID Control. In: Proceedings IFAC Workshop Digital Control: Past, Present and Future of PID Control
- Åström KJ, Hägglund T (2006) Advanced PID Control. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, USA
- Bagis A (2011) Tabu search algorithm based PID controller tuning for desired system specifications. *Journal of the Franklin Institute* 348(10):2795–2812, DOI 10.1016/j.jfranklin.2011.09.001
- Benki A, Habbal A, Mathis G (2018) A metamodel-based multicriteria shape optimization process for an aerosol can. *Alexandria Engineering Journal* 57(3):1905–1915, DOI 10.1016/J.AEJ.2017.03.036, URL <https://www.sciencedirect.com/science/article/pii/S1110016817301291>
- Binitha S, Sathya SS (2012) A Survey of Bio inspired Optimization Algorithms. *International Journal of Soft Computing and Engineerin* 2(2):137 –151, URL <http://www.ijscce.org/wp-content/uploads/papers/v2i2/B0523032212.pdf>
- Blasco X, Herrero J, Sanchis J, Martínez M (2008) A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences* 178(20):3908–3924, DOI 10.1016/J.INS.2008.06.010, URL <https://www.sciencedirect.com/science/article/pii/S0020025508002016>
- Bosman PAN, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 7(2):174–188, DOI 10.1109/TEVC.2003.810761
- Brito T, Paiva A, Ferreira J, Gomes J, Balestrassi P (2014) A normal boundary intersection approach to multiresponse robust optimization of the surface roughness in end milling process with combined arrays. *Precision Engineering* 38(3):628–638, DOI 10.1016/j.precisioneng.2014.02.013, URL <https://www.sciencedirect.com/science/article/pii/S0141635914000439>
- Cao Y (2020) Pareto Set. URL <https://www.mathworks.com/matlabcentral/fileexchange/15181-pareto-set>
- Cespedes M, Contreras M, Cordero J, Montoya G, Valverde K, Rojas JD (2016) A comparison of bio-inspired optimization methodologies applied to the tuning of industrial controllers. In: 2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI), IEEE, pp 1–6, DOI 10.1109/CONCAPAN.2016.7942340, URL <http://ieeexplore.ieee.org/document/7942340/>
- Chen D, Seborg DE (2002) PI/PID Controller Desing Based on Direct Synthesis and Disturbance Rejection. *Ind Eng Chem Res* 41:4807–4822
- Chien IL, Fruehauf PS (1990) Consider IMC Tuning to Improve Controller Performance. *Chemical Engineering Progress* 86:33–41
- Chiha I, Liouane N, Borne P (2012) Tuning PID Controller Using Multiobjective Ant Colony Optimization. *Applied Computational Intelligence and Soft Computing* 2012:1–7, DOI 10.1155/2012/536326, URL <http://www.hindawi.com/journals/acisc/2012/536326/>
- Contreras Leiva M, Rojas JD (2015) New Tuning Method for PI Controllers based on Pareto-Optimal Criterion with Robustness Constraint. *IEEE Latin America Transactions* 13(2):434–440, DOI 10.1109/TLA.2015.7055561
- Contreras-Leiva MP, Rivas F, Rojas JD, Arrieta O, Vilanova R, Barbu M (2016) Multi-objective optimal tuning of two degrees of freedom PID controllers using the ENNC method. In: 20th International Conference on System Theory, Control and Computing, Sinaia, Romania

- Corripi AB (2001) Tuning of Industrial Control Systems, 2nd edn. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, USA.
- Dahlin EG (1968) Designing and Tuning Digital Controllers. *Instrumentation and Control Systems* 41(6):77–81
- Das A, Mazumder T, Gupta A (2012) Pareto frontier analyses based decision making tool for transportation of hazardous waste. *Journal of Hazardous Materials* 227–228:341–352, DOI 10.1016/j.jhazmat.2012.05.068, URL <https://www.sciencedirect.com/science/article/pii/S0304389412005717> <https://linkinghub.elsevier.com/retrieve/pii/S0304389412005717>
- Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization* 14(1):63–69, DOI 10.1007/BF01197559
- Das I, Dennis JE (1998) Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization* 8(3):631–657, DOI 10.1137/S1052623496307510
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197, DOI 10.1109/4235.996017, URL <http://ieeexplore.ieee.org/document/996017>
- Domingues PH, Freire RZ, Coelho LDS, Ayala HV (2019) Bio-Inspired Multiobjective Tuning of PID-Controlled Antilock Braking Systems. 2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings pp 888–895, DOI 10.1109/CEC.2019.8790023
- Dorigo M, Blum C (2005) Ant colony optimization theory: A survey. *Theoretical Computer Science* 344(2-3):243–278, DOI 10.1016/j.tcs.2005.05.020, URL <https://linkinghub.elsevier.com/retrieve/pii/S0304397505003798>
- Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Computational Intelligence Magazine* 1(4):28–39, DOI 10.1109/MCI.2006.329691, URL <http://ieeexplore.ieee.org/document/4129846/>
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, vol 0-7803-267, pp 39–43, DOI 10.1109/MHS.1995.494215, URL <http://ieeexplore.ieee.org/document/494215/>
- Filipič B, Tušar T (2018) A taxonomy of methods for visualizing pareto front approximations. In: Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO '18, ACM Press, New York, New York, USA, pp 649–656, DOI 10.1145/3205455.3205607, URL <http://dl.acm.org/citation.cfm?doid=3205455.3205607>
- Gambier A (2009) Optimal PID controller design using multiobjective Normal Boundary Intersection technique. In: Asian Control Conference, 2009. ASCC 2009. 7th, Hong Kong, China, pp 1369–1374
- Gambier A, Badreddin E (2007) Multi-objective Optimal Control: An Overview. In: 2007 IEEE International Conference on Control Applications, pp 170–175, DOI 10.1109/CCA.2007.4389225
- Ganesan T, Vasant P, Elamvazuthi I (2013) Normal-boundary intersection based parametric multi-objective optimization of green sand mould system. *Journal of Manufacturing Systems* 32(1):197–205, DOI 10.1016/j.jmsy.2012.10.004, URL <https://www.sciencedirect.com/science/article/pii/S0278612512000933> <https://linkinghub.elsevier.com/retrieve/pii/S0278612512000933>
- Gao H, Nie H, Li K (2019) Visualisation of Pareto Front Approximation: A Short Survey and Empirical Comparisons. In: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 1, pp 1750–1757, DOI 10.1109/CEC.2019.8790298, URL <https://ieeexplore.ieee.org/document/8790298/>
- Garpinger O, Hägglund T, Åström KJ (2012) Criteria and Trade-offs in PID Design. In: Proceedings of the IFAC Conference on Advances in PID Control, Brescia, Italy
- Gerry JP (1987) A comparison of PID algorithms. *Control Engineering* 34 (3):102–105

- Goss S, Aron S, Deneubourg JL, Pasteels JM (1989) Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* 76(12):579–581, DOI 10.1007/BF00462870, URL <http://link.springer.com/10.1007/BF00462870>
- Grimholt C, Skogestad S (2012) Optimal PI-Control and Verification of the SIMC Tuning Rule. In: IFAC Conference on Advances in PID Control PID'12, Brescia (Italy)
- Hägglund T, Åström KJ (2008) Revisiting The Ziegler-Nichols Tuning Rules For Pi Control. *Asian Journal of Control* 4(4):364–380, DOI 10.1111/j.1934-6093.2002.tb00076.x, URL <http://doi.wiley.com/10.1111/j.1934-6093.2002.tb00076.x>
- Han Y, Brdys M, Piotrowski R (2008) Nonlinear PI control for dissolved oxygen tracking at wastewater treatment plant. *IFAC Proceedings Volumes* 41(2):13,587–13,592, DOI 10.3182/20080706-5-KR-1001.02301, URL <http://dx.doi.org/10.3182/20080706-5-KR-1001.02301> <https://linkinghub.elsevier.com/retrieve/pii/S1474667016411675>
- Henze M, Harremoës P, Arvin E, Jansen JJC (1997) Wastewater treatment, biological and chemical process, 2nd edn. Springer Verlag, New York, USA
- Herreros A, Baeyens E, Peran JR (2002) Design of PID-type controllers using multiobjective genetic algorithms. *ISA Transactions* 41(4):457–472
- Ho WK, Hang CC, Cao LS (1995) Tuning PID Controllers Based on Gain and Phase Margin Specifications. *Automatica* 31(3):497–502
- Hoffman PE, Grinstein GG, Marx K, Grosse I, Stanley E (2002) A survey of visualizations for high-dimensional data mining. In: Information visualization in data mining and knowledge discovery, pp vol. 104, p. 4
- Hosseini SA, Amjadi N, Shafie-khah M, Catalão JP (2016) A new multi-objective solution approach to solve transmission congestion management problem of energy markets. *Applied Energy* 165:462–471, DOI 10.1016/j.apenergy.2015.12.101, URL <https://www.sciencedirect.com/science/article/pii/S0306261915016748> <https://linkinghub.elsevier.com/retrieve/pii/S0306261915016748>
- Houska B, Ferreau HJ, Diehl M (2011) ACADO toolkit-An open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods* 32(3):298–312, DOI 10.1002/oca.939, URL <http://doi.wiley.com/10.1002/oca.939>
- Huang HP, Jeng JC (2002) Monitoring and Assessment of Control Performance for Single Loop Systems. *Industrial & Engineering Chemistry Research* 41(5):1297–1309, DOI 10.1021/ie0101285, URL <https://pubs.acs.org/doi/10.1021/ie0101285>
- HUANG L, WANG N, ZHAO JH (2008) Multiobjective Optimization for Controller Design. *Acta Automatica Sinica* 34(4):472–477, DOI 10.3724/SP.J.1004.2008.00472, URL <https://linkinghub.elsevier.com/retrieve/pii/S1874102908600245>
- Ibrahim A, Rahnamayan S, Martin MV, Deb K (2016) 3D-RadVis: Visualization of Pareto front in many-objective optimization. In: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp 736–745, DOI 10.1109/CEC.2016.7743865, URL <http://ieeexplore.ieee.org/document/7743865/>
- Ierapetritou MG, Jia Z (2007) Short-term scheduling of chemical process including uncertainty. *Control Engineering Practice* 15(10 SPEC. ISS.):1207–1221, DOI 10.1016/j.conengprac.2006.10.009, URL <http://www.sciencedirect.com/science/article/pii/S0967066106001808>
- Jeppsson U (1996) Modelling Aspects of Wastewater Treatment Processes. PhD thesis, Lund Institute of Technology (LTH)
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, Perth, WA, Australia, vol 4, pp 1942–1948, DOI 10.1109/ICNN.1995.488968, URL <http://ieeexplore.ieee.org/document/488968/>
- Kravaris C, Daoutidis P (1990) Nonlinear state feedback control of second order nonminimum-phase nonlinear systems. *Computers Chem Eng* 14(4-5):439–449
- Kundu D, Suresh K, Ghosh S, Das S, Panigrahi B, Das S (2011) Multi-objective optimization with artificial weed colonies. *Information Sciences* 181(12):2441–2454, DOI 10.1016/j.ins.2010.09.026, URL <https://linkinghub.elsevier.com/retrieve/pii/S0020025510004809>

- Liu C, Gong Z, Joseph Lee HW, Teo KL (2019) Robust bi-objective optimal control of 1,3-propanediol microbial batch production process. *Journal of Process Control* 78:170–182, DOI 10.1016/J.JPROCONT.2018.10.001, URL <https://www.sciencedirect.com/science/article/pii/S0959152418303706>
- Logist F, Erdeghem PMMV, Impe JFV (2009) Efficient deterministic multiple objective optimal control of (bio)chemical processes. *Chemical Engineering Science* 64(11):2527–2538, DOI 10.1016/j.ces.2009.01.054
- Logist F, Vallerio M, Houska B, Diehl M, Van Impe J (2012) Multi-objective optimal control of chemical processes using ACADO toolkit. *Computers & Chemical Engineering* 37(0):191–199, DOI 10.1016/j.compchemeng.2011.11.002, URL <http://www.sciencedirect.com/science/article/pii/S0098135411003231> <https://linkinghub.elsevier.com/retrieve/pii/S0098135411003231>
- Longo S, D'Antoni BM, Bongards M, Chaparro A, Cronrath A, Fatone F, Lema JM, Mauricio-Iglesias M, Soares A, Hospido A (2016) Monitoring and diagnosis of energy consumption in wastewater treatment plants. A state of the art and proposals for improvement. *Applied Energy* 179:1251–1268, DOI 10.1016/j.apenergy.2016.07.043, URL <http://dx.doi.org/10.1016/j.apenergy.2016.07.043>
- López AM, Miller JA, Smith CL, Murrill PW (1967) Tuning Controllers with Error-Integral Criteria. *Instrumentation Technology* 14:57–62
- MacArthur R, Wilson E (1967) The theory of Island Biogeography, 1st edn. Princeton University Press, Princeton New Jersey
- Mahdavian M, Wattanapongsakorn N (2014) Multi-objective optimization of PID controller tuning for greenhouse lighting control system considering RTP in the smart grid. 2014 International Computer Science and Engineering Conference, ICSEC 2014 pp 57–61, DOI 10.1109/ICSEC.2014.6978129
- Marler RT, Arora JS (2010) The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization* 41(6):853–862, DOI 10.1007/s00158-009-0460-7, URL <http://link.springer.com/10.1007/s00158-009-0460-7>
- Marler RT, JS A (2004) Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 26(6):369–395, DOI 10.1007/s00158-003-0368-6
- Marlin TE (2015) Process Control Designing Processes and Control Systems for Dynamic Performance. McGraw-Hill Science/Engineering/Math
- Mehrabian A, Lucas C (2006a) A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics* 1(4):355–366, DOI 10.1016/j.ecoinf.2006.07.003, URL <https://linkinghub.elsevier.com/retrieve/pii/S1574954106000665>
- Mehrabian AR, Lucas C (2006b) A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics* 1(4):355–366, DOI 10.1016/j.ecoinf.2006.07.003
- Messac A, Puemi-Sukam C, Melachrinoudis E (2000) Aggregate Objective Functions and Pareto Frontiers: Required Relationships and Practical Implications. *Optimization and Engineering* 1(2):171–188, DOI 10.1023/A:1010035730904
- Messac A, Ismail-Yahaya A, Mattson C (2003) The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization* 25(2):86–98, DOI 10.1007/s00158-002-0276-1, URL <http://link.springer.com/10.1007/s00158-002-0276-1>
- Mitchell M (1995) Genetic algorithms: An overview. *Complexity* 1(1):31–39, DOI 10.1002/cplx.6130010108, URL <http://doi.wiley.com/10.1002/cplx.6130010108>
- Mittal P, Mitra K (2017) Decomposition based multi-objective optimization to simultaneously determine the number and the optimum locations of wind turbines in a wind farm. *IFAC-PapersOnLine* 50(1):159–164, DOI 10.1016/J.IFACOL.2017.08.027, URL <https://www.sciencedirect.com/science/article/pii/S2405896317300393>
- Moura D, Barcelos V, Samanamud GRL, França AB, Lofrano R, Loures CCA, Naves LLR, Amaral MS, Naves FL (2018) Normal boundary intersection applied as multivariate and multiobjective optimization in the treatment of amoxicillin synthetic solution. *Environmental Monitoring and Assessment* 190(3):140, DOI 10.1007/s10661-018-6523-8, URL <http://link.springer.com/10.1007/s10661-018-6523-8>

- Moya F, Rojas JD, Arrieta O (2017) Pareto-based polynomial tuning rule for 2DoF PID controllers for time-delayed dominant processes with robustness consideration. In: 2017 American Control Conference (ACC), IEEE, Seattle, pp 5282–5287, DOI 10.23919/ACC.2017.7963775, URL <http://ieeexplore.ieee.org/document/7963775/>
- Murril PW (1967) Automatic control of processes. International Textbook Co.
- Naves FL, de Paula TI, Balestrassi PP, Moreira Braga WL, Sawhney RS, de Paiva AP (2017) Multivariate Normal Boundary Intersection based on rotated factor scores: A multiobjective optimization method for methyl orange treatment. Journal of Cleaner Production 143:413–439, DOI 10.1016/J.JCLEPRO.2016.12.092, URL <https://www.sciencedirect.com/science/article/pii/S0959652616321564>
- Nejjari F, Dahhou B, Benhammou A, Roux G (1999) Non-linear multivariable adaptive control of an activated sludge wastewater treatment process. International Journal of Adaptive Control and Signal Processing 13(5):347–365, DOI 10.1002/(SICI)1099-1115(199908)13:5;347::AID-AC543_3.0.CO;2-8, URL <http://doi.wiley.com/10.1002/%28SICI%291099-1115%28199908%2913%3A5%3C347%3A%3AAID-AC543%3E3.0.CO%3B2-8>
- O'Dwyer A (2000) A summary of PI and PID controller tuning rules for processes with time delay. Part 1: PI controller tuning rules. In: Proceedings of PID'00: IFAC Workshop on Digital Control, Terrassa, Spain, pp 175–180
- O'Dwyer A (2009) Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London, UK
- Ogata K (1996) Modern control engineering; 3rd ed. Prentice-Hall electrical engineering, Prentice-Hall, Englewood Cliffs, NJ
- Olsson G, Newell B (1999) Wastewater Treatment Systems. Modelling, Diagnosis and Control, 1st edn. IWA Publishing, London, UK
- Pereira RBD, Leite RR, Alvim AC, de Paiva AP, Ferreira JR, Davim JP (2017) Multi-objective robust optimization of the sustainable helical milling process of the aluminum alloy Al 7075 using the augmented-enhanced normalized normal constraint method. Journal of Cleaner Production 152:474–496, DOI 10.1016/j.jclepro.2017.03.121, URL <https://www.sciencedirect.com/science/article/pii/S0959652617305656> <https://linkinghub.elsevier.com/retrieve/pii/S0959652617305656>
- Pierezan J, Ayala HH, Da Cruz LF, Freire RZ, Dos S Coelho L (2014) Improved multiobjective particle swarm optimization for designing PID controllers applied to robotic manipulator. IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CICA 2014: 2014 IEEE Symposium on Computational Intelligence in Control and Automation, Proceedings pp 1–8, DOI 10.1109/CICA.2014.7013255
- Ragazzini JR, Franklin GF (1958) Sampled-Data Control Systems. SMcGraw-Hill: New York
- Reynoso-Meza G, Garcia-Nieto S, Sanchis J, Blasco FX (2013) Controller Tuning by Means of Multi-Objective Optimization Algorithms: A Global Tuning Framework. Control Systems Technology, IEEE Transactions on 21(2):445–458, DOI 10.1109/TCST.2012.2185698
- Rivera DE, Morari M, Skogestad S (1986) Internal Model Control. 4. PID Controller Desing. Ind Eng Chem Des Dev 25:252–265
- Rojas JD, Valverde-Mendez D, Alfaro VM, Arrieta O, Vilanova R (2015) Comparison of multi-objective optimization methods for PI controllers tuning. In: 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), IEEE, pp 1–8, DOI 10.1109/ETFA.2015.7301410, URL <http://ieeexplore.ieee.org/document/7301410/>
- Roman C, Rosehart W (2006) Evenly distributed pareto points in multi-objective optimal power flow. Power Systems, IEEE Transactions on 21(2):1011–1012, DOI 10.1109/TPWRS.2006.873010
- Rovira A, Murrill P, Smith C (1969a) Tuning controllers for setpoint changes. Instruments and Control Systems 42:67–69
- Rovira AJA, Murrill PW, Smith CL (1969b) Tuning controllers for setpoint changes. Instruments and Control Systems 42:67–69

- Rubinstein RY, Kroese DP (2004) The Cross-Entropy Method, 1st edn. Information Science and Statistics, Springer New York, New York, NY, DOI 10.1007/978-1-4757-4321-0, URL <http://link.springer.com/10.1007/978-1-4757-4321-0>
- Rubio-Largo Á, Zhang Q, Vega-Rodríguez MA (2014) A multiobjective evolutionary algorithm based on decomposition with normal boundary intersection for traffic grooming in optical networks. *Information Sciences* 289(1):91–116, DOI 10.1016/j.ins.2014.08.004, URL <https://www.sciencedirect.com/science/article/pii/S0020025514007932> <https://linkinghub.elsevier.com/retrieve/pii/S0020025514007932>
- Sabina Sánchez H, Visioli A, Vilanova R (2017) Optimal Nash tuning rules for robust PID controllers. *Journal of the Franklin Institute* 354(10):3945–3970, DOI 10.1016/j.jfranklin.2017.03.012
- Sánchez HS, Padula F, Visioli A, Vilanova R (2017) Tuning rules for robust FOPID controllers based on multi-objective optimization with FOPDT models. *ISA Transactions* 66:344–361, DOI 10.1016/j.isatra.2016.09.021, URL <https://linkinghub.elsevier.com/retrieve/pii/S0019057816303652>
- Sanchis J, Martínez M, Blasco X, Salcedo JV (2008) A new perspective on multi-objective optimization by enhanced normalized normal constraint method. *Structural and Multidisciplinary Optimization* 36(5):537–546, DOI 10.1007/s00158-007-0185-4, URL <http://dx.doi.org/10.1007/s00158-007-0185-4>
- Seaman CM, Desrochers AA, List GF (1994) Multiobjective optimization of a plastic injection molding process. *Control Systems Technology, IEEE Transactions on* 2(3):157–168, DOI 10.1109/87.317974
- Sendín OH, Otero I, Alonso AA, Banga JR (2004) Multi-objective optimization for the design of bio-processes. In: Computer Aided Chemical Engineering, vol 18, Elsevier, pp 283–288, DOI 10.1016/S1570-7946(04)80113-5, URL <https://www.sciencedirect.com/science/article/pii/S1570794604801135> <https://linkinghub.elsevier.com/retrieve/pii/S1570794604801135>
- Shamsuzzoha M, Lee M (2008) Analytical design of enhanced PID filter controller for integrating and first order unstable processes with time delay. *Chemical Engineering Science* 63:2717–2731
- Shen JC (2002) New tuning method for PID controller. *ISA Transactions* 41(4):473–484, DOI 10.1016/S0019-0578(07)60103-7, URL <https://linkinghub.elsevier.com/retrieve/pii/S0019057807601037>
- Shi Y (2004) Particle swarm optimization. *IEEE Connections* 2(1):8–13
- Shinskey FG (1994) Feedback controllers for the process industries. McGraw-Hill Professional
- Shinskey FG (2002) Process Control: As Taught vs as Practiced. *Industrial & Engineering Chemistry Research* 41(16):3745–3750, DOI 10.1021/ie010645n, URL <https://pubs.acs.org/doi/10.1021/ie010645n>
- Shukla PK (2007) On the Normal Boundary Intersection Method for Generation of Efficient Front. *Computational Science - ICCS 2007* 4487:310–317, DOI 10.1007/978-3-540-72584-8_40, URL http://dx.doi.org/10.1007/978-3-540-72584-8_40
- Simab M, Javadi MS, Nezhad AE (2018) Multi-objective programming of pumped-hydro-thermal scheduling problem using normal boundary intersection and VIKOR. *Energy* 143:854–866, DOI 10.1016/j.energy.2017.09.144, URL <https://linkinghub.elsevier.com/retrieve/pii/S0360544217316651>
- Simon D (2008) Biogeography-Based Optimization. *IEEE Transactions on Evolutionary Computation* 12(6):702–713, DOI 10.1109/TEVC.2008.919004, URL <http://ieeexplore.ieee.org/document/4475427/>
- Simon D (2013) Evolutionary Optimization Algorithms, 1st edn. John Wiley and Sons
- Skogestad S (2003) Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13(4):291–309, DOI 10.1016/S0959-1524(02)00062-8, URL <https://www.sciencedirect.com/science/article/pii/S0959152402000628> <https://linkinghub.elsevier.com/retrieve/pii/S0959152402000628>

- Smith CA, Corripio AB (1985) Principles and practice of automatic process control. John Wiley and Sons, New York
- Stehr G, Graeb H, Antreich K (2003) Performance trade-off analysis of analog circuits by normal-boundary intersection. In: Proceedings of the 40th conference on Design automation - DAC '03, ACM Press, New York, New York, USA, p 958, DOI 10.1109/DAC.2003.1219159, URL <http://portal.acm.org/citation.cfm?doid=775832.776073>
- Tan MYH, Najafi AR, Pety SJ, White SR, Geubelle PH (2018) Multi-objective design of microvascular panels for battery cooling applications. *Applied Thermal Engineering* DOI 10.1016/j.applthermaleng.2018.02.028
- Tavakoli S, Tavakoli M (2003) Optimal Tuning of PID Controllers for First Order Plus Time Delay Models Using Dimensional Analysis. In: The Fourth International Conference on Control and Automation 2003 ICCA Final Program and Book of Abstracts ICCA-03, IEEE, pp 942–946, DOI 10.1109/ICCA.2003.1595161, URL <http://ieeexplore.ieee.org/document/1595161/>
- Tavakoli S, Griffin I, Fleming PJ (2005) Robust PI Controller for Load Disturbance Rejection and Setpoint Regulation. In: Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on, pp 1015–1020, DOI 10.1109/CCA.2005.1507263
- Tian Y, Wang Q, Wang Y, Jin Q (2014) A novel design method of multi-objective robust PID controller for industrial process. Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, ICIEA 2014 pp 242–246, DOI 10.1109/ICIEA.2014.6931166
- Toivonen HT, Totterman S (2006) Design of fixed-structure controllers with frequency-domain criteria: a multiobjective optimisation approach. *IEE Proc D, Control Theory and Applications* 153(1)
- Toro R, Ocampo-Martínez C, Logist F, Impe JV, Puig V (2011) Tuning of Predictive Controllers for Drinking Water Networked Systems. *IFAC Proceedings Volumes* 44(1):14,507–14,512, DOI 10.3182/20110828-6-IT-1002.00415, URL <https://www.sciencedirect.com/science/article/pii/S1474667016459598> <https://linkinghub.elsevier.com/retrieve/pii/S1474667016459598>
- Vahidinasab V, Javid S (2010) Normal boundary intersection method for suppliers' strategic bidding in electricity markets: An environmental/economic approach. *Energy Conversion and Management* 51(6):1111–1119, DOI 10.1016/j.enconman.2009.12.019
- Vallerio M, Van Impe J, Logist F (2014) Tuning of NMPC controllers via multi-objective optimisation. *Computers and Chemical Engineering* 61:38–50, DOI 10.1016/j.compchemeng.2013.10.003, URL <https://www.sciencedirect.com/science/article/pii/S0098135413003256> <https://linkinghub.elsevier.com/retrieve/pii/S0098135413003256>
- Vilanova R (2008) IMC based Robust PID design: Tuning guidelines and automatic tuning. *Journal of Process Control* 18:61–70
- Vilanova R, Visioli A (2012) PID Control in the Third Millennium. *Advances in Industrial Control*, Springer London, London, DOI 10.1007/978-1-4471-2425-2, URL <http://link.springer.com/10.1007/978-1-4471-2425-2>
- Vilanova R, Visioli A (2017) The Proportional-Integral-Derivative (PID) Controller. In: *Wiley Encyclopedia of Electrical and Electronics Engineering*, American Cancer Society, pp 1–15, DOI 10.1002/047134608X.W1033.pub2, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/047134608X.W1033.pub2>
- Vilanova R, Alfaro VM, Arrieta O (2011) Analytical Robust Tuning Approach for Two-Degree-of-Freedom {PI/PID} Controllers. *Engineering Letters (IAENG)* 19:204–214
- Visioli A (2001) Optimal tuning of PID controllers for integrating and unstable processes. *IEE Proc - Control Theory Appl* 148(1):180–184
- van de Vusse J (1964) Plug-flow type reactor versus tank reactor. *Chemical Engineering Science* 19(12):994–996, DOI 10.1016/0009-2509(64)85109-5, URL <https://linkinghub.elsevier.com/retrieve/pii/0009250964851095>
- Wang QG, Ye Z, Cai WJ, Hang CC (2008) PID Control for Multivariable Processes, Lecture Notes in Control and Information Sciences, vol 373. Springer Berlin Heidelberg, Berlin, Heidelberg, DOI 10.1007/978-3-540-78482-1, URL <http://link.springer.com/10.1007/978-3-540-78482-1>

- Zangeneh A, Jadid S (2007) Normal boundary intersection for generating Pareto set in distributed generation planning. In: 2007 International Power Engineering Conference (IPEC 2007), Singapore, pp 773–778
- Zhang D, Huang D, He Y (2004) Intelligent temperature control strate for LiTaO₃ thin film deposition process. In: Intelligent Mechatronics and Automation, 2004. Proceedings. 2004 International Conference on, pp 339–344, DOI 10.1109/ICIMA.2004.1384216
- Zhao H (2007) A multi-objective genetic programming approach to developing Pareto optimal decision trees. *Decision Support Systems* 43(3):809–826, DOI 10.1016/j.dss.2006.12.011, URL <https://www.sciencedirect.com/science/article/pii/S016792360600217X>
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1(1):32–49, DOI 10.1016/j.swevo.2011.03.001
- Zhou X, Zhou J, Yang C, Gui W (2018) Set-Point tracking and Multi-Objective Optimization-Based PID control for the goethite process. *IEEE Access* 6(2):36,683–36,698, DOI 10.1109/ACCESS.2018.2847641
- Zhuang M, Atherton DP (1993) Automatic Tuning of Optimum PID Controllers. *IEE Proc D, Control Theory and Applications* 140(3):216–224
- Ziegler J, Nichols N (1942) Optimum Settings for automatic controllers. *ASME Transactions* 64:759–768
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4):257–271, DOI 10.1109/4235.797969