José David Rojas, Orlando Arrieta, Ramon Vilanova

Industrial PID Controller Tuning

with a multi-objective framework using MATLAB®

February 15, 2019

Springer

Contents

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 [8].

The most used control algorithm in industry is the 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 [2] and represents an area of active study since the first tuning methodology was proposed in the 1940s [20]. In the case of this particular project, in order to have a direct impact on the industry and the research community in process control systems, it will focus on the problem of the tuning of the parameters of controllers PID.

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 multivariate multiobjective optimization problem. In the particular case of industrial PID controllers, this problem is also non-lineal 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 bee studied for several decades. For example, in [16] a type of MOO is used to tune PID controllers in a plastic injection molding process. In [1], 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 [14]. In [3], 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 [4] 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.

Beside bio-inspired methods for MOOP, there are several methdologies 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) [11]. With the WS method, the multiobjective cost function is transform in a one dimensional function using a weighted sum that give 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 [11]. 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 [5, 13, 10].

In order to obtain the Pareto front correctly, other methodologies have emerged that surpass the WS. The normal boundary intersection (NBI) method [6] consist in rewriting the optimization problem so that the feasible area is shortened by an equality constraint that depends on an extra parameter. 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 [7] 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) [12], 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 [15] and distributed generation planning [19], for the control of biochemical processes [9], circuit analysis [17], development of optimal supply strategies for the participants of oligopolistic energy markets [18].

Although in the area of process control, there are examples of the use of these algorithms [7], they do not take into account the different sources of disturbance to the system, but are different measures of performance to the same source of disturbance (the change in reference), and using a simple PID controller of a single degree of freedom.

References

[1] A. Abbas and P.E. Sawyer. “A multiobjective design algorithm: Application to the design of SISO control systems”. In: *Computers & Chemical Engineering* 19.2 (1995), pp. 241–248. ISSN: 0098-1354. DOI: 10.1016/0098-1354(94)00044-O.

[2] Karl Johan Åström and Tore Hägglund. *Advanced PID Control*. Research Triangle Park, NC 27709: ISA - The Instrumentation, Systems, and Automation Society, 2005.

[3] Aytekin Bagis. “Tabu search algorithm based PID controller tuning for desired system specifications”. In: *Journal of the Franklin Institute* 348.10 (2011), pp. 2795–2812. ISSN: 0016-0032. DOI: 10.1016/j.jfranklin.2011.09.001.

[4] Ibtissem Chiha, Noureddine Liouane, and Pierre Borne. “Tuning PID Controller Using Multiobjective Ant Colony Optimization”. In: *Applied Computational Intelligence and Soft Computing* 2012 (2012). Article ID 536326, pp. 1–7. DOI: 10.1155/2012/536326.

[5] Indraneel Das and John E. Dennis. “A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems”. In: *Structural and Multidisciplinary Optimization* 14 (1 1997), pp. 63–69. ISSN: 1615-147X. DOI: 10.1007/BF01197559.

[6] Indraneel Das and John E. Dennis. “Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems”. In: *SIAM Journal on Optimization* 8.3 (1998), pp. 631–657. DOI: 10.1137/S1052623496307510. eprint: http://epubs.siam.org/doi/pdf/10.1137/S1052623496307510.

[7] A. Gambier. “Optimal PID controller design using multiobjective Normal Boundary Intersection technique”. In: *Asian Control Conference, 2009. ASCC 2009. 7th*. Hong Kong, China, Aug. 2009, pp. 1369–1374.

[8] Olof Garpinger, Tore Hägglund, and Karl Johan Åström. “Criteria and Tradeoffs in PID Design”. In: *Proceedings of the IFAC Conference on Advances in PID Control*. Brescia, Italy, Mar. 2012.

[9] F. Logist, P.M.M. Van Erdeghem, and J.F. Van Impe. “Efficient deterministic multiple objective optimal control of (bio)chemical processes”. In: *Chemical Engineering Science* 64.11 (2009), pp. 2527–2538. ISSN: 0009-2509. DOI: 10.1016/j.ces.2009.01.054.

[10] R. Marler and Jasbir Arora. “The weighted sum method for multi-objective optimization: new insights”. In: *Structural and Multidisciplinary Optimization* 41 (6 2010), pp. 853–862. ISSN: 1615-147X. DOI: 10.1007/s00158-009-0460-7.

[11] R.T. Marler and J.S. Arora. “Survey of multi-objective optimization methods for engineering”. In: *Structural and Multidisciplinary Optimization* 26 (6 2004), pp. 369–395. ISSN: 1615-147X. DOI: 10.1007/s00158-003-0368-6.

[12] A. Messac, A. Ismail-Yahaya, and C.A. Mattson. "The normalized normal constraint method for generating the Pareto frontier". In: *Structural and Multidisciplinary Optimization* 25 (2 2003), pp. 86–98. ISSN: 1615-147X. DOI: 10.1007/s00158-002-0276-1.

[13] Achille Messac, Cyriaque Puemi-Sukam, and Emanuel Melachrinoudis. "Aggregate Objective Functions and Pareto Frontiers: Required Relationships and Practical Implications". In: *Optimization and Engineering* 1 (2 2000), pp. 171–188. ISSN: 1389-4420. DOI: 10.1023/A:1010035730904.

[14] G. Reynoso-Meza et al. "Controller Tuning by Means of Multi-Objective Optimization Algorithms: A Global Tuning Framework". In: *Control Systems Technology, IEEE Transactions on* 21.2 (2013), pp. 445–458. ISSN: 1063-6536. DOI: 10.1109/TCST.2012.2185698.

[15] C. Roman and W. Rosehart. "Evenly distributed pareto points in multiobjective optimal power flow". In: *Power Systems, IEEE Transactions on* 21.2 (May 2006), pp. 1011–1012. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2006.873010.

[16] C.M. Seaman, A.A. Desrochers, and G.F. List. "Multiobjective optimization of a plastic injection molding process". In: *Control Systems Technology, IEEE Transactions on* 2.3 (Sept. 1994), pp. 157–168. ISSN: 1063-6536. DOI: 10.1109/87.317974.

[17] G. Stehr, H. Graeb, and K. Antreich. "Performance trade-off analysis of analog circuits by normal-boundary intersection". In: *Design Automation Conference, 2003. Proceedings*. June 2003, pp. 958–963. DOI: 10.1109/DAC.2003.1219159.

[18] V. Vahidinasab and S. Jadid. "Normal boundary intersection method for suppliers' strategic bidding in electricity markets: An environmental/economic approach". In: *Energy Conversion and Management* 51.6 (2010), pp. 1111–1119. ISSN: 0196-8904. DOI: 10.1016/j.enconman.2009.12.019.

[19] A. Zangeneh and S. Jadid. "Normal boundary intersection for generating Pareto set in distributed generation planning". In: *Power Engineering Conference, 2007. IPEC 2007. International*. Dec. 2007, pp. 773–778.

[20] J.G. Ziegler and N.B. Nichols. "Optimum settings for automatic controllers". In: *Transactions of the ASME* November (1942), pp. 759–768.

Chapter 2
Process control as a multi-objective problem

Abstract In this chapter, the problem of the tuning of closed-loop control system is posed as a multi-objective optimization problem. The main variables and constraints are presented.

2.1 Characteristics of the controlled system

A feedback control system like the one shown in Figure 2.1, also called closed-loop

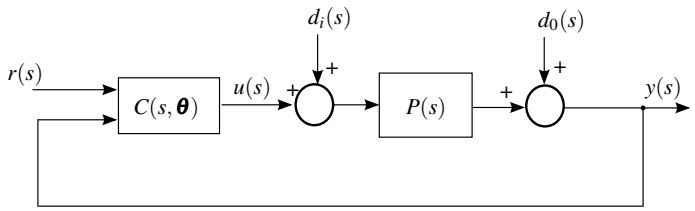

Fig. 2.1 Feedback control loop.

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 signal is used to compute the control signal $u(s)$ needed in order to achieve $y(s) = r(s)$.

In Figure. 2.1, $C(s,\boldsymbol{\theta})$ is the two degrees of freedom (2DoF) PID controller with parameters:

$$\boldsymbol{\theta} = \left[K_p \; T_i \; T_d \; \beta \right]^T$$

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:

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

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 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:

$$u(s) = C_r(s,\boldsymbol{\theta})r(s) - C_y(s,\boldsymbol{\theta})y(s), \tag{2.2}$$

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

$$C_r(s,\boldsymbol{\theta}) = K_p\left(\beta + \frac{1}{T_i s} + \gamma\frac{T_d s}{\alpha T_d s+1}\right), \tag{2.3}$$

and the part applied to the process output is:

$$C_y(s,\boldsymbol{\theta}) = K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s+1}\right). \tag{2.4}$$

It is common to set $\alpha = 0.1$ and $\gamma = 0$. For this reason, the controller parameter vector is give as $\boldsymbol{\theta} = [\, K_p \; T_i \; T_d \; \beta \,]^T$.

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:

$$M_s = \max_{\omega}\left\{\frac{1}{\left|1 + C_y(j\omega)P(j\omega)\right|}\right\} \tag{2.5}$$

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

2.2 Problem formulation

One of the objectives of the project was to develop a tuning rule based on the results of a MOO problem. As it is widely established, the controller tuning can be solved as a multi-objective optimization problem [4]. One common indicator of performance, is the integral of the absolute value of the error (IAE) given by:

$$J(\boldsymbol{\theta}) = \int_0^\infty |e(t,\boldsymbol{\theta})|\, dt. \tag{2.6}$$

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

$$e(t,\boldsymbol{\theta}) = r(t) - y(t,\boldsymbol{\theta}). \tag{2.7}$$

When (2.6) 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 $J_{do}(\boldsymbol{\theta})$. In general, it is not possible to optimize $\boldsymbol{\theta}$ for those three functions at the same time. Such impossible point where all the cost functions are optimal is called Utopia point, but optimizing one of them always produce a degradation in the other remaining functions. All the solutions that are closest to the utopia point, create the Pareto frontier. Interest reader are encouraged to go to Marler and Arora [5] for a good introduction in the subject.

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, 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. For this paper, only the case $M_{s,max} = 2.0$ is considered. The cost function then becomes:

$$\mathbf{J}(\boldsymbol{\theta}) = [J_{di}(\boldsymbol{\theta}), J_{do}(\boldsymbol{\theta}), J_r(\boldsymbol{\theta})]^T, \tag{2.8}$$

and solved by finding all possible optimal solutions of:

$$\begin{aligned} \mathbf{J}(\boldsymbol{\theta}^*) &= \min_{\boldsymbol{\theta}} \mathbf{J}(\boldsymbol{\theta}), \\ \text{s.t.} \quad & M_s \leq M_{s,max} \end{aligned} \tag{2.9}$$

This problem can be solved using different methods. Some authors have presented results where a bio-inspired methods have been applied in order to find the Pareto front of the problem (see for [8, 7, 6, 2] references). In this work, to avoid the stochastic nature of the results of those bio-inspired methods, the enhanced normalized normal constraint (ENNC) method was used to scalarize the problem and then use an standard mathematical version to solve the rewritten problem. For a study on the feasibility of the ENNC in PID tuning see Contreras-Leiva et al. [3], and for a comparison of using bio-inspired methods in PID tuning see Céspedes et al. [1].

References

[1] Macarena Céspedes et al. “A comparison of bio-inspired optimization methodologies applied to the tuning of industrial controllers”. In: *XXXVI Convención de Centroamérica y Panamá, CONCAPAN XXXVI*. San José, Costa Rica, Oct. 2016.

[2] Ibtissem Chiha, Noureddine Liouane, and Pierre Borne. “Tuning PID Controller Using Multiobjective Ant Colony Optimization”. In: *Applied Computational Intelligence and Soft Computing* 2012 (2012). Article ID 536326, pp. 1–7. DOI: 10.1155/2012/536326.

[3] Mónica P. Contreras-Leiva et al. “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*. Accepted. Sinaia, Romania, Oct. 2016.

[4] A. Gambier and E. Badreddin. “Multi-objective Optimal Control: An Overview”. In: *2007 IEEE International Conference on Control Applications*. Oct. 2007, pp. 170–175. DOI: 10.1109/CCA.2007.4389225.

[5] R.T. Marler and J.S. Arora. “Survey of multi-objective optimization methods for engineering”. In: *Structural and Multidisciplinary Optimization* 26 (6 2004), pp. 369–395. ISSN: 1615-147X. DOI: 10.1007/s00158-003-0368-6.

[6] G. Reynoso-Meza et al. “Controller Tuning by Means of Multi-Objective Optimization Algorithms: A Global Tuning Framework”. In: *Control Systems Technology, IEEE Transactions on* 21.2 (2013), pp. 445–458. ISSN: 1063-6536. DOI: 10.1109/TCST.2012.2185698.

[7] Gilberto Reynoso-Meza et al. “Multiobjective evolutionary algorithms for multivariable PI controller design”. In: *Expert Systems with Applications* 39.9 (2012), pp. 7895–7907. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2012.01.111.

[8] M.M. Sayed et al. “A novel method for PID tuning using a modified biogeography-based optimization algorithm”. In: *Control and Decision Conference (CCDC), 2012 24th Chinese* 1647 (2012), pp. 23–25.

Chapter 3
Multi-objective optimization methods

3.1 Multi-objective optimization problem formulation

A 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 function do not necessarily optimize the other cost functions.

In those cases, given a set of cost functions:

$$\mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), F_3(\mathbf{x}), \ldots, F_k(\mathbf{x})]^T \tag{3.1}$$

that depends on n different variables $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, $x_i \in \mathbf{X}$, where $\mathbf{X}$ is the feasible decision space. The MOOP may be formulated as follows [3]:

$$\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}), \tag{3.2a}$$

s.t.

$$g_j(\mathbf{x}) \leq 0, \qquad j = 1, 2, \ldots, m \tag{3.2b}$$

$$h_l(\mathbf{x}) = 0, \qquad l = 1, 2, \ldots, e \tag{3.2c}$$

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

In general, it is not possible to find a set of variables values that minimizes all F functions. In fact, the optimization problem in (3.2) have multiple equally optimal solutions in the sense of the Pareto optimality. According to [3]:

> "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 3.1 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

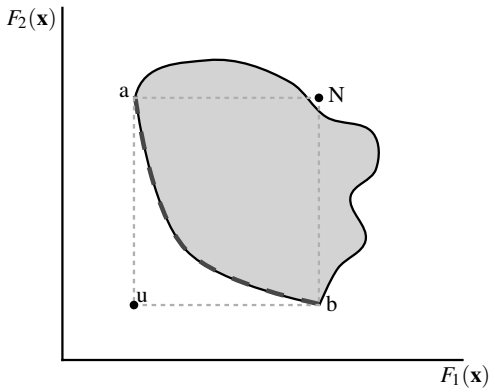

Fig. 3.1 Function space

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 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 (3.2) 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 3.1, 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.

3.2 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.

3.2.1 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 [3]. For example, if the two objectives to optimize are $f_1(\mathbf{x})$ and $f_1(\mathbf{x})$, the new function is given by:

$$F_{WS}(\mathbf{x}) = \alpha_{1WS}\hat{f}_1(\mathbf{x}) + \alpha_{2WS}\hat{f}_2(\mathbf{x}), \tag{3.3}$$

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 [3]) is given by:

$$\hat{f}_1(\mathbf{x}) = \frac{f_1(\mathbf{x}) - \min(f_1(\mathbf{x}))}{\max(f_1(\mathbf{x})) - \min(f_1(\mathbf{x}))}. \tag{3.4}$$

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:

$$\begin{aligned} &\min_{\mathbf{x}} F_{WS}(\mathbf{x}), \\ &\text{s.t. } h(\mathbf{x}) = 0, \\ &g(\mathbf{x}) \leq 0, \end{aligned} \tag{3.5}$$

where $h(\mathbf{x})$ and $g(\mathbf{x})$ are the equality and inequality constraints of the original problem.

There are certain drawbacks with this approach. In first place, when (3.3) 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 [1].

3.2.2 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 obtained an even spaced Pareto front [2]. In figure 3.2, 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 joints the anchor points) is parametrized 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 ith anchor point, $\mathbf{F}(\mathbf{x}^*)$ is the value of the function at the utopia point, and β is chosen as:

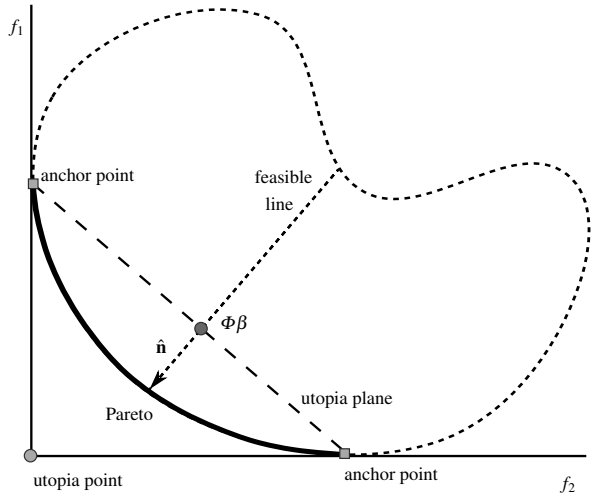

Fig. 3.2 NBI optimization method.

$$\beta = \begin{bmatrix} \alpha_{1NBI} \\ \alpha_{2NBI} \end{bmatrix}, \tag{3.6}$$

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

Then, the 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 pseudo normal as proposed in [2]) to the utopia plane. In other words, the objective is to find the border of the feasible region that is closer to the utopia point, but, by varying the parameter β in a systematic way, it is possible to obtain an even spaced realization of the Pareto front.

The problem then is posed as follows:

$$\begin{aligned} &\max_{\mathbf{x},v} \; v, \\ &\text{s.t. } \Phi\boldsymbol{\beta} + v\hat{\mathbf{n}} = \mathbf{F}(\mathbf{x}), \\ &\qquad h(\mathbf{x}) = 0, \\ &\qquad g(\mathbf{x}) \leq 0. \end{aligned} \tag{3.7}$$

In practice, the NBI method adds an equality constraint to the problem in such a way that, by maximizing a new variable v (which represents the distance from the utopia plane towards the utopia point), the border that is closer to the utopia point is found. Depending on the shape of the frontier, it is possible that the NBI finds points that are not Pareto optimal but belong to the edge of the feasible space. These points can be useful to have a better idea of the convexity (or lack thereof) of the Pareto front.

3.2.3 Normalized Normal Constraint

The NNC is presented in [4] 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 3.3: the utopia plane is parameterized in a similar way as the NBI but,

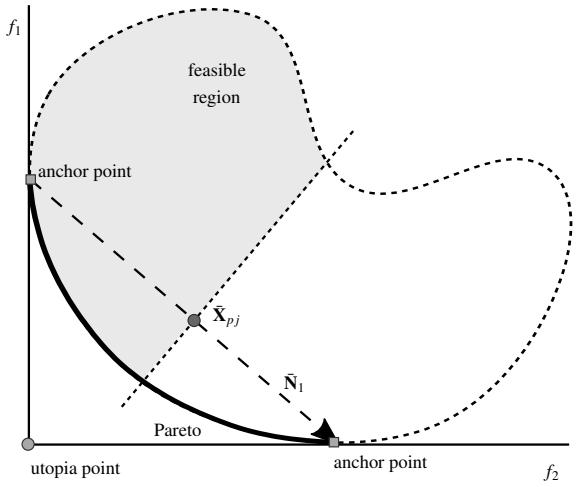

Fig. 3.3 NNC optimization method.

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 normal 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

$$\bar{\mathbf{X}}_{pj} = \alpha_{1NNC}\hat{\mathbf{F}}(\mathbf{x}_1^*) + \alpha_{2NNC}\hat{\mathbf{F}}(\mathbf{x}_2^*). \tag{3.8}$$

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:

$$\begin{aligned} &\min_{\mathbf{x}} \ \hat{f}_1(\mathbf{x}), \\ &\text{s.t. } \bar{\mathbf{N}}_1^T \left(\hat{\mathbf{F}}(\mathbf{x}) - \bar{\mathbf{X}}_{pj}\right) \leq 0, \\ &h(\mathbf{x}) = 0, \\ &g(\mathbf{x}) \leq 0, \end{aligned} \tag{3.9}$$

where $\bar{\mathbf{N}}_1$ is the vector that contains the direction of the utopia plane.

3.2.4 Enhanced Normalized Normal Constraint

The ENNC [5], 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:

$$F_i^{**} = \left[f_1^N \; f_2^N \; \cdots \; f_i^* \; \cdots \; f_n^N \right], \tag{3.10}$$

where f_i^N if 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.

References

[1] Indraneel Das and John E. Dennis. “A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems”. In: *Structural and Multidisciplinary Optimization* 14 (1 1997), pp. 63–69. ISSN: 1615-147X. DOI: 10.1007/BF01197559.

[2] Indraneel Das and John E. Dennis. “Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems”. In: *SIAM Journal on Optimization* 8.3 (1998), pp. 631–657. DOI: 10.1137/S1052623496307510. eprint: http://epubs.siam.org/doi/pdf/10.1137/S1052623496307510.

[3] R.T. Marler and J.S. Arora. “Survey of multi-objective optimization methods for engineering”. In: *Structural and Multidisciplinary Optimization* 26 (6 2004), pp. 369–395. ISSN: 1615-147X. DOI: 10.1007/s00158-003-0368-6.

[4] A. Messac, A. Ismail-Yahaya, and C.A. Mattson. “The normalized normal constraint method for generating the Pareto frontier”. In: *Structural and Multidisciplinary Optimization* 25 (2 2003), pp. 86–98. ISSN: 1615-147X. DOI: 10.1007/s00158-002-0276-1.

[5] J. Sanchis et al. “A new perspective on multiobjective optimization by enhanced normalized normal constraint method”. English. In: *Structural and Multidisciplinary Optimization* 36.5 (2008), pp. 537–546. ISSN: 1615-147X. DOI: 10.1007/s00158-007-0185-4. URL: http://dx.doi.org/10.1007/s00158-007-0185-4.

Chapter 4
Implementation of the multi-objective optimization methods using MATLAB®

Writing in process

Chapter 5
Application examples

Abstract In this chapter, the methods presented in the book applied to some common industrial proceses. It has been divided in three parts: the results of the application of the methodology to different processes, the tuning rule derived from the data that was obtained when the ENNC was applied to the optimization of the parameters of a 2DoF PID controlling a generalized ODSOPTD.

5.1 Comparison of different PID tuning for a benchmark problem

In order to test the capabilities of a 2DoF PID controller optimized using the ENNC method, a benchmark process presented in [5] is used for comparison. It is a fourth order plant given by:

$$P(s) = \frac{1}{\prod_{n=0}^{n=3}(0.5^n s+1)}. \tag{5.1}$$

A low order model can be found in order to design a suitable PID controller for the full order plant. Using the 123c method [3], the obtained model is given by:

$$F(s) = \frac{e^{-0.297s}}{(0.9477s+1)(0.6346s+1)} \tag{5.2}$$

In Figure 5.1, the Pareto frontier obtained for the model (5.2) is presented, the curve was obtained with the ENNC optimization of the cost functions J_r and J_{di}, and the optimal parameters of PID controller with two degrees of freedom were obtained.

It is interesting to note that, in order to improve the response of J_{di}, J_r has to be augmented (worsening the regulator response), however the degradation is not as much as the improvement in the J_d function. This is a clear example of one of the many advantages of using a multi-objective framework for controller tuning.

The optimal parameters for the 2DoF PID controller, for optimum tuning in disturbance rejection (J_{di}) and optimal servo control (J_r), are presented in in Table 5.1,

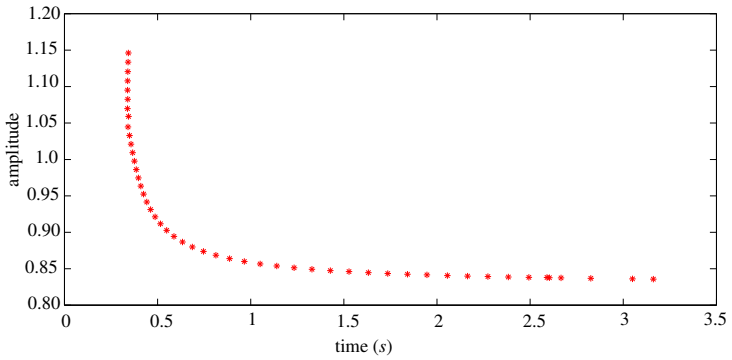

Fig. 5.1 The Pareto frontier for the benchmark process

along with two other tunings: the ART_2 method [7] and the $uSORT_2$ [4], to compare the performance of the closed loop control. It is important to clarify that these two tuning are just the extreme points of the Pareto front, thanks to the ENNC method, the control engineer is able to select any closed-loop response between these two extremes, all of them equally optimal and robust.

In Fig. 5.2, the response of the control loop with the four different tunings is presented, for a change in the reference value. The best system response is given by the ENNC optimal controller J_r, emphasizing its characteristics of less IAE, as shown in Table 5.2.

The optimal controller for disturbance rejection is presented in Fig. 5.3. Also it is clear how the loop response with optimal controller for changes in reference value is the worst for disturbance rejection.

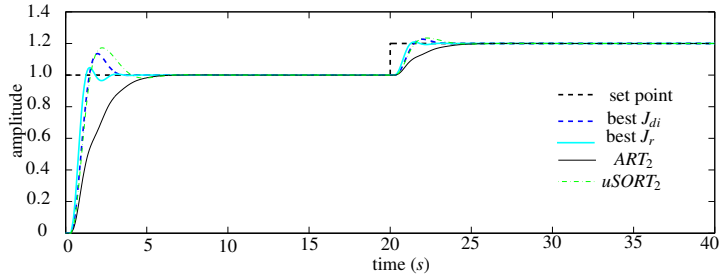

Fig. 5.2 Optimal response of the control system J_r

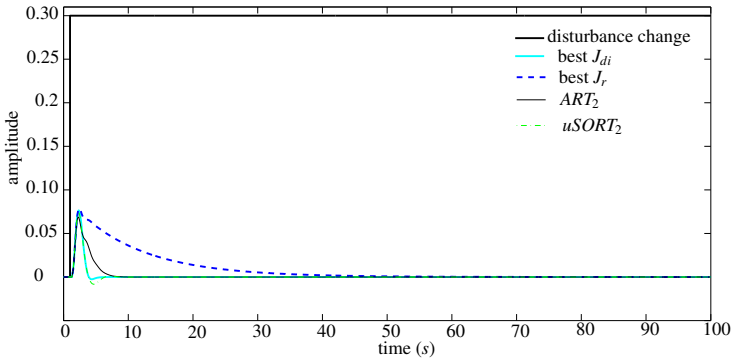

Fig. 5.3 Optimal response of the control system J_{di}

Table 5.1 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

Table 5.2 Servo performance response

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

5.2 MOOP Formulation of PID Control for $LiTaO_3$ Thin Film Deposition Process

According to [8], the temperature control is a key factor in the deposition process of lithium tantalate ($LiTaO_3$) by means of metal organic chemical vapor deposition (MOCVD). 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 follow a predefined temperature profile accurately (servo control) while been able to reject other disturbances (regulatory control).

The model of the MOCVD chamber can be given by:

$$G(s) = \frac{Ke^{-Ls}}{Ts+1}, \tag{5.3}$$

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

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 5.4. In this case, the

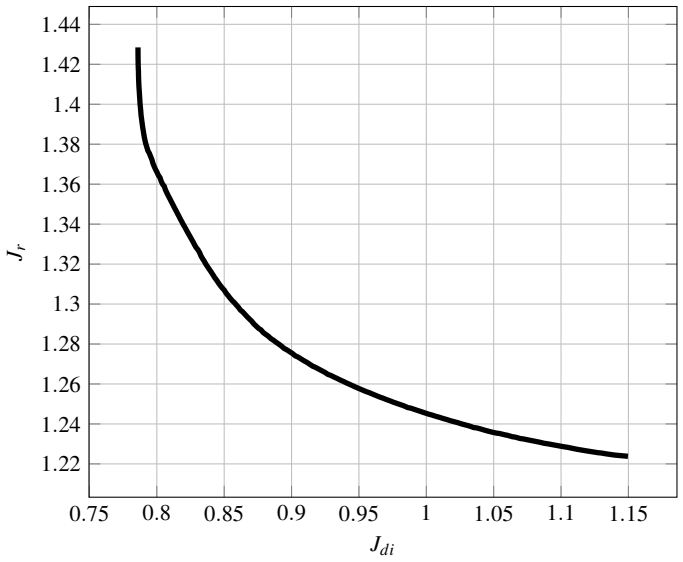

Fig. 5.4 Pareto front for the $LiTaO_3$ thin film deposition process.

Pareto front is fairly convex.

The lowest value for J_{di} is 0.78602. If the control engineer allows J_{di} to be degraded to a value of $J_{di} = 0.80012$, it represents a degradation of 1.8%, however, it represents an improvement of 4.37% for J_r. Since the Pareto is not in general, a straight line in the function space, an improvement in one of the functions does not necessarily represent an equal degradation in the other function. How much is it viable to degrade a function in favor of other functions depends entirely on the control engineer, an the Pareto front can be an excellent tool for the decision process.

In the particular case of figure 5.4, the Pareto front contains 200 different controllers. To help the control engineer to decide the tuning for the given process, it may be useful to plot the variation of the controller parameters as a function of the degradation of the most important cost function. For example, one may argue that, for the $LiTaO_3$ thin film deposition process, the most important operation of the controller is to follow the temperature setpoint profile, therefore, one may consider J_r as the main function. If that is the case, the variation of the controller gain K_p is presented in figure 5.5 where m is defined as the normalized degradation of J_r. It is clear that the value of K_p does not vary much from a controller optimal for servo action ($m = 0\%$) to a controller optimal for regulatory action ($m = 100\%$). This small variation is also related to keeping a robustness of $M_S = 2$ for all possi-

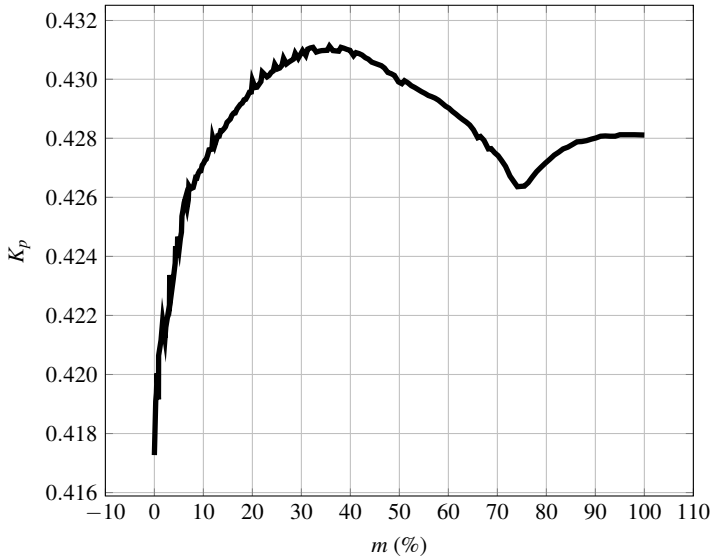

Fig. 5.5 K_p variation for the $LiTaO_3$ thin film deposition process as a function of J_r degradation.

ble controllers. When this restriction is not taken into account, the variation of the tuning parameters may be larger.

For the case of the integral time T_i, the variation is as given in figure 5.6. In this

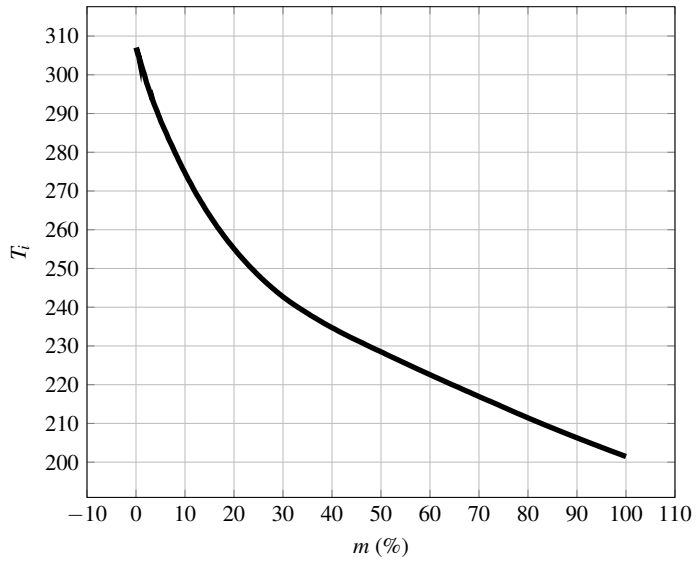

Fig. 5.6 T_i variation for the $LiTaO_3$ thin film deposition process as a function of J_r degradation.

case, it is clear that the value of T_i is highly dependent of the value of J_r degradation.

The case for the derivative time T_d is given in figure 5.7 and the variation of the

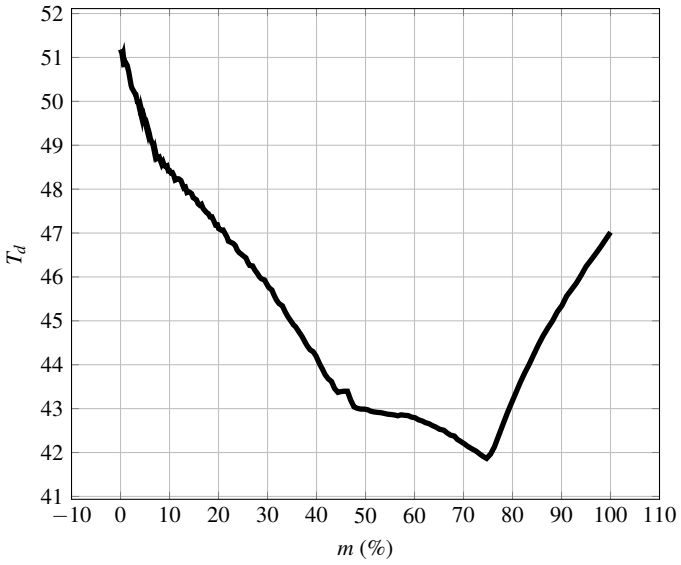

Fig. 5.7 T_i variation for the $LiTaO_3$ thin film deposition process as a function of J_r degradation.

setpoint weight factor β is presented in figure 5.8.

From this point on, one may take two different paths: find equations of the variation of the parameters or use the parameters data as a database in a software tool.

Using the data obtained from the 200 optimizations it may be possible to find a model of the variation of each parameter as a function of the desired degradation. In the case of K_p, the variation is so small, that it may be easier to take the average value and apply it to all possible controller tuning. On the other hand, for T_i and β may be possible to obtained a piecewise linear function for each parameter. However, as it can be seen, the relationship between T_d and the degradation of J_r is quite complex. One may try to approximate a high order polynomial or a non-linear function in order to obtain a good mathematical description of the behavior.

If the function for each parameter became too complex, it may be easier to take the Pareto front as a database, in order to interpolate all possible allowed degradation values of J_r. This is easily achievable in any spreadsheet software or using a programming language as MATLAB®.

The simulation of the response of the controlled system to a step change in the reference is presented in figure 5.9 for the servo response and in figure 5.10 for the regulation response. In both cases, the responses of the two extreme cases of the Pareto are compared with the $uSORT_2$ tuning rule [1].

For the reference tracking in figure 5.9, it is clear that the best servo response taken from the Pareto is better than any other controller with the constraint of having

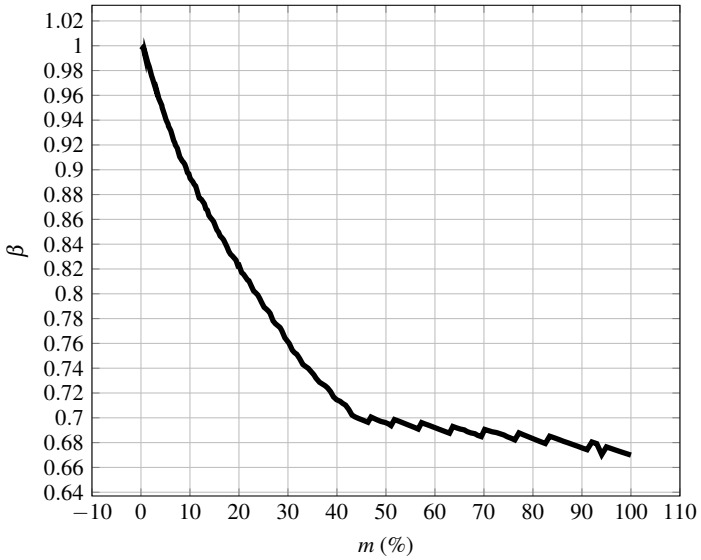

Fig. 5.8 β variation for the $LiTaO_3$ thin film deposition process as a function of J_r degradation.

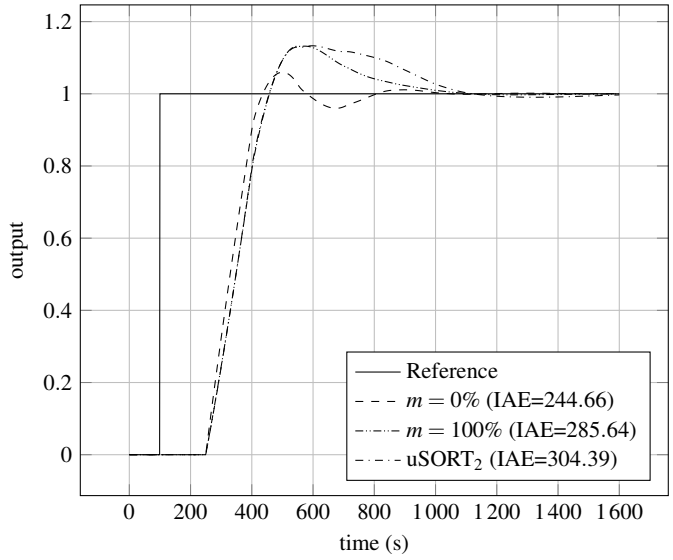

Fig. 5.9 Servo response of the $LiTaO_3$ thin film deposition process with tree different tuning.

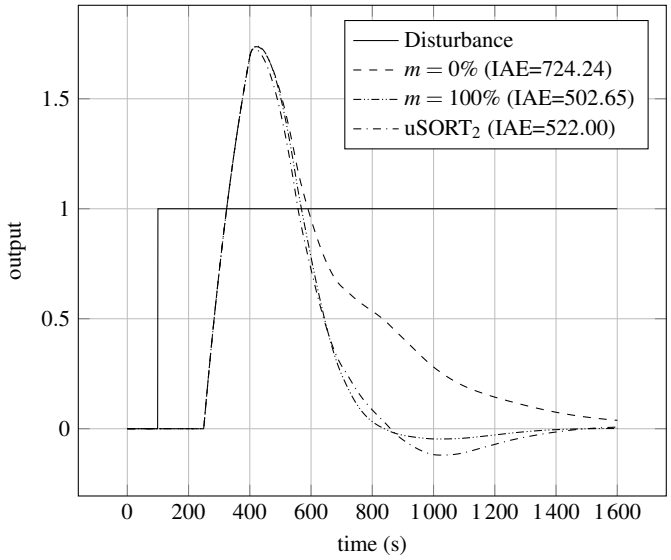

Fig. 5.10 Regulation response of the $LiTaO_3$ thin film deposition process with tree different tuning.

a maximum sensitivity of $M_{S,max} = 2$. The uSORT$_2$ method is intended primarily as a regulation tuning with a given fixed M_S, and therefore, it is not surprising that this method has a higher IAE.

In the case of the disturbance rejection case, the best tuning is the one given by the best regulator of the Pareto front. The response of this controller is much better than the response given by the best servo controller. When comparing this response with the given by the uSORT$_2$ method, the response is quite similar, since the regulation response of the uSORT$_2$ controller is just 3.85% higher.

However, it is clear that, from the information obtained with the Pareto front, one may be able to find more than just two controller tunings. In fact, one may find all possible optimal controllers in the IAE sense within these two extremes. For example in figure 5.11, five of the possible tuning on the Pareto are presented from $m = 100\%$ to $m = 0\%$ for the reference tracking response. In all cases $M_{S,max} \leq 2.0$. As it can be seen, one of the main advantages of finding the Pareto front is to be able to choose the desired response from an equally optimal set of controller parameters. In figure 5.12, the regulator response to the same controllers taken from the Pareto front is presented. The task of selecting which controller best fits the requisites of the controlled system depends entirely to the control engineer. However, initial choice may be the controller that is closer to the utopia point in the normalized function space.

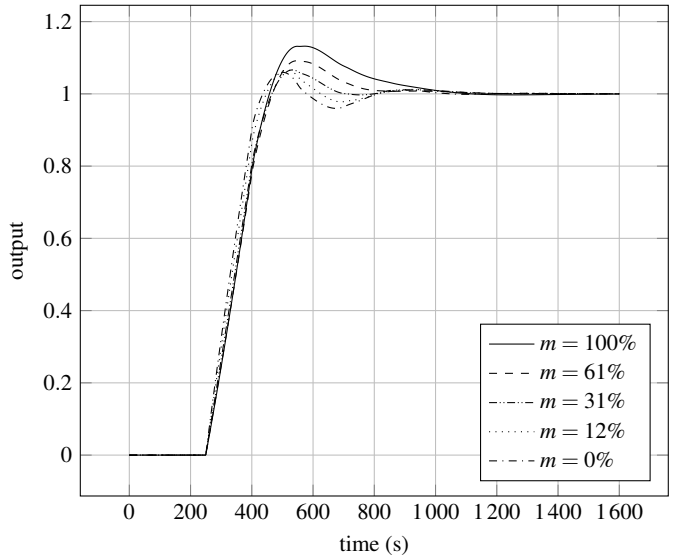

Fig. 5.11 Servo response of the $LiTaO_3$ thin film deposition process varying the tuning across the Pareto front with different degradations.

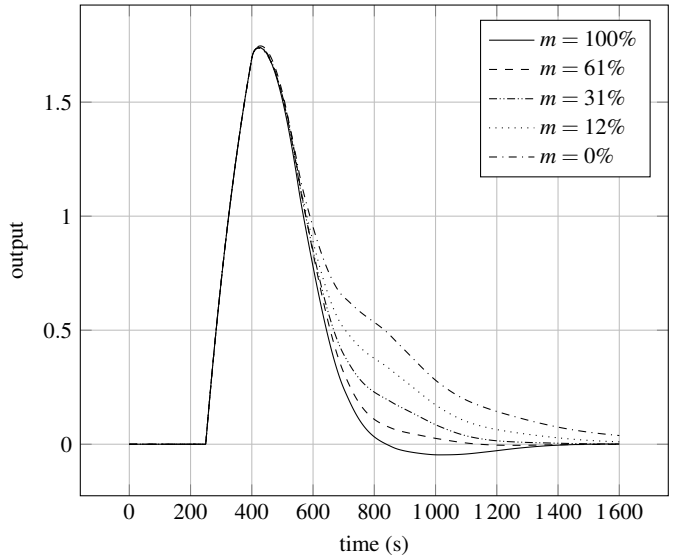

Fig. 5.12 Regulator response of the $LiTaO_3$ thin film deposition process varying the tuning across the Pareto front with different degradations.

5.3 2DoF-PID optimized tuning for a non-linear CSRT process

Continuous stirred tanks reactors (CSTR) are one of the most common sub-system in the chemical process field. Depending on the reaction, the dynamic of this plant can be highly non-linear, however, it is common to operate them in a given operation point with the controllers tuned for disturbances rejection.

Consider the CSTR non-linear model based on [6] and references therein:

$$\dot{x}_1 = -x_1 + D_a(1-x_1)e^{\frac{x_2}{1+x_2/\varphi}}, \tag{5.4}$$

$$\dot{x}_2 = -(1+\delta)x_2 + BD_a(1-x_1)e^{\frac{x_2}{1+x_2/\varphi}} + \delta u, \tag{5.5}$$

$$y = x_2, \tag{5.6}$$

where x_1 and x_2 represents dimensionless reactant concentration and reactor temperature, u is the dimensionless cooling jacket temperature which is considered as the control input, y is the output of the system, $D_a = 0.072$ is the Damköhler number, $\varphi = 20$ is the activated energy, $B = 8$ is the heat of the reaction and $\delta = 0.05$ is the heat transfer coefficient. The system is considered to be controlled from the equilibrium point $x_1 = 0.931$ and $x_2 = 7.095$ with input $u = 0$.

In order to find a suitable PID controller for this plant, it is necessary to find a low order linear model. After performing a step change of amplitude $\Delta u = 10$ in the input, the model found is given by:

$$F(s) = \frac{0.061e^{-0.0037}}{(0.9213s+1)(0.0056s+1)}. \tag{5.7}$$

This model fits the plant dynamics at the operating point very well, as can be

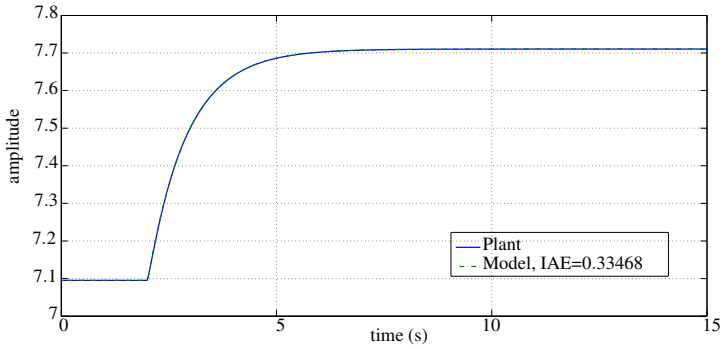

Fig. 5.13 Comparison between the nonlinear response of the CSTR and the linear model.

seen in Fig. 5.13, where the IAE between the output of the plant and the output of the model is 0.33468.

In Fig. 5.14, the Pareto frontier obtained using the CSTR model is showed. Again, the curve was obtained with the ENNC optimization method with the cost functions J_r and J_{di}.

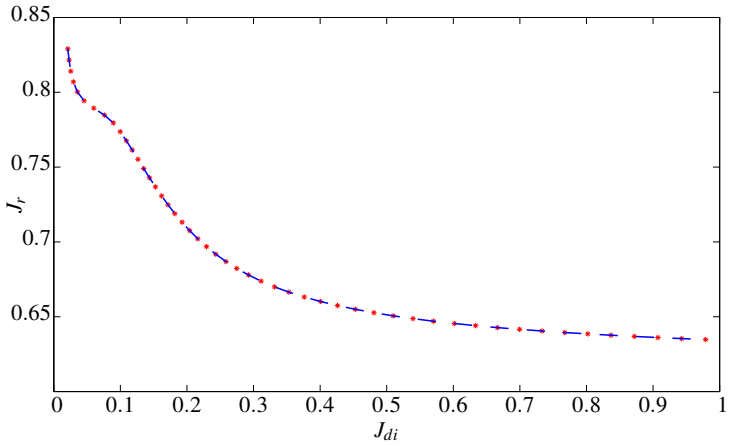

Fig. 5.14 The Pareto frontier for the CSTR process

The tuning parameters for a 2DoF PID controller optimized for this process, are presented in Table 5.3. The response obtained for an optimum tuning for changes in

Table 5.3 2DoF PID controller, for the CSRT.

Tuning	K_p	T_i	T_d	β
optimum J_{di}	10	0.1	0.1	1.9406
optimum J_r	10	10	0.1	2.5759

the reference value is showed in Fig. 5.15, while the obtained performance values are presented in Table 5.4. While the tuning was computed using the linear low order

Table 5.4 Performance rates with optimal tuning J_r, for the CSRT.

Tuning	IAE	M_s
optimum J_r	0.4547	2
optimum J_{di}	0.5835	2

model, the simulation presented in this paper was performed using the non-linear model of the CSTR. It is clear how the performance of the best J_r tune overcomes the closed loop response of the best J_{di} tune, presenting a minor IAE index.

In Table 5.5, the corresponding values of IAE for the servo and regulator re-

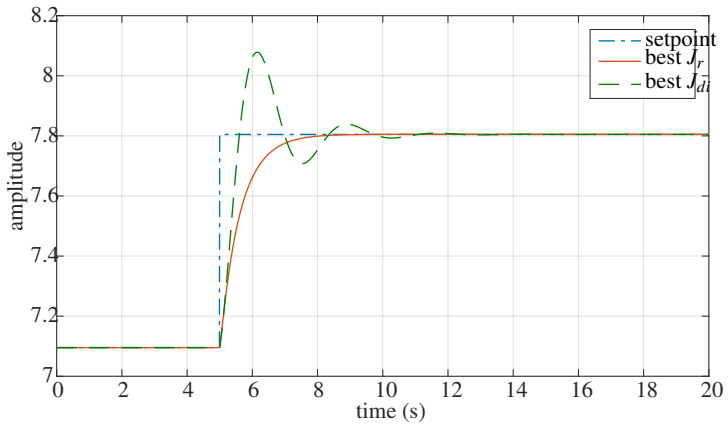

Fig. 5.15 Setpoint tracking for the CSRT system.

Table 5.5 Performance rates with optimal tuning J_{di}, for the CSRT.

Tuning	IAE	M_s
optimum J_r	4.9390	2
optimum J_{di}	0.1108	2

sponses are presented. Note the differences between J_{di} and J_r in tables 5.4 and 5.5. The obtained Pareto front shows that considerable changes in J_{di} does not correspond to large degradation in J_r. This is consistent with the simulation results shown, and therefore, one can initially tune the controller for optimal J_{di}, knowing that the servo response is not highly degraded, while also guaranteeing a safe robustness value. If a better servo response is needed, it can be found by picking other tuning that also belongs to the Pareto front, but with the knowledge on how this decision affects the regulator response.

5.4 Tuning of PID controllers using a MOO approach

The first step to find a suitable tuning rule for 2DoF PID controllers for ODSOPTD plants in a Pareto-optimal sense, is to gather the data. These data was found by applying the ENNC methodology to problem (2.9), using a normalized version of the plants. Defining the normalized variable $\hat{s} = Ts$, then the normalized version of the time delay, τ_0, becomes [2]:

$$\tau_0 = \frac{L}{T}, \tag{5.8}$$

which allows to find the optimal tuning for a complete family of plants rather than a single plant only. The cases that were computed contained a values from 0 to 1 in 0.1 steps and τ_0 values from 1 to 2 in 0.1 steps and $M_{s,max} = 2$. A value of

$\tau_0 = 1$ represents a plant where the time delay is equal to the largest lag time of the system, while in the case of $\tau_0 = 2$, the time delay is twice the largest lag time. The corresponding normalized PID parameters become:

$$\kappa_p = K_p K, \qquad \tau_i = \frac{T_i}{T}, \qquad \tau_d = \frac{T_d}{T}. \tag{5.9}$$

The Pareto front was found for each plant family with around 1000 points each one. Since each point in the Pareto Front represents a different tuning, the number of possible Pareto optimal PID controllers found, count to approximately 100 000[1].

Once the data is found, the next step is to perform a curve fitting procedure in order to find appropriate equations that match the pattern of the different values of κ_p, τ_i and τ_d as a function of a and τ_0.

In order to determine the PID controller parameters equations for κ_p, τ_i, τ_d and β, a curve fitting procedure was applied. In order to parameterize the tuning rule, the idea of "allowed degradation" is introduced. When J_{di} and J_{do} are normalized as:

$$\delta = \frac{J_{di}(\theta) - J_{di,min}(\theta)}{J_{di,max}(\theta) - J_{di,min}(\theta)}, \tag{5.10}$$

$$\gamma = \frac{J_{do}(\theta) - J_{do,min}(\theta)}{J_{do,max}(\theta) - J_{do,min}(\theta)}, \tag{5.11}$$

where both $0 \leq \delta \leq 1$ and $0 \leq \gamma \leq 1$, then, a value of $\delta = 1$ represents an allowed degradation of 100% of the response of the controlled system to an input disturbance. Since the data was obtained from the Pareto front of three functions, it is only required to choose the degradation of two functions. If the allowed degradation for both J_{di} and J_{do} are set to $\delta = \gamma = 1$, the resulting tuning is expected to represent the optimal tuning for servo control.

Almost two hundred and twenty regressions were run for all considered values of a and τ_0 for every parameter in $\boldsymbol{\theta}$. Regression analysis revealed that a second order fit was the most appropriate; as it had an average adjusted R-square of 0.92494, 0.97997 and 0.92275 for κ_p, τ_i and τ_d, respectively. First order fits had a poorer adjusted R-square, compared to a second order fit adjusted R-square; except for β, which was possible to model as a first order function of γ and δ, with an average adjusted R-square of 0.96063. Some results for the corresponding fits of κ_p, τ_i, τ_d and β, is shown in Fig. 5.16.

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

[1] The data base contain other cases beyond the scope of this paper and can be downloaded from `http://www2.eie.ucr.ac.cr/~jdrojas/Research/MOOP_PIDTuning/`

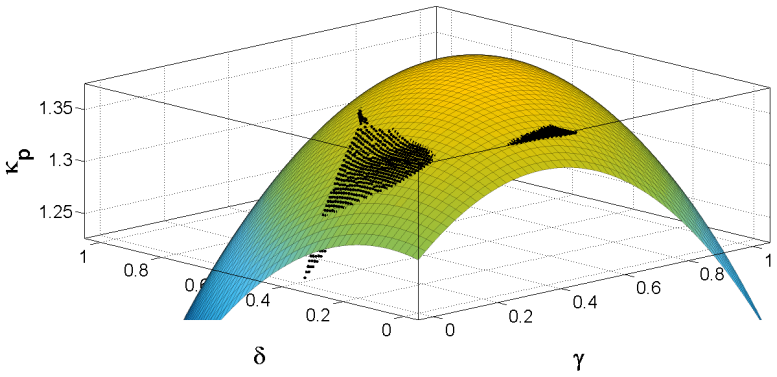

Fig. 5.16 Second order fit for κ_p when $a = 0.1$ and $t_0 = 1$

$$\kappa_p = p_{00} + p_{01} \cdot \gamma + p_{02} \cdot \delta + p_{03} \cdot \gamma^2 + p_{04} \cdot \gamma \cdot \delta + p_{05} \cdot \delta^2, \tag{5.12}$$

$$\tau_i = p_{10} + p_{11} \cdot \gamma + p_{12} \cdot \delta + p_{13} \cdot \gamma^2 + p_{14} \cdot \gamma \cdot \delta + p_{15} \cdot \delta^2, \tag{5.13}$$

$$\tau_d = p_{20} + p_{21} \cdot \gamma + p_{22} \cdot \delta + p_{23} \cdot \gamma^2 + p_{24} \cdot \gamma \cdot \delta + p_{25} \cdot \delta^2, \tag{5.14}$$

$$\beta = p_{30} + p_{31} \cdot \gamma + p_{32} \cdot \delta, \tag{5.15}$$

The coefficients p_{ij}, where $i = \{0,1,2,3\}$ and $j = \{0,1,2,3,4,5\}$, all vary according to a and τ_0. This dependency suggested to perform another regressions over these parameters in terms of a and τ_0. Therefore a curve fitting procedure was also carried out for every p_{ij}. As an example of these regressions, Fig. 5.17 shows the

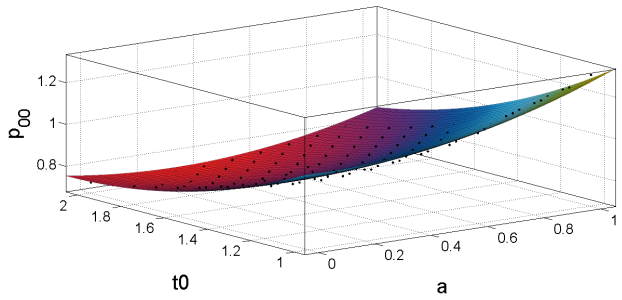

Fig. 5.17 Second order fit for p_{00} in κ_p

result for the p_{00} parameter as a function of a and τ_0. For time-delay dominant pro-

cess, the values for τ_0 were considered to be in the range such that $1 \leq \tau_0 \leq 2$. In these cases, the adjusted R-square was higher than 0.9. It may be possible to find a tuning equation for values of τ_0 from 0.1 to 2 , however, this equations may become too complex to achieve a high enough adjusted R-square. This analysis is beyond the scope of this article but will be reported elsewhere.

The most appropriate fit for every coefficient in the range of $1 \leq \tau_0 \leq 2$, was also a second order fit. The resulting fit is described by:

$$p_{ij} = b_{i0} + b_{i1}a + b_{i2}\tau_0 + b_{i3}a^2 + b_{i4}a\tau_0 + b_{i5}\tau_0^2. \tag{5.16}$$

Analog to $\boldsymbol{\theta}$, two hundred and twenty regressions were made for the coefficients p_{ij}, for every parameter in $\boldsymbol{\theta}$, except for β, which had fewer coefficients. The results of the regression analysis for every coefficient for each controller's parameter are shown in Tables 5.6, 5.7, 5.8 and 5.9.

Table 5.6 Coefficients for κ_p.

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

5.4.1 Comparison of regression results vs. ENCC results

After setting up the proposed equations for $\boldsymbol{\theta}$, simulations were run to compare the original data against the proposed methodology. The test plant is modeled as:

$$P_1(s) = \frac{e^{-1.5\hat{s}}}{(\hat{s}+1)(0.5\hat{s}+1)} \tag{5.17}$$

Table 5.7 Coefficients for τ_d.

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

Table 5.8 Coefficients for τ_i.

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

Where $K = 1$, $T = 1$ s, $L = 1.5$ s and $a = 0.5$. Table 5.10 shows a comparison between the results of the tuning from the MOO problem against the results of using the proposed tuning, where the values for δ and γ were chosen to be the worst case scenario for both J_{do} and J_{di}, so $\delta = 1$ and $\gamma = 1$.

As it can be seen from Table 5.10, results obtained from the proposed tuning rule are very similar to those obtained from the ENNC MOO method. Plots for each method were drawn as shown in Fig. 5.18. The response of the control signal is

Table 5.9 Coefficients for β.

p_{ij}	b_{ik}	
p_{00}	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_{01}	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_{02}	b_{20}	0.585
	b_{21}	-0.082
	b_{22}	-0.280
	b_{23}	0.116
	b_{24}	0.011
	b_{25}	0.044

Table 5.10 Comparison of the ENNC MOO data vs. fitted data, using $\delta = 1$ and $\gamma = 1$.

$\boldsymbol{\theta}$ parameters and IAEs	ENCC Method	Fitted data
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

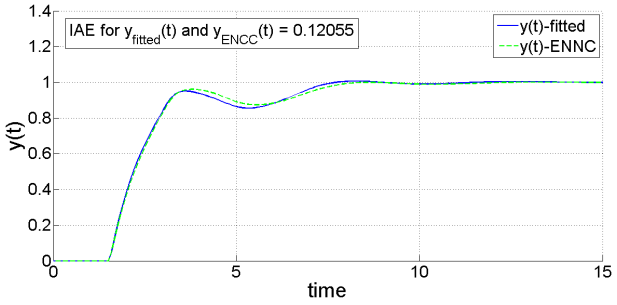

Fig. 5.18 Servo response for the ENCC results and regressions results.

given in Fig. 5.19.

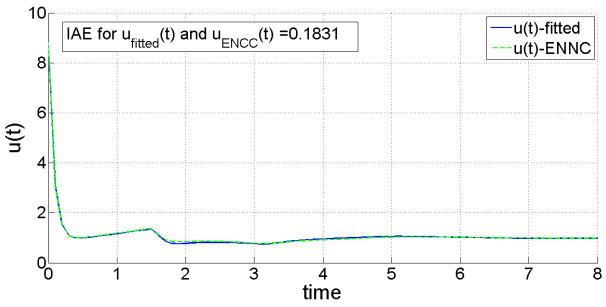

Fig. 5.19 Control action response for ENCC results and regressions, for a reference step change.

The comparison of the response to an input-disturbance is presented in Fig. 5.20. The error between the proposed methodology and the MOO results has an IAE of 0.08, showing that the polynomial equations are able to encapsulate the optimal tuning of the parameters. In Fig. 5.21, the response to the output disturbance is presented and, as it can be seen, the match is good as well, with an IAE value of 0.12.

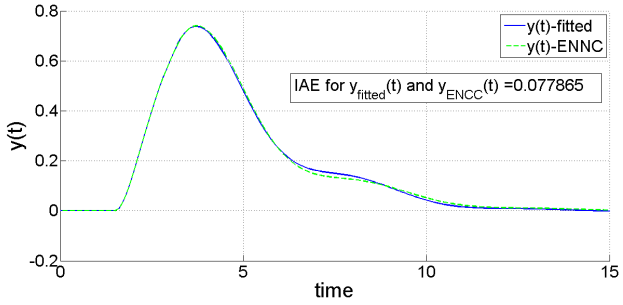

Fig. 5.20 Step input disturbance response for ENNC results and regressions results.

From this example, it is clear that the obtained polynomial equations effectively reproduce the behavior of the parameters found from the ENNC optimization.

The method is also tested for the extreme case scenarios for τ_0 ($\tau_0 = 1$ and $\tau_0 = 2$), as given by the models:

$$P_F(\hat{s}) = \frac{e^{-\hat{s}}}{(\hat{s}+1)(0.5\hat{s}+1)}, \tag{5.18}$$

$$P_S(\hat{s}) = \frac{e^{-2\hat{s}}}{(\hat{s}+1)(0.5\hat{s}+1)}, \tag{5.19}$$

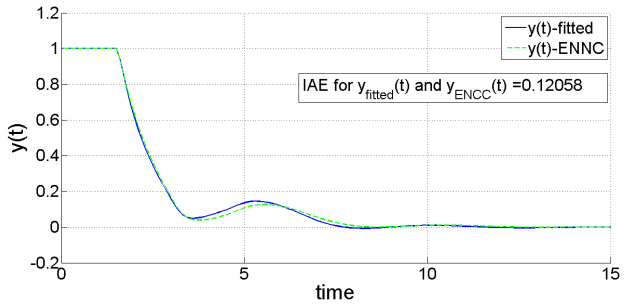

Fig. 5.21 Step output disturbance response for ENNC results and regressions results.

where $P_F(\hat{s})$ and $P_S(\hat{s})$ stand for the fastest and slowest time-delayed dominant test-bench plants, respectively.

For each of these plants, it was established that $\delta = 0.5$ and $\gamma = 0.5$, which is an intermediate case for degradation for both δ and γ in the Pareto front. Results for J_r for P_F and P_S are shown in Table 5.11.

Table 5.11 Results for J_{di}, J_{do} and J_r, using $\delta = 0.5$ and $\gamma = 0.5$.

$\boldsymbol{\theta}$ and IAE	For $P_F(s)$	For $P_S(s)$
κ_p	1.150	0.742
τ_i	1.987 s	2.345 s
τ_d	0.425 s	0.629 s
β	0.887	0.919
J_r	1.955	3.360
J_{di}	1.729	3.162
J_{do}	1.874	3.237
M_s	2.024	1.976

From these examples, it is clear that the proposed methodology is able to produce Pareto-optimal controllers, for a large set of plants. The obtained dynamical response can be easily changed by the control engineer and, if the selection of δ and γ is appropriate, the controller parameters are likely to produce a closed-loop system with a maximum sensitivity such that $M_s \leq 2$. The principal characteristic of the proposed methodology, is its ability to let the user select the dynamical behavior, taking into account multiple sources of disturbances, unlike other PID tuning rules.

5.4.2 Comparison of proposed tuning rule against uSORT2 tuning rule

Servo and disturbance responses were obtained for the test-bench plant in (5.17), by using the proposed tuning rule, and then, by using the uSORT2 tuning method for 2DoF PID controllers. The objective is to compare both rules to see which one is more flexible in terms of robustness and performance. The uSORT2 method was selected for comparison given that it minimizes J_{di} and find a sub-optimal solution for J_r while keeping $M_s = 2$.

Two cases were considered, one in which δ and γ were maximum as in the comparison made in Table 5.12, and the others were when δ and γ have their minimum achievable value for the obtained Pareto front as in Table 5.13. It is clear that for

Table 5.12 Result comparison of the proposed tuning rule vs. uSORT2 tuning rule, using $\delta = 1$ and $\gamma = 1$.

$\boldsymbol{\theta}$ and IAEs	Proposed tuning rule	uSORT2 tuning rule
κ_p	0.793	0.814
τ_i	2.113 s	1.676 s
τ_d	0.720 s	0.775 s
β	1.000	0.788
J_r	2.691	2.692
J_{di}	2.673	2.290
J_{do}	2.691	2.451
M_s	1.945	2.007

Table 5.13 Comparison of the proposed tuning rule vs. uSORT2 tuning rule, using $\delta = 0.302$ and $\gamma = 0$.

$\boldsymbol{\theta}$ parameters and IAEs	Proposed tuning rule	uSORT2 tuning rule
κ_p	0.820	0.814
τ_i	1.808 s	1.676 s
τ_d	0.670 s	0.775 s
β	0.8261	0.788
J_r	2.653	2.692
J_{di}	2.307	2.290
J_{do}	2.431	2.451
M_s	1.935	2.007

the two different variations of δ and γ, the minimum IAE for servo response was obtained according to the Pareto front. Also it is important to note that, for the proposed tuning rule, although variations of δ and γ were performed, the maximum sensitivity remained under the value of $M_s \leq 2.0$. This means that given any two inputs within the established Pareto front, the proposed tuning rule gives the opti-

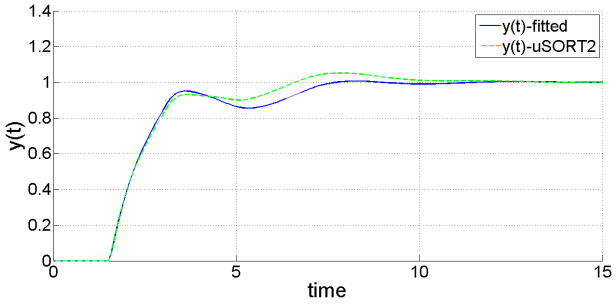

Fig. 5.22 Servo input response for both proposed rule and uSORT2 rule.

mized value for J_r and it manages to maintain the controller robustness. On the other hand, the uSORT2 tuning rule only focuses on the latter. The uSORT2 methodology provides excellent results, but it is focused on optimizing J_{di}. The proposed methodology, although more complex in its equations, gives the control engineers, the freedom to mold the dynamic response to their needs. As an example, the servo response of the controlled system with the proposed methodology and the uSORT is presented in Fig. 5.22

5.5 MOOT-PID, multi-objetive optimization tool for PID controllers

An application that allows the user to optimally tune 2DoF PID using a ODSOPTD model for the plant was developed. Since all the results in the obtained database are optimal from a Pareto standpoint, it is necessary to provide the user with a tool to select the final tuning of the controller.

The user has to provide the following inputs:

1. A plant model in the form of a ODSOPTD transfer function.
2. A desired robustness (given by the maximum sensitivity).
3. The allowed degradation of each cost function. The user can choose a degradation level from zero to one for each cost function (J_r, J_{di}, J_{do}). A value of zero means no degradation and one represents a total degradation. One of the cost functions needs to have a degradation of zero. If this is not true, the program automatically finds the lowest value of J_{di}.

The tool then provides the Pareto-optimal tuning along with a simulation of the closed-loop response. It has to be noticed that the tool interpolates only from the Pareto front. Even if the user let any function to have a degradation of 1, the tuning still is Pareto optimal. Therefore, the performance of the closed loop will be better (in the Pareto sense and with the given constraints) than any other possible parameter combinations.

5.5.1 Presentation of the tool

The main window is shown in Fig 5.23. The GUI has four main components:

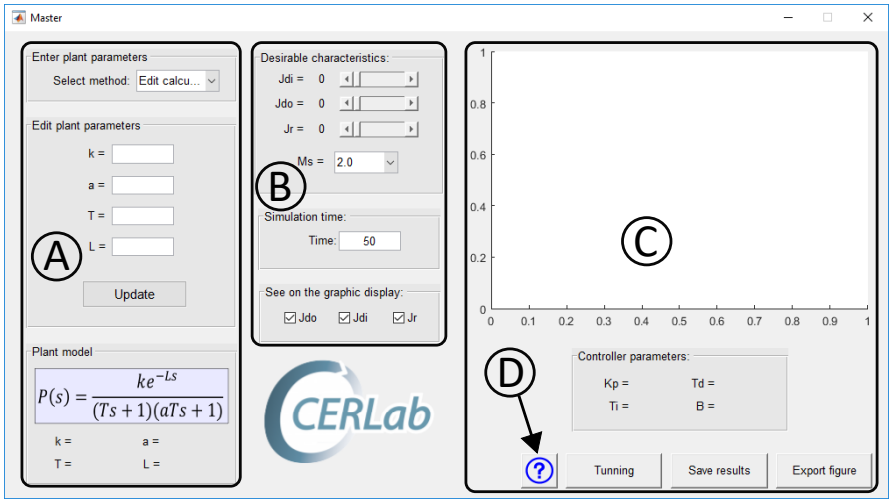

Fig. 5.23 The developed graphical user interface (GUI) for tuning PID.

- "A" part was designed to introduce the plant parameters using one of three possible methods: directly introducing its parameters, identifying the model plant from workspace data or identifying a plant model from a comma-separated values (CSV) file. These options are selected in the left top corner of the GUI.
- "B" part allows to choose the desirable characteristics for the tuning of a 2DoF PID, using the degradation values of J_{di}, J_{do} and J_r. Also, the closed-loop robustness can be selected by the variable M_S.
- "C" part where the results of "A" and "B" parts are shown, in graphical and numerical form. It is possible to generate a report of results in plain text format and also to export the generated figure to MATLAB®.
- The "D" part is the help manual, that explains the steps for using the GUI.

As it can be seen, the user interface is simple and easy to use. Since all the optimization have been performed off-line, the tool is fast in finding the final tuning. The simulation of the closed-loop was computed with a MEX compiled file that solves the delay differential equations[2] using a fourth order Runge-Kutta algorithm.

[2] Compiled MEX files are provided for 64 bits Linux and Windows machines. Other architectures or operating systems should be compiled from source.

5.5.2 Plant model

To model the plant, either from the workspace or from a CSV file, there are two possible ways to find the parameters of the ODSOPTD method. The first method is using the 123-C algorithm, that uses three points of the reaction curve to find the model [3]. The other method uses the optimization toolbox to find the best parameters of the ODSOPTD transfer function.

The tool was tested by applying an experimental identification of a ODSOPTD plant described by

$$P(s) = \frac{e^{-0.2s}}{(s+1)(0.5s+1)}. \tag{5.20}$$

The input data correspond to an artificial response produced using (5.20). From the data, the results using the optimization method for system identification presented in Fig. 5.24. As it can be seen, the programmed optimization was able to find

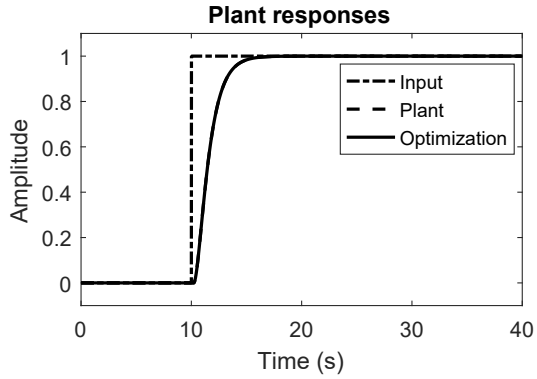

Fig. 5.24 Operating scheme.

exactly the model of the plant. It is very important to count with a good model since the interpolation performed in the tool depends heavily from the knowledge of the parameters. The tool is able to find the PID controller if the following conditions on the plant are fulfilled:

$$\begin{aligned} 0.1 \leq \frac{L}{T} \leq 2 \\ 0 \leq a \leq 1 \end{aligned} \tag{5.21}$$

5.5.3 Example of use.

The ODSOPTD plant in (5.20) is used to verify the correct operation of this tool. A typical closed-loop response is presented in Fig 5.25. As it can be seen, the tool is

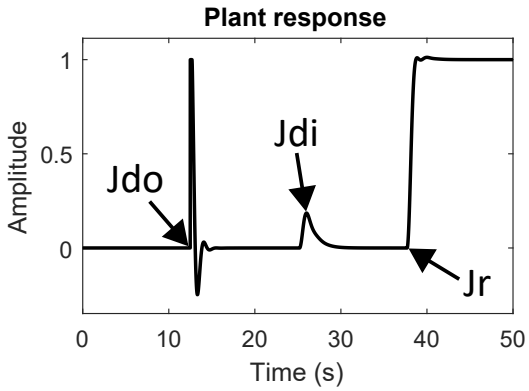

Fig. 5.25 Control loop response given by the tool.

able to plot the response of each cost function individually.

In order to demonstrate the usefulness of the tool, three different scenarios where tested. On Table 5.14 the degradation settings are shown. Each case represents the

Table 5.14 Degradation of parameters.

Parameter	Figure	Jdi	Jdo	Jr	Ms
J_{di}	Fig. 5.26	0.1/0.9	0.5	0	2.0
J_{do}	Fig. 5.27	0	0.1/0.9	0.5	2.0
J_r	Fig. 5.28	0	0.5	0.1/0.9	2.0

change in the response where one degradation is changed while the others are kept constant. In all cases one of the cost functions degradation is set to zero, this means that, from all the possible optimal tuning that fulfills the degradation requirements, the tool is going to give the tuning that produces the lowest cost function for that particular cost function. When the user selects the desired levels of degradation that do not correspond to the Pareto front (that is, a point so close to the utopia point that does not correspond to the feasible region), a pop-up window appears asking the user to relax the degradation limits.

On Fig. 5.26, two different responses are shown where the allowed degradation of J_{do} is changed while maintaining the other parameter constant. The first response corresponds to a change in $d_o(s)$, the second one is a change in $d_i(s)$ and the final response is a change in $r(s)$. It is clear that, when allowing a higher degradation

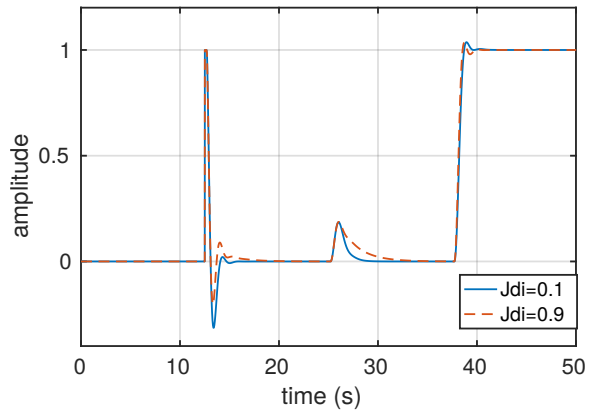

Fig. 5.26 Variation of J_{di} degradation while keeping the other degradations constant.

of J_{di}, the response to $d_i(s)$ has a larger IAE. However, since the responses to each disturbance sources are not independent, varying one degradation settings makes the other responses to vary. However, since the degradation of J_r was kept constant at zero, the tuning given by the tool correspond to the lower value of J_r, while satisfying the other degradation levels.

The response of the second case of Table 5.14 is presented in Fig. 5.27. This case

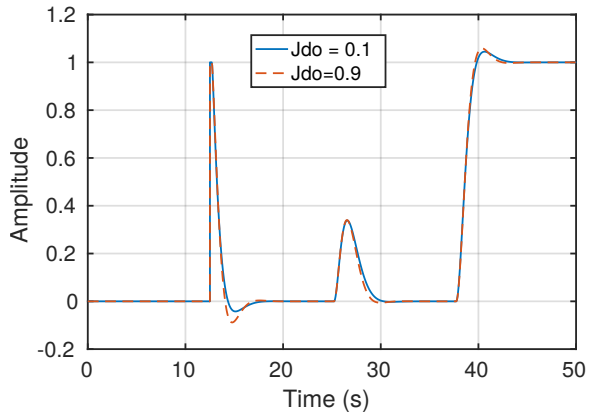

Fig. 5.27 Jdo comparison.

is interesting because it reflects one of the benefits of analyzing the tuning of PID parameters using the Pareto framework. It is clear that, from the point of view of the control engineer, the tuning with a degradation of $J_{do} = 0.1$ is more advantageous than for the case of $J_{do} = 0.9$, because the degradation on the other responses is minimal while improving the response to a change in $d_o(s)$. Finding this results is

easy using the tool presented in this paper, without the need to perform individual optimization for each case.

The third case is presented in Fig 5.28. In this case is interesting to note that when

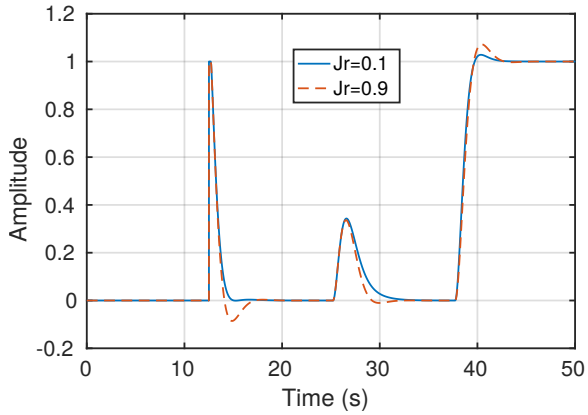

Fig. 5.28 Jr comparison.

the allowed degradation of J_r is set to 0.1, the performance of input disturbance rejection response is also improved. This is important to note because, even thought all three cost function has some kind of antagonism, there are some sections of the Pareto front where is possible to improve two function at the cost of worsening the third function as in this case. The tuning of all the controllers are presented in Table 5.15 for reference. Finally, the last characteristic of the tool is that it is able

Table 5.15 Obtained parameter of the presented cases.

Case	value	K_p	T_i	T_d	β
J_{di}	0.1	4.8874	1.1198	0.2566	0.6111
	0.9	4.7449	2.0346	0.2809	0.9095
J_{do}	0.1	1.6891	1.2225	0.2738	0.7322
	0.9	1.7400	1.1333	0.2255	0.6647
J_r	0.1	1.7237	1.4461	0.2522	0.8874
	0.9	1.7166	1.0971	0.2535	0.6685

to save the results of the tuning in plain text. An example of the output report is presented in Fig. 5.29.

```
Characteristic,Value
Plant parameters,Value
K,1
T,1
a,0.5
L,0.2
Desirable characteristics,Value
Jdi,0.26
Jdo,0.24
Jr,0
Ms,2.0
Controller parameters,Value
Kp,4.7206
Ti,1.3629
Td,0.2848
B,0.7379
```

Fig. 5.29 Report output.

References

[1] V.M. Alfaro and R. Vilanova. "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*. Sept. 2012, pp. 1–7. DOI: 10.1109/ETFA.2012.6489607.

[2] V.M. Alfaro and R. Vilanova. "Simple Robust Tuning of 2DoF PID Controllers From A Performance/Robustness Trade-off Analysis". In: *Asian Journal of Control* 15.6 (2013), pp. 1700–1713. ISSN: 1934-6093. DOI: 10.1002/asjc.653. URL: http://dx.doi.org/10.1002/asjc.653.

[3] Víctor M. Alfaro. "Identificación de modelos de orden reducido a partir de la curva de reacción del proceso". In: *Ciencia y Tecnología* 24.2 (2006). (in Spanish), pp. 197–216. URL: http://revistas.ucr.ac.cr/index.php/cienciaytecnologia/article/view/2647/2598.

[4] Víctor M. Alfaro and Ramon Vilanova. "Optimal Robust Tuning for 1DoF PI/PID Control Unifying FOPDT/SOPDT Models". In: *IFAC Proceedings Volumes* 45.3 (2012). 2nd IFAC Conference on Advances in PID Control, pp. 572–577. ISSN: 1474-6670. DOI: http://dx.doi.org/10.3182/20120328-3-IT-3014.00097. URL: http://www.sciencedirect.com/science/article/pii/S1474667016310874.

[5] K. J. Åström and T. Hägglund. "Benchmark Systems for PID Control". In: *Proceedings IFAC Workshop Digital Control: Past, Present and Future of PID Control*. 5-7 abril, Terrasa, España. 2000.

[6] Wei-Der Chang. “Nonlinear CSTR control system design using an artificial bee colony algorithm”. In: *Simulation Modelling Practice and Theory* 31.0 (2013), pp. 1–9. ISSN: 1569-190X. DOI: http://dx.doi.org/10.1016/j.simpat.2012.11.002.

[7] R. Vilanova, V. M. Alfaro, and O. Arrieta. “Analytical Robust Tuning Approach for Two-Degree-of-Freedom PI/PID Controllers”. In: *Engineering Letters (IAENG)* 19 (2011), pp. 204–214.

[8] Deyin Zhang, Dagui Huang, and Yanqiu He. “Intelligent temperature control strate for LiTaO3 thin film deposition process”. In: *Intelligent Mechatronics and Automation, 2004. Proceedings. 2004 International Conference on*. Aug. 2004, pp. 339–344. DOI: 10.1109/ICIMA.2004.1384216.