Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 354 (2017) 3945–3970

www.elsevier.com/locate/jfranklin

Optimal Nash tuning rules for robust PID controllers

Helem Sabina Sánchez[a], Antonio Visioli[b,*], Ramon Vilanova[c]

[a] *Research Center for Supervision, Safety and Automatic Control (CS2AC) Universitat Politècnica de Catalunya, Rambla Sant Nebridi, s/n, 08022 Terrassa, Spain*

[b] *Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Via Branze 38, Brescia I-25123, Italy*

[c] *Departament de Telecomunicació i Enginyeria de Sistemes, Universitat Autònoma de Barcelona, Bellaterra, Barcelona 08193, Spain*

Received 18 September 2016; received in revised form 10 February 2017; accepted 16 March 2017
Available online 3 April 2017

Abstract

In this paper, we propose tuning rules for one degree-of-freedom proportional-integral-derivative controllers, by considering important aspects such as the trade-off in the performance in the servo and regulation operation modes and the control system robustness by constraining the maximum sensitivity peak. The different conflicting objectives are dealt with by using a multi-objective optimization algorithm to generate the trade-off optimal solutions. In this context, a simple tuning rule is determined by using the Nash solutions as a multi-criteria decision making technique. The Nash criteria is shown to provide convenient trade-off solutions for the controller tuning problem. Illustrative simulation examples show the effectiveness of the method.

1. Introduction

It is well-known that proportional-integral-derivative (PID) controllers are still the most widespread controllers in the process industry owing to the cost/benefit ratio they can provide, which is often difficult to improve with more advanced control techniques. In fact, even if PID controllers have been employed for almost a century and a lot of experience in their

* corresponding author.
E-mail addresses: helemsabina.sanchez@uab.cat (H.S. Sánchez), antonio.visioli@unibs.it (A. Visioli), ramon.vilanova@uab.cat (R. Vilanova).

http://dx.doi.org/10.1016/j.jfranklin.2017.03.012

use has been gained, researchers are continuously investigating new design methodologies in order to improve their overall performance, thanks also to the advancement of the computing technology that makes the application of optimization techniques easier.

The research in PID controllers has always been especially focused on the tuning issue, that is, the selection of the PID parameters that are most suitable for a given application. In particular, the development of tuning rules that allow the user to determine the controller gains starting from a simple process model is a topic that has received great attention since the first proposal made by Ziegler and Nichols [47]. Indeed, many tuning rules have been devised [27]. They are related to different controller structures (the PID can be, for example, in interacting or non interacting form), different process models (for example, a first- or second-order plus dead time transfer function), different control tasks they are required to address (for example, set-point following or load disturbance rejection) and different approaches they are based on (for example, empirical, analytical, or optimal). In fact, it has to be recognized that the tuning of the controller has often to be performed by taking into account conflicting requirements.

From the point of view of output performance, we can identify a design trade-off by considering the effects of load disturbances and set-point changes on the feedback control system. On the other side, there is also a trade-off between performance and robustness. It is worth stressing this point because, in the literature, design techniques normally focus on performance in either servo or regulatory mode, see for example [27,41] for a historical review.

In general, it has to be taken into account that obtaining a fast load disturbance response usually implies increasing the bandwidth of the control system at the expense of a more oscillatory set-point step response [6]. Further, a decrement of the settling time of the response can usually be obtained at the expense of a decrement of the robustness of the closed-loop control system (and, consequently, an increment of the control effort) [1]. It appears therefore that the tuning of the controller is critical if a one-degree-of-freedom PID controller is used and both tracking and regulatory tasks have to be faced.

The issues mentioned before can be considered as a strong motivation for the development of an intermediate tuning that considers the trade-off between servo/regulation operation modes and between performance and robustness. In this context, the use of multi-objective optimization tools can help in determining the most suitable PID parameters for a given application, by taking into account the above mentioned issues [7,14–16,29,32,38–40].

However, it is clear that having a tuning rule is much more desirable in order to keep the simplicity of the use of PID controllers, which is one of their most appreciated features. For this reason, tuning rules that achieve the minimization of integral performance criteria have been proposed in the past, by assuming a first-order-plus-dead-time (FOPDT) process model [46], which is known to capture well the dynamics of many self-regulating processes. In that work, the set-point following and the load disturbance rejection performance have been considered separately, which makes the choice of the tuning difficult if both tasks have to be addressed in a given application, especially if a one-degree-of-freedom control structure is considered. Tuning rules for weighted servo/regulation control operations, namely, tuning rules that balance the optimal tuning in the two modes, have been presented in [3]. Nevertheless, in these cases the robustness of the control system has not been considered and this might imply a significant performance decrease if the dynamics of the process changes and/or the control effort is too high.

Taking this into account, the purpose of this work, which is an extended version of [35], is to provide a new solution to the tuning problem. First, a multi-objective optimization (MOO)

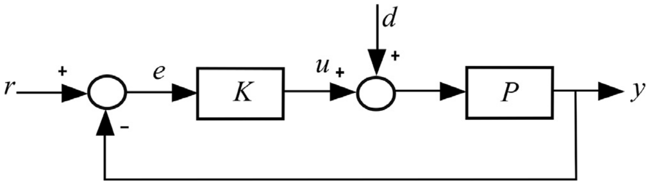


Fig. 1. The considered feedback control scheme.

procedure is used in order to maximize the performance while, at the same time, achieving a satisfactory level of robustness. In particular, a multi-objective optimization design (MOOD) procedure based on the Normalized Normal Constraint (NNC) algorithm is applied to a family of FOPDT processes, yielding a set of Pareto fronts. Secondly, the Nash solution [18] is calculated for each case and tuning rules are determined by fitting the results obtained for the different PID parameters. A comparison with the non-robust tuning rules proposed in [3] is also performed.

The paper is organized as follows. Section 2 is devoted to the problem statement. Then, in Section 3 the main concepts related to the MOO are briefly reviewed. In Section 4, the Nash solution is presented as a multi-criteria decision making technique, while in Section 5, the optimization results are used to devise the tuning rules for optimal servo/regulation trade-off. Some examples and simulation results are given in Section 6 and conclusions are drawn in Section 7.

2. Problem statement

In this section, the control framework and the standard evaluation indexes used in this paper to measure the performance and the robustness are introduced.

2.1. Control framework

The purpose of a control system is to obtain a desired response for a given system. We consider the typical feedback control system represented in Fig. 1, where P is the process, K is the controller, r is the set-point signal, u is the control signal, d is the load disturbance, y is the process output and $e = r - y$ is the control error. The process dynamics, described by the transfer function P, is considered to be self-regulating and represented by a first-order-plus-dead-time (FOPDT) model of the form:

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

where K is the process gain, T is the time constant, and L is the dead-time. The process dynamics can be fully characterized in terms of the normalized dead-time, defined as $\tau = L/T$, which can be associated to a measure of the difficulty in controlling the process. Note that FOPDT models are frequently used in process control because they are simple and they describe with a sufficient accuracy the dynamics of many industrial processes [42]. Further, the estimation of the parameters of this kind of models can be performed by means of a very simple step-test experiment to be applied to the process. This represents a clear advantage in industry with respect to more advanced identification methods. The controller K is a one-degree-of-freedom PID controller with a derivative time filter, whose transfer function is given by:

$$K(s) = K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d}{N} s}\right) \tag{2}$$

where K_p is the proportional gain, T_i is the integral time constant and T_d is the derivative time constant. The derivative time noise filter constant N usually takes values within the range 5–33 [5,42]. Here, without loss of generality, the value $N = 10$ has been selected [46].

The Laplace transform of the process variable of the control system of Fig. 1 is given by

$$y(s) = \frac{K(s)P(s)}{1 + K(s)P(s)} r(s) + \frac{P(s)}{1 + K(s)P(s)} d(s) \tag{3}$$

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

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

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

$$y_{sp}(s) := \frac{K(s)P(s)}{1 + K(s)P(s)} r(s) \tag{4}$$

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

$$y_{ld}(s) := \frac{P(s)}{1 + K(s)P(s)} d(s) \tag{5}$$

When both tasks have to considered, the tuning of the PID controller becomes difficult as the two specifications are conflicting (achieving a high performance in the load disturbance rejection tasks requires in general a high bandwidth of the control system which yields poor stability margins and large overshoots in the set point step responses [42]). For this reason, the servo/regulatory trade-off has to be explicitly taken into account.

2.2. *Performance/robustness*

Usually, the tuning of the controller must be done by taking into account different objectives, such as:

- Output performance: in order to measure it, we use the well-known integrated square error (ISE) defined as

$$ISE := \int_0^\infty e^2(t)dt. \tag{6}$$

- Robustness: in order to measure it, we use the maximum sensitivity M_s, which is the inverse of the minimum distance from the Nyquist plot to the critical point $(-1,0)$. It is also defined as:

$$M_s := \max_{w\in[0,+\infty)} \left| \frac{1}{1+K(s)P(s)} \right|_{s=jw}. \tag{7}$$

The desirable robustness level can be established using the typical range between 1.4 (high robustness level) and 2.0 (low robustness level) [5]. As the derivation of the tuning rule will be based on simple FOPDT models, it is important to recognize here the need to introduce a measure, such as M_s, in order to ensure robustness guarantees for the resulting closed-loop system.

It is worth stressing that the designer can easily extend the proposed methodology to other performance indexes of interest. Here the choice of the integral square error is motivated by the work presented in [3], which is the main source for comparison with the tuning technique presented here. To provide a complete comparison, two different ISE indexes have been considered: the one obtained from the set-point step response (servo mode) and the other obtained from the load-disturbance step response (regulatory mode) by achieving, at the same time, a required level of robustness. In this context, a MOO procedure will be introduced in the next section and the control problem will be redefined as a multi-objective problem (MOP).

3. Multi-objective optimization

This section provides a brief description of MOO and how a bargaining solution can enter at the decision making stage. The definition of the MOP as well as the generation of the potential solutions are presented in a succinct way as much attention will be devoted to the last step of the MOOD procedure where the final solution has to be chosen and where the bargaining option plays a key role.

3.1. Generalities

A MOO can be defined as the problem of finding a vector of decision variables that satisfies equality and inequality constraints and optimizes a vector (cost function), whose elements represent objective goals. These objective functions mathematically state design specifications, which are usually in conflict with each other. Usually, the definition given by Pareto is employed for multi-objective optimality [30]. This implies the existence of a set of solutions, where no solution is better than the others, but they differ in the degree of performance between the objectives [26]. This set of solutions implies that there is some additional flexibility at the decision making stage that allows the introduction of some criteria for choosing the best compromise solution. The role of the designer is to select the most preferable solution according to his/her needs and preferences for a particular situation. A MOP can be tackled by performing a simultaneous optimization of all the defined cost functions. MOOD procedures have shown to be a valuable tool for control engineers. These procedures consist at least of three main steps (see Fig. 2): (1) the MOP definition, (2) the MOO process and (3) the multi-criteria decision making (MCDM) stage.

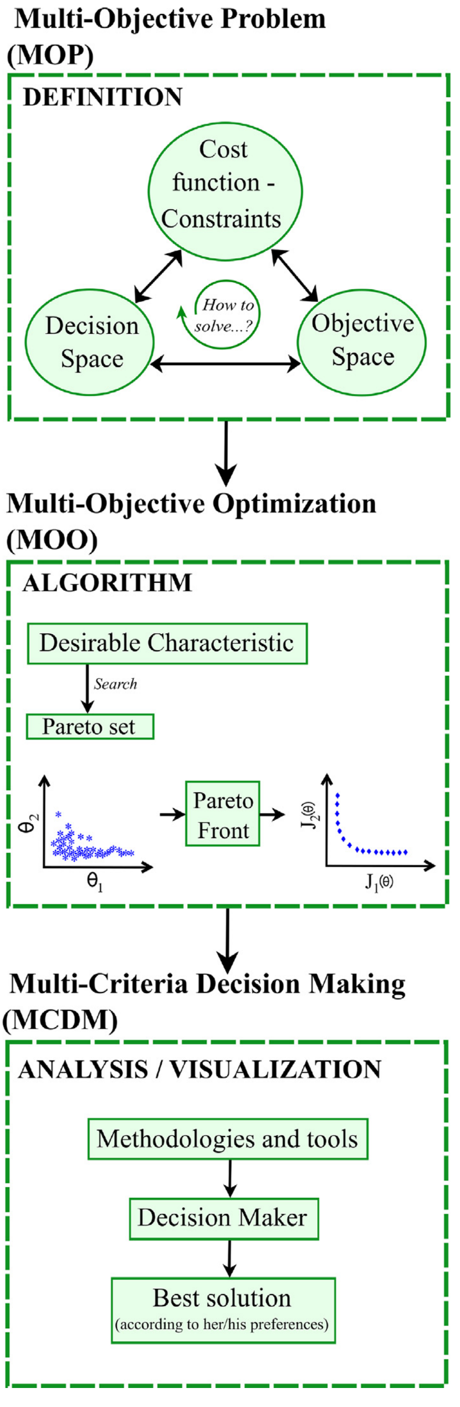


Fig. 2. A multi-objective optimization design (MOOD) procedure.

3.2. Multi-objective problem definition

As already mentioned, control requirements often include specifications related to load disturbance compensation, set-point following and robustness to process modelling uncertainties. In the previous sections, the control problem has been formulated. Then, the MOP statement for PID controller tuning can be represented as:

$$\min_{\boldsymbol{\theta_c}} \boldsymbol{J}(\boldsymbol{\theta_c}) = [J_{ld}(\boldsymbol{\theta_c}), J_{sp}(\boldsymbol{\theta_c})] \tag{8}$$

such that

$$g_{ls} \leq g(\boldsymbol{\theta_c}) \leq g_{us} \tag{9}$$

where

$$\boldsymbol{\theta_c} = \left[K_p, T_i, T_d\right] \tag{10}$$

are the parameters of the PID controller,

$$J_{ld}(\boldsymbol{\theta_c}) = ISE_{ld} := \int_0^\infty e^2(t)dt, \quad r = 0 \tag{11}$$

is the integrated square error when a load disturbance step response is considered, and

$$J_{sp}(\boldsymbol{\theta_c}) = ISE_{sp} := \int_0^\infty e^2(t)dt, \quad d = 0 \tag{12}$$

is the integrated square error when a set-point step response is considered. As constraint we take

$$g(\boldsymbol{\theta_c}) = M_s(\boldsymbol{\theta_c}) := \max_{w\in[0,+\infty)} \left| \frac{1}{1 + K(s; \boldsymbol{\theta_c})P(s)} \right|_{s=jw} \tag{13}$$

that is the maximum sensitivity of the control system, which is constrained in the interval $g_{ls} = 1.4$ and $g_{us} = 2.0$ as it is usually done in process control [5].

The multi-objective problem, as it has been stated, is based on establishing the robustness of the resulting control system as a constraint to be ensured instead of a quantity that has to be (max)minimized along with the other objectives. There are other formulation of the robust PID control problem such as those presented in [17] and [44] where robust stability is incorporated as another objective function. This leads to an optimization problem of different nature from the one stated here. In this work, the fact of introducing the robustness as a constraint is twofold. On the one hand, it will allow the optimization search to focus on minimizing J_{ld} and J_{sp} while keeping M_s within the specified limits. On the other hand, it will allow us to solve, in the next sections, the problem under two different scenarios: to trade-off the servo and regulation operation without and with the robustness constraint. The solutions of the unconstrained case will provide a measure of the needed performance degradation when robustness is considered and also a comparison with the intermediate tuning rule of [3].

3.3. Generation of the Pareto front (MOO)

A MOP can be handled by performing a simultaneous optimization of all objectives. This implies the existence of a set of solutions, where no one is better than the others, but differ in the degree of performance between the objectives [26]. This set of solutions will offer a higher degree of flexibility in order to select a preferable solution at the multi-criteria decision making stage. This set of solutions is what is reflected by the Pareto front. The generation of the Pareto front involves the application of some sort of optimization procedure.

Some of the classical strategies to approximate the Pareto set include: Normal constraint method [23], Normal boundary intersection (NBI) method [10], Epsilon constraint techniques [26] and Physical programming [24]. Multiobjective Evolutionary algorithms (MOEAs) have also been used to approximate a Pareto set [45], due to their flexibility when evolving an entire population towards the Pareto front. A comprehensive review of the early stages of MOEAs is contained in [9]. The most popular techniques include Genetic Algorithms (GA) [21,36], Particle Swarm Optimization (PSO) [20], and Differential Evolution (DE) [11,25,37]. Nevertheless, evolutionary techniques as Artificial Bee Colony (ABC) [19] or Ant Colony Optimization (ACO) [13] algorithms are becoming popular. No evolutionary technique is better than the others, since each one has drawbacks and advantages.

MOO process has already been proposed for the tuning of PI-PID controllers under different frameworks [10,17,31,43,44]. However, some experiments already conducted by the authors confirm the suitability of using the Normalized Normal Constraint (NNC)[1] algorithm to generate the set of optimal solutions. As a set of MOP needs to be solved, this more simple but effective numerical approach is preferred to more advanced evolutive formulations. Using this algorithm, the optimization problem is separated into several constrained single optimization problems. After a series of optimizations, a set of evenly distributed Pareto solutions results. The NNC method incorporates a critical linear mapping of the design objectives. This mapping has the desirable property that the resulting performance of the method is entirely independent from the design objectives scales. Further, a well distributed set of Pareto points is generated even in numerically demanding situations [23]. It is worth noting that MOO techniques search for a discrete approximation of the Pareto set capable of generating a good approximation of the Pareto front. The NNC method is employed here to solve a bi-objective problem, but it can be generalized to n-objectives. For more details about the method see [23]. This algorithm has been shown to be useful for PID controllers in [34].

Finally, when the Pareto front approximation has been obtained, the MCDM stage is carried out in order to select the most preferable solution for the designer.

3.4. Multi-criteria decision making (MCDM)

All points within the Pareto front are equally acceptable solutions. Nevertheless, there is the need to choose one of those points as the final solution to the MOP for the implementation phase. Several tools and methodologies are available, in order to facilitate the decision making stage [33]. The decision making can be undertaken by using two different approaches: (i) by including additional criteria such that at the end only one point from the Pareto front satisfies all of them, and (ii) by considering one point that represents a fair compromise between all used criteria. From a controller design point of view, the first option can be used to improve the control performance by introducing additional criteria. In other words, as the MOP establishes the search among the Pareto front for a compromise among a set of performance indexes, an additional performance (probably of secondary importance) can be

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

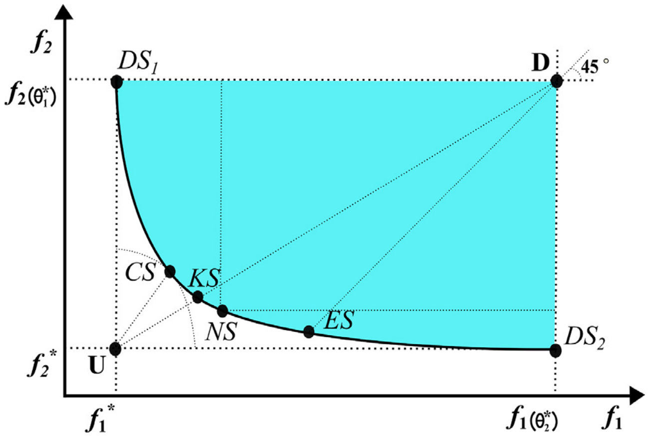


Fig. 3. Location of the bargaining solutions into the Pareto front.

introduced. In this way, a new optimization problem will start with the search domain located in the Pareto set in order to find the best solution. The second option does not introduce more information for the decision making and a *fair* point should be selected in order to represent an appropriate trade-off among the different considered cost functions. In the context of finding a PID controller tuning rule, this second option has been preferred because it can be somehow easily automated. It means that a single proposal for the controller design will be the outcome for the MOP. Obviously, the ideal setup would be to reach the utopia point. However, the utopia point is normally unattainable and does not belong to the Pareto front approximation. This is because it is not possible to optimize all individual objective functions independently and simultaneously. Thus, it is only possible to find a solution that is as close as possible to the utopia point. Such solution is called the *compromise solution* (CS) and is Pareto optimal. This approach, however, starts from a neither attainable nor feasible solution. Therefore it is not very practical as it does not take into account what can be achieved for each one of the individual objectives functions. Another procedure to select a fair point is to use bargaining games [8]. This solution leads to a practical procedure for choosing a unique point, as it will be seen in the next section.

4. Bargaining and *trade-off* solutions selection

In a transaction, when the seller and the buyer value a product differently, a surplus is created. A bargaining solution is then a way in which buyers and sellers agree to divide the surplus. There is an analogous situation regarding a controller design method that is facing two different cost functions for a system. When the controller locates the solution on the disagreement point (D), as shown in Fig. 3, there is a way for the improvement of both cost functions. We can move within the feasible region towards the Pareto front in order to get lower values for both cost functions. Let θ_1^* and θ_2^* denote the values for the free parameter vector θ that achieve the optimal values for each one of the cost functions f_1 and f_2, respectively. Let these optimal values be $f_1^* = f_1(\theta_1^*)$ and $f_2^* = f_2(\theta_2^*)$. On that basis, the utopia point will have coordinates f_1^* and f_2^* whereas the disagreement point will be located at $(f_1(\theta_2^*), f_2(\theta_1^*))$. As the utopia (U) point is not attainable, we need to analyze the Pareto front in order to obtain a solution. A fair point that represents an appropriate trade-off among the cost functions f_1 and f_2 is defined by the coordinates $(f_1^{Pf}, f_2^{Pf}) = (f_1(\theta_1^{Pf}), f_2(\theta_2^{Pf}))$, where the superindex *Pf* means Pareto front. On the basis of this formalism, we can identify, in economic terms, the benefit of each one of the cost functions (buyer and seller) as the differences $f_1(\theta_2^*) - f_1^{Pf}$ and $f_2(\theta_1^*) - f_2^{Pf}$. The bargaining solution will provide a choice for (f_1^{Pf}, f_2^{Pf}) therefore a benefit for both f_1 and f_2 with respect to the disagreement. It is important to notice that the problem setup is completely opposite to the one that generates the compromise solution (CS) as the closest one to the utopia point.

Formally, a bargaining problem is denoted by a pair $< S; d >$ where $S \in \mathbb{R}^2$, $d = (d_1, d_2) \in S$ represents the disagreement point and there exists $s = (s_1, s_2) \in S$ such that $s_i < d_i$. In our case, S is the shaded area shown in Fig. 3 delimited by the Pareto front and its intersection with the axis corresponding to the coordinates of the disagreement point. In Fig. 3, different solutions for selecting a point from the Pareto front are displayed:

1. The disagreement solution (D): it is the solution associated to the disagreement point. Even if it is not the preferred solution for none of the players, it is a well-defined solution.
2. The dictatorial solution for player 1 (DS1): it is the point that minimizes the cost function for player 1. The same concept can be applied to player 2, yielding the dictatorial solution for player 2 (DS2).
3. The egalitarian solution (ES): it is the greatest feasible point (f_1^{Pf}, f_2^{Pf}) that satisfies $f_1(\theta_2^*) - f_1^{Pf} = f_2(\theta_1^*) - f_2^{Pf}$. This point coincides with the intersection of the 45° diagonal line that passes through the disagreement point with the Pareto front.
4. The Kalai–Smorodinsky solution (KS): it is the point (f_1^{Pf}, f_2^{Pf}) corresponding to the intersection of the Pareto front with the straight line that connects the utopia and the disagreement point.
5. The Nash solution (NS): it selects the unique solution to the following maximization problem:

$$\begin{array}{ll} \max\limits_{(f_1^{Pf}, f_2^{Pf})} & (f_1(\theta_2^*) - f_1^{Pf})(f_2(\theta_1^*) - f_2^{Pf}) \\ s.t. & \qquad f_1^{Pf} \le f_1(\theta_2^*) \\ & \qquad f_2^{Pf} \le f_2(\theta_1^*) \end{array}$$

In order to illustrate the different solutions that can be selected from the Pareto front, consider as an example the FOPDT process

$$P_1(s) = \frac{1}{s+1} e^{-0.5s} \tag{14}$$

where, evidently, $K = 1$, $T = 1$, $L = 0.4$ and $\tau = 0.4$. The different solutions that the designer can obtain using the bargaining concept can be seen in Fig. 4. It is important to highlight that in some cases the NS matches with KS: this happens when both negotiators are in a neutral risk (see, for example, [4]).

In his pioneering work on bargaining games, Nash in [18] established a basic two-person bargaining framework between two rational players, and proposed an axiomatic solution concept which is characterized by a set of predefined axioms and does not rely on the detailed bargaining process of players. Nash proposed four axioms that should be satisfied by a reasonable bargaining solution:

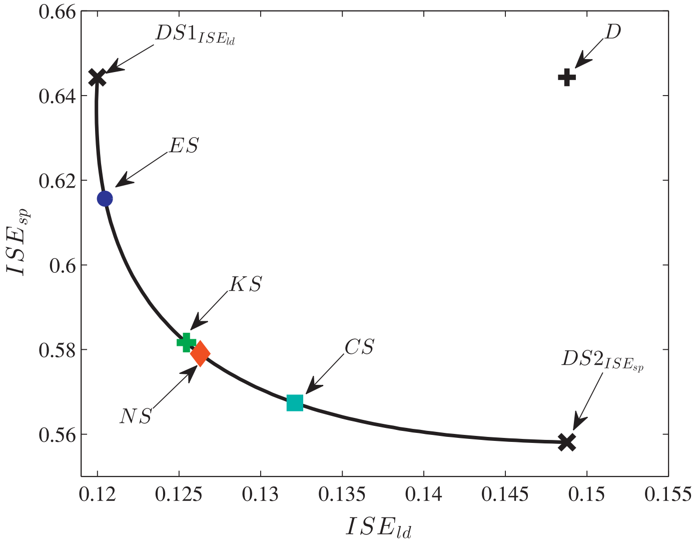


Fig. 4. Pareto front for $\tau = 0.5$. Location of the bargaining solutions.

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

Nash proved that, under mild technical conditions, there is a unique bargaining solution called *Nash bargaining solution* satisfying the four previous axioms. Indeed, by considering the different options for selecting a point from the Pareto front, the NS is the only solution that satisfies these four axioms [18]. In fact, the Nash solution is simultaneously utilitarian (Pareto efficient) and egalitarian (fair). Also from a MOO point of view, by maximizing the product $(f_1(\theta_2^*) - f_1^{Pf})(f_2(\theta_1^*) - f_2^{Pf})$, we are maximizing the area of the rectangle that represents the set of solutions dominated by the NS. Actually, the NS provides the Pareto front solution that dominates the larger number of solutions, therefore being absolutely better (that is, with respect to both cost functions at the same time) than any one of the solutions of such rectangle. These are the reasons why the NS represents an appropriate choice for the automatic selection of the fair point from the Pareto front.

All the previous presentation regarding the *trade-off* solutions selection has concentrated on a bi-dimensional problem formulation with two competing objective functions. In fact, the design problem stated in this work involves two objective functions. However, the main reason comes from the formulation of the Nash solution to the bargaining problem. Nash work [18] provided an axiomatic framework for a t wo-person bargaining framework by establishing the previously presented axioms that should be satisfied by a reasonable bargaining solution. When extending the Nash approach to a multidimensional n-player problem, the game solution is much more involved as, for example, cooperation among players may arise [28]. However, for what matters for the multi-criteria decision making step needed here, we retain the idea of the NS to provide the Pareto front solution that dominates the larger number of solutions, therefore being absolutely better (that is, with respect to all cost functions at the same time) than any one of the solutions within the (hyper)rectangle defined by the Pareto and the disagreement points. This idea can be extended to the case of n competing objective functions by referring to the hypercube defined in terms of the disagreement point (now a point in $\mathbf{R}^n$) and every point in the Pareto surface.

In order to work out such extension to the n-dimensional case, let $\Theta = \{\theta_1, \ldots, \theta_n\}$ be the decision vector. Correspondingly, denote as $\Theta_i^* = \{\theta_{i1}^*, \ldots, \theta_{in}^*\}$ the values of the free decision vector that achieves the optimal value for the objective function f_i, being these optimal values $f_i^* = f_i(\Theta_i^*)$. The Utopia (U) is now defined in terms of the previous optimal values as $U = \{f_1^*, \ldots, f_n^*\}$. The disagreement point is denoted as $D = \{d_1, \ldots, d_n\}$ where each one of its coordinates is computed according to:

$$d_i = \max[f_i((\Theta_1^*)\ f_i((\Theta_2^*)\ \ldots\ f_i((\Theta_n^*)] \quad i = 1, \ldots, n \tag{15}$$

In order to formulate the Pareto front selection, denote as $f^{Pf} = \{f_1^{Pf}, \ldots, f_n^{Pf}\}$ one of the points in the objective space that belongs to the Pareto front. The selection of the compromise solution form the Pareto front is defined as the solution is the following maximization problem:

$$\begin{aligned} &\max_{(f_1^{Pf},\ldots,f_n^{Pf})} \prod_{i=1}^{n} \quad (d_i - f_i^{Pf}) \\ &s.t. \quad f_i^{Pf} \leq f_i(\Theta_i^*) \ \ i = 1, \ldots, n \end{aligned} \tag{16}$$

With this selection we maximize the volume of the dominated solutions set. However, there is one point that should be taken into account regarding the bargaining approach the Nash solution comes from. From the economics point of view (the domain from where the Nash approach comes from) the solution to the bargaining problem becomes more complex as cooperation among different players[2] is considered. In our case, what really matters is the idea of maximizing dominance of the selected Pareto solution with respect to the objective functions space. In that sense, the strict bargaining sense is lost but what is important is the dominance maximization as explained.

5. Optimal tuning and comparison

Returning to the optimization problem described in Eqs. (8)–(9), it has been solved in two different ways:

1. Case 1: by unconstraining the maximum sensitivity M_s in order to compare the proposed method with the intermediate tuning rules presented in [3] where the robustness of the system was not taken into account explicitly.
2. Case 2: by constraining the maximum sensitivity M_s in a range such as $1.4 \leq M_s \leq 2.0.$, therefore guaranteeing that the resulting control system has an acceptable degree of robustness

[2] In game theory and economics literature the term player is used instead of objective function.

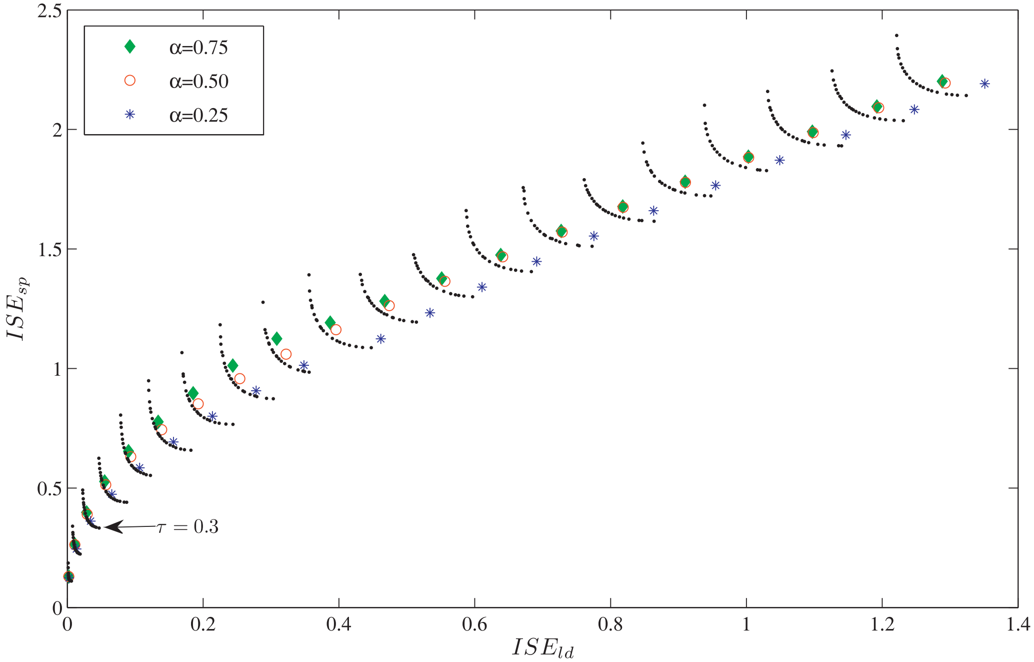


Fig. 5. Pareto fronts for the unconstrained case and performance comparison with the tuning rules proposed in [3] for different normalized dead times.

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

For the first case, the optimal parameters obtained by applying the tuning rule proposed in [3] have been used as initial guess for the NNC algorithm. With this tuning rule, the user can select a weighting factor α that determines the importance of the regulation task with respect to the servo task. Therefore, depending on the operation for the control system, the following cases are identified: $\alpha = 0$ means that only the servo mode is relevant, $\alpha = 0.25$ means that the servo mode is more important than the regulatory mode, $\alpha = 0.50$ means that the two modes are equally important, $\alpha = 0.75$ means that the regulatory mode is more relevant than the servo mode and, finally, $\alpha = 1$ means that only the regulatory mode is relevant. The tuning rules developed in this work have been devised starting from the tuning rules already proposed in [3] for the minimization of the integral square error for the extreme cases, that is, for $\alpha = 0$ and $\alpha = 1$.

The obtained Pareto fronts for the different normalized dead time values are shown in Fig. 5, where the performance obtained with the intermediate tuning rules proposed in [3] are also shown. As expected, the performance deteriorates when the normalized dead time increases, but it is worth noting that:

- for $\tau < 0.3$, the performance obtained with the three weighting factors are located in the Pareto front approximation;
- for $\tau > 0.3$, the performance obtained with the three weighting factors are located outside of the Pareto front approximation.

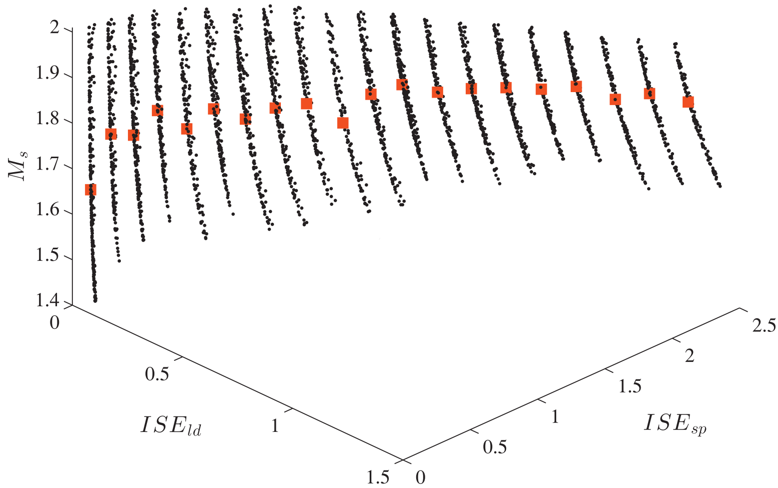


Fig. 6. Pareto fronts for the robust case for different normalized dead times and the corresponding Nash solutions (red square). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Results of the comparison between the intermediate tuning rules proposed in [3] and the NS (constrained case).

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

This indicates that these cases are (slightly) suboptimal from the Pareto front point of view (note that, as already mentioned, the intermediate tuning have been developed for the extreme cases and they are the results of fitting strategies).

The second case is then developed by applying the MOO process with the robustness constraint (9). The corresponding results are shown in Fig. 6. A comparison of the two cases can be performed more simply by considering Fig. 7 where the results related to $\tau = 1.0$ and $\tau = 1.5$ are highlighted. It appears that achieving a higher robustness is paid in terms of performance or, in other words, just optimizing the performance generally yields a control system with a high maximum sensitivity. The necessity of considering the robustness in the optimal selection of the PID parameters appears clearly in Table 1, where it can be noticed that the selection obtained through the constrained Pareto front (NS) proposed in this paper

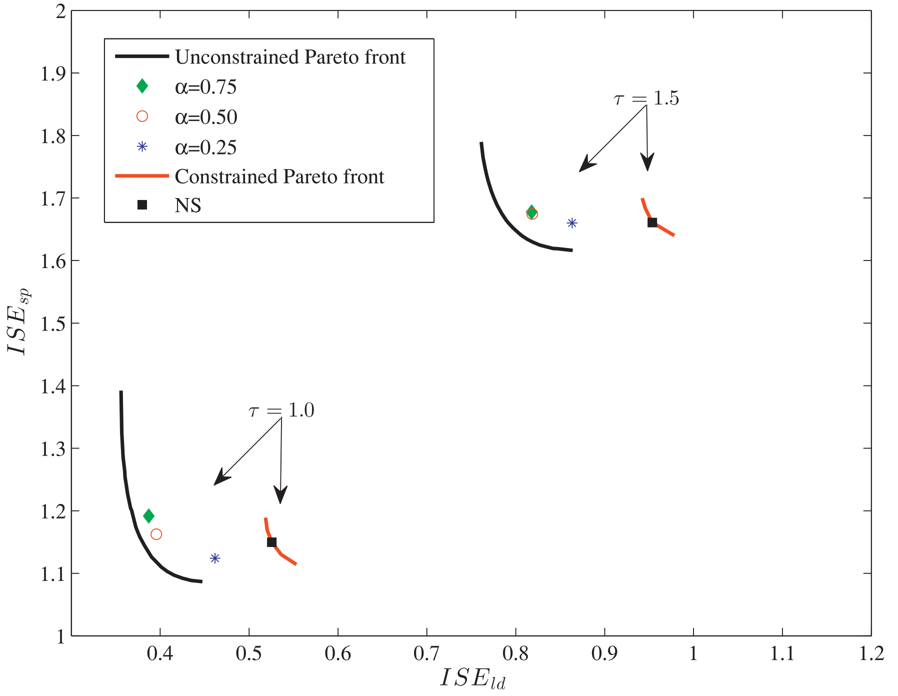


Fig. 7. Comparison between the unconstrained and the constrained (robust case) Pareto fronts.

achieves better robustness values and the performance values are not so high in comparison with the intermediate tuning rules from [3] that achieve better performance without taking into account the robustness. Moreover, it is evident that the value of the maximum sensitivity can be critical for those applications where robustness is important.

5.1. Tuning rules

After the MCDM stage, a set of NSs is obtained as it is shown in Fig. 6. Then, the tuning rules have been obtained by a curve fitting procedure that minimizes the least squares errors with respect to the various optimal values. The curve fitting is shown in Fig. 8. As a result, the following tuning rules have been determined:

$$K_pK = \left(a\tau^b + c\tau^d\right), \tag{17}$$

$$\frac{T_i}{T} = \left(a\tau^2 + b\tau + c\right), \tag{18}$$

$$\frac{T_d}{T} = \left(a\tau^2 + b\tau + c\right), \tag{19}$$

where the values of the parameters are shown in Table 2.

A more thorough comparison of the proposed tuning rules with the results obtained by using the intermediate tuning proposed in [3] for the case of $\alpha = 0.5$ is shown in Fig. 9. It

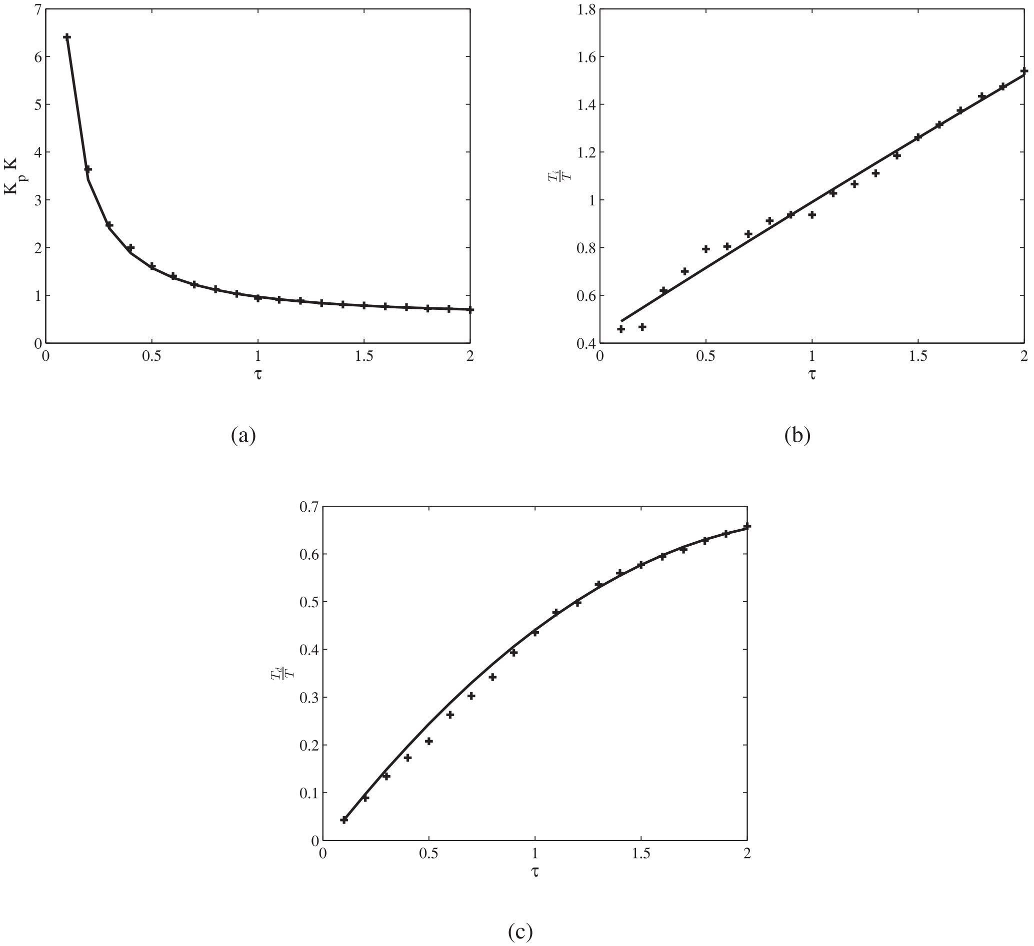


Fig. 8. Tuning parameters for PID controller. Plus sign: optimal values of the parameter. Solid line: fitting function.

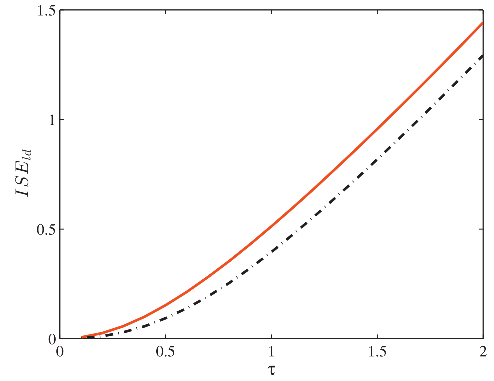
Table 2
Parameters for the proposed tuning rules.

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

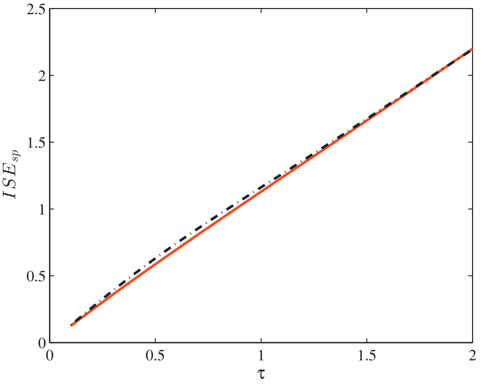
appears that the robustness is increased significantly at the expense of only a slight decrement of the performance in both the set-point and load disturbance step response.

6. Simulation examples

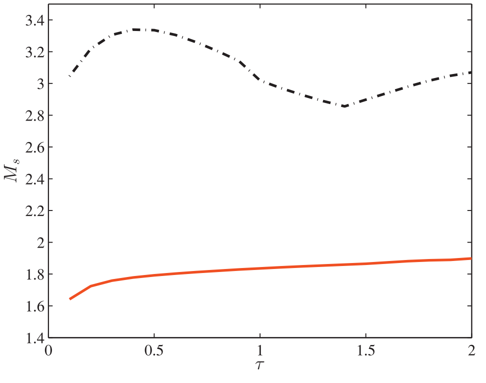
In this section, the tuning rules presented in the previous section are evaluated by considering simulation examples with processes with a different dynamics and a case-study example. A comparison with the tuning rules proposed in [3] is also performed. As the proposal in



(a)



(b)



(c)

Fig. 9. Comparison between Nash tuning (solid red lines) and intermediate tuning with $\alpha = 0.5$ [3] (dashed black lines) for different normalized dead times. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

[3] is based on the ISE performance index, for each process, the ISE in both the servo and the regulatory tasks, are computed. In addition, in order to provide a more global comparison framework, the M_s and the total variation (TV) of the control variable for each task are computed . The TV is a measure of the smoothness of control action and it is defined as

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

6.1. FOPDT system

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

$$P_1(s) = \frac{1}{10s+1} e^{-4s} \tag{20}$$

Table 3
Results related to $P_1(s)$ ($\tau = 0.4$).

Tuning rule	K_p	T_i	T_d	ISE_{sp}	ISE_{ld}	TV_{sp}	TV_{ld}	M_s
$\alpha = 0$ (SP)	2.38	9.54	2.17	4.50	0.91	66.62	2.17	2.49
NS	**1.88**	**6.60**	**1.97**	**4.80**	**1.00**	**48.92**	**1.50**	**1.83**
$\alpha = 0.5$	2.74	5.83	2.23	5.13	0.57	91.60	3.45	3.35
$\alpha = 1$ (LD)	3.58	4.50	2.31	11.71	0.54	208.90	13.49	10.98

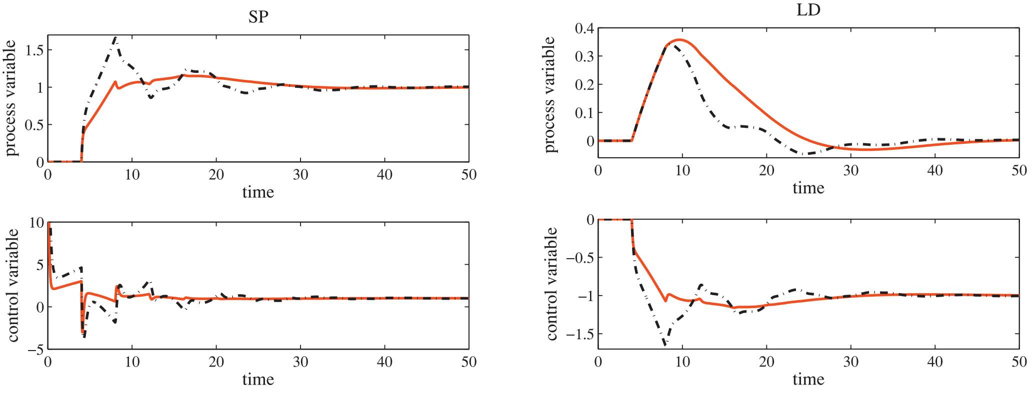


Fig. 10. Set-point (left) and load disturbance (right) step responses for $P_1(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dash-dot line: tuning rules for PID controllers ($\alpha = 0.5$) proposed in [3]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where, evidently, $K = 1$, $T = 10$, and $L = 4$. The parameters resulting from the application of the tuning rules are shown in Table 3. The simulation results related to the both set-point and load disturbance unit step signals are plotted in Fig. 10. The resulting values of the integrated square error, total variation of the control variable (in order to evaluate the control effort) and maximum sensitivity are also shown in Table 3. It can be seen that the NS tuning provides, with respect the extreme-optimal tunings, intermediate performance index values. At the same time, as it is constrained by the optimization procedure, the achieved robustness is substantially higher than the rest. This way the optimal NS tuning provides more than reasonable performance for both operation tasks and at the same time very good robustness.

The step responses obtained with the tuning rules proposed in this paper have been compared with those obtained by applying the tuning rules specifically devised in [3] for $\alpha = 0.5$ (see Fig. 10) which means that the load disturbance and the set-point responses are considered to be equally important (intermediate tuning rules). Further, a comparison with the case $\alpha = 0$ and $\alpha = 1$, which means that only the set-point or load disturbance response, respectively, is relevant, has been also performed. Results are shown in Fig. 11 (dash-dot line) where, in order to show that it is worth developing an intermediate tuning rule, it has also been plotted the results obtained by inverting the use of the tuning rules proposed in [3] that is, the tuning rule devised for the set-point following task is applied to the load disturbance task and vice versa (dotted line).

From the obtained results, it can be seen that the proposed robust intermediate tuning rules generate smoother responses with less control effort. This can be also observed in Table 3, where the values of TV are generally lower in comparison with the other tuning rules. Indeed,

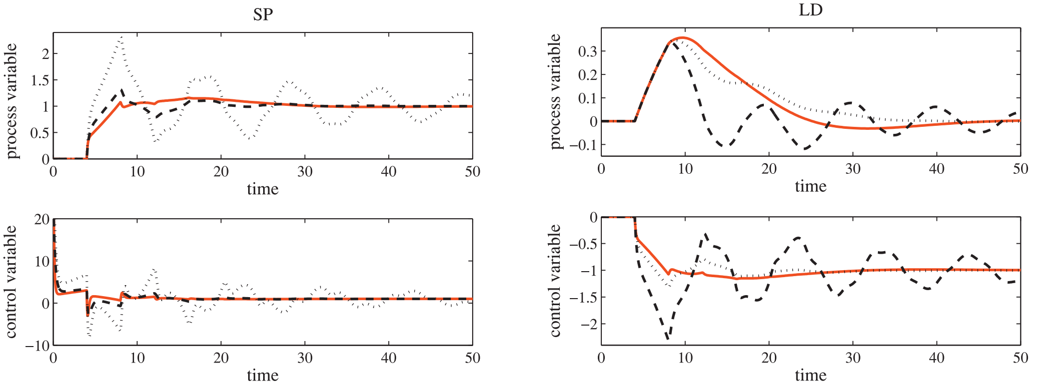


Fig. 11. Set-point (left) and load disturbance (right) step responses for $P_1(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dashed line: tuning rules for servo/regulation PID controllers proposed in [3] applied to the case they have been devised. Dotted line: tuning rules for PID controllers proposed in [3] used for the other control task they have been devised. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Results related to $P_2(s)$ ($\tau = 0.49$).

Tuning rule	K_p	T_i	T_d	ISE_{sp}	ISE_{ld}	TV_{sp}	TV_{ld}	M_s
$\alpha = 0$ (SP)	1.99	2.86	0.75	1.60	0.41	44.11	1.75	2.59
NS	**1.60**	**2.06**	**0.69**	**1.94**	**0.53**	**35.28**	**2.02**	**1.84**
$\alpha = 0.5$	2.31	1.91	0.77	1.96	0.33	53.023	2.87	3.31
$\alpha = 1$ (LD)	2.94	1.52	0.81	3.04	0.32	76.227	5.87	9.27

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

6.2. High-order processes

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

$$P_2(s) = \frac{1}{(s+1)^4} \tag{21}$$

In order to apply the proposed tuning rule, the process has been modelled as a FOPDT with $K = 1$, $T = 2.9$, $L = 1.42$ (thus, the normalized dead time can be determined as $\tau = 0.49$). The results related to both the set-point and load disturbance unit step signals are plotted in Figs. 12 and 13 and the resulting PID parameters as well as performance indexes are summarized in Table 4. From the obtained results it can be appreciated that the proposed approach provides also in this case a balanced performance and the sensitivity value is significantly reduced with respect to the other cases where robustness has not been taken into account explicitly. Therefore, with the proposed multi-objective optimization approach, the performance can be almost maintained whereas the robustness has been increased.

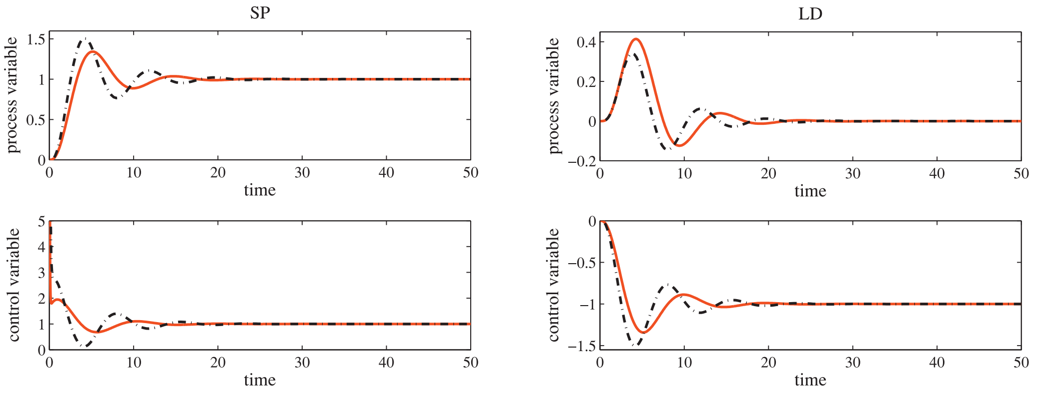


Fig. 12. Set-point (left) and load disturbance (right) step responses for $P_2(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dash-dot line: tuning rules for PID controllers ($\alpha = 0.5$) proposed in [3]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

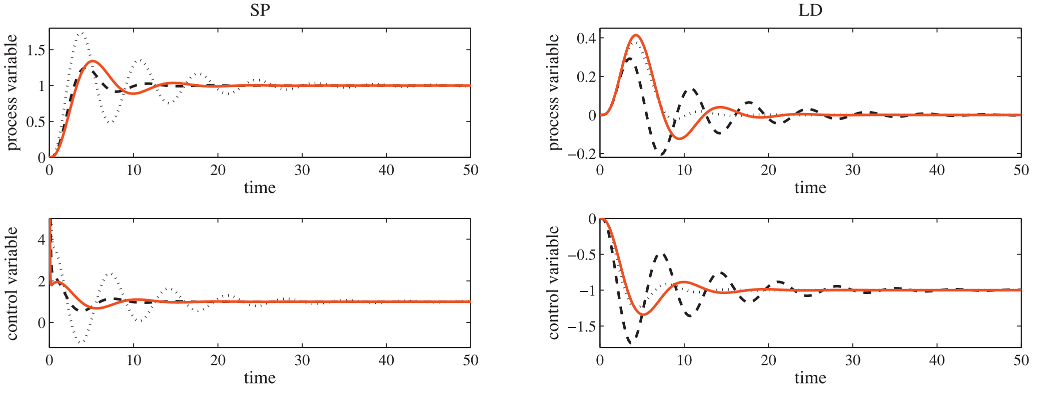


Fig. 13. Set-point (left) and load disturbance (right) step responses for $P_2(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dashed line: tuning rules for servo/regulation PID controllers proposed in [3] applied to the case they have been devised. Dotted line: tuning rules for PID controllers proposed in [3] used for the other control task they have been devised. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

$$P_3(s) = \frac{1}{(s+1)^{20}} \tag{22}$$

In order to apply the proposed tuning rules, the process has been modelled as a FOPDT system with $K = 1$, $T = 7.76$, $L = 12.72$ (thus, the normalized dead time can be determined as $\tau = 1.64$). The results are shown in Table 5, the responses are plotted in Fig. 14 and the resulting PID parameters as well as the performance indexes are summarized in Table 5. As it can be observed, the proposed tuning rules in general provide the minimum ISE for the servo and regulatory task. In the responses shown in Fig. 14, it can be seen that the tuning rule proposed in this work offers an optimal balance in comparison with the extreme cases ($\alpha = 0$ and $\alpha = 1$).

Table 5
Results related to $P_3(s)$ ($\tau = 1.64$).

Tuning rule	K_p	T_i	T_d	ISE_{sp}	ISE_{ld}	TV_{sp}	TV_{ld}	M_s
$\alpha = 0$ (SP)	0.87	11.31	5.40	17.39	11.68	21.75	3.82	2.59
NS	**0.76**	**10.35**	**4.69**	**16.10**	**11.62**	**17.41**	**2.12**	**1.94**
$\alpha = 0.5$	0.91	10.63	5.63	19.03	12.41	23.82	4.83	2.97
$\alpha = 1$ (LD)	1.06	9.43	6.52	67.34	35.69	48.00	18.94	7.66

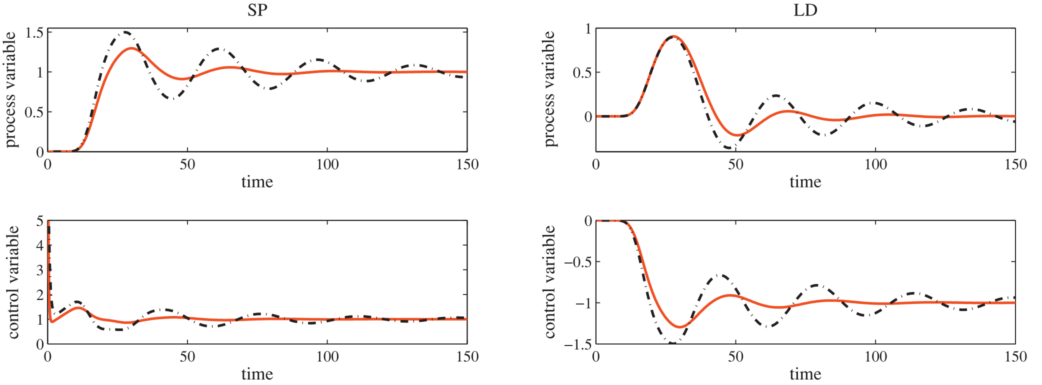


Fig. 14. Set-point (left) and load disturbance (right) step responses for $P_3(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dash-dot line: tuning rules for PID controllers ($\alpha = 0.5$) proposed in [3]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

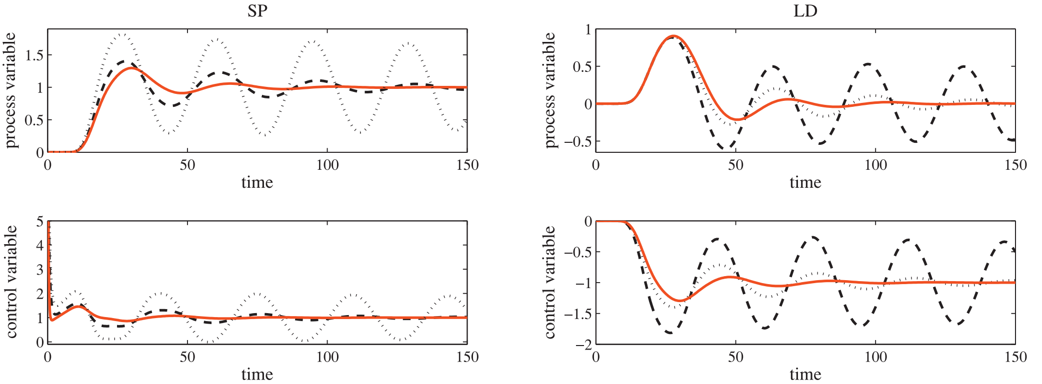


Fig. 15. Set-point (left) and load disturbance (right) step responses for $P_3(s)$. Solid red line: proposed tuning rules (NS) for PID controllers. Dashed line: tuning rules for servo/regulation PID controllers proposed in [3] applied to the case they have been devised. Dotted line: tuning rules for PID controllers proposed in [3] used for the other control task they have been devised. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.3. Case study

In order to demonstrate the effectiveness of the proposed tuning rules, a case study example is also provided. We consider the isothermal Continuous Stirred Tank Reactor (CSTR), where

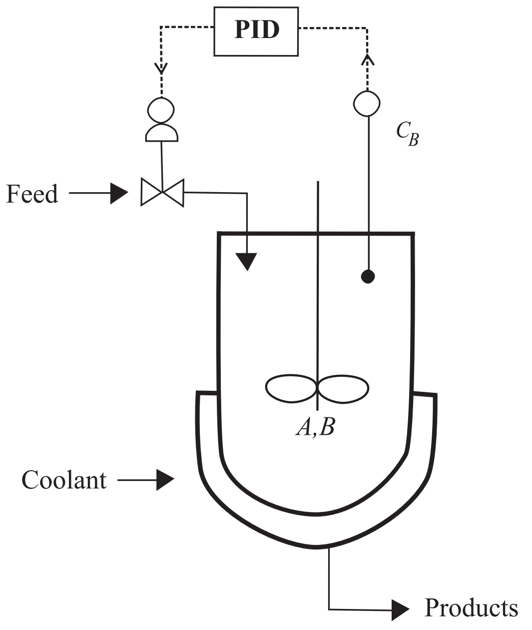


Fig. 16. The CSTR control system.

Table 6
Parameters values of the CSTR control system.

k_1	k_2	$k3$	C_{Ai}	V
5/6 $\min^{-1}$	5/3 $\min^{-1}$	1/6 $\mathrm{lmol}^{-1}\min^{-1}$	10 mol^{-1}	700 l

the isothermal series/parallel Van de Vusse reaction takes place (see [12,22] for details). The rates of formation of the products A and B are assumed to be:

$$r_A = -k_1 C_A - k_3 C_A^2 \tag{23}$$

$$r_B = k_1 C_A - k_2 C_B \tag{24}$$

where C_A and C_B are the reactant concentrations in the reactor, and k_1, k_2, k_3 are the reaction rate constants for the three reactions. The mass balance for A and B is given by

$$\frac{dC_A(t)}{dt} = \frac{F_r(t)}{V}[C_{Ai} - C_A(t)] - k_1 C_A(t) - k_3 C_A^2(t), \tag{25}$$

$$\frac{dC_B(t)}{dt} = -\frac{F_r(t)}{V} C_B(t) + k_1 C_A(t) - k_2 C_B(t) \tag{26}$$

where F_r is the feed flow rate of product A, and V is the reactor volume which is kept constant during the operation. The concentration C_B is considered as the controlled variable, the flow through the reactor F_r as the manipulated variable, and the variation of the concentration C_{Ai} of A in the feed flow is considered as the disturbance. The system under consideration is sketched in Fig. 16 and the values of the parameters are shown in Table 6.

Table 7
Results related to $P_4(s)$ (CSTR system).

Tuning rule	K_p	T_i	T_d	ISE_{sp}	ISE_{ld}	TV_{sp}	TV_{ld}	M_s
$\alpha = 0$ (SP)	3.77	0.70	0.26	76.08	9.20	103.61	25.37	2.32
NS	**3.34**	**0.57**	**0.24**	**82.04**	**10.40**	**108.4**	**24.98**	**1.88**
$\alpha = 0.5$	4.54	0.58	0.27	87.7	5.44	124.74	40.71	3.18
$\alpha = 1$ (LD)	5.36	0.49	0.30	100.21	3.41	226.11	267.18	6.67

Initially the system is at the steady state (operating point) with $C_{Ao} = 2.9175$ mol/l and $C_{Bo} = 1.10$ mol/l. With this information, the measurement range for C_B can be selected from 0 to 1.5714 mol/l and the variation of the flow for the control valve can be selected from 0 to 634.1719 l/min (from 0 to 34.1719 l/min). The signals (y, u and r) will be in percentage from 0 to 100%, so that,

$$y(t)_{\%} = \left(\frac{100}{1.5714}\right) C_B(t) \tag{27}$$

and

$$F_r(t) = \left(\frac{634.1719}{100}\right) u(t)_{\%} \tag{28}$$

where Eq. (27) refers to the sensor–transmitter element, and the control valve with a linear flow characteristic is represented by Eq. (28). It is supposed that the process output can vary in the 0–100% normalized range and that in the normal operating point, the controlled variable has a value close to 70%.

The nonlinear system is approximated by a FOPDT model, using the identification method proposed in [2]:

$$P_4(s) = \frac{0.32}{0.62s + 1} e^{-0.53s} \tag{29}$$

The parameters resulting from the application of the tuning rules are shown in Table 7. The simulation results related to the both set-point and load disturbance unit step signals are shown in Fig. 10. The resulting values of the integrated square error, total variation of the control variable (in order to evaluate the control effort) and maximum sensitivity are also summarized in Table 7. In this case, as with the previous examples, the intermediate performance shows that the NS tuning performance, even if it is intermediate between servo and regulation, is more closer to the servo operations. This way, it improves the servo performance with reference to the $\alpha = 0.5$ case, but not the load disturbance. Therefore this represents a different balance of performances that occurs as a consequence of the increment of performance (and of the nonlinearity of the process).

The process outputs ($y(t)_{\%}$) and the control signal ($u(t)_{\%}$) of the system are shown in Fig. 17, for a set-point step change of -10% and a load disturbance step of +10%. It can seen that the set-point tuning gives the best performance, however the robustness of the system is very low. In fact, the proposed tuning rules (NS) provide the higher level of robustness in comparison with the other tuning methods. Overall, it can be observed that the proposed tuning yields a smooth control action when the system operates in both modes (which means that the same controller can be used effectively in both operation modes).

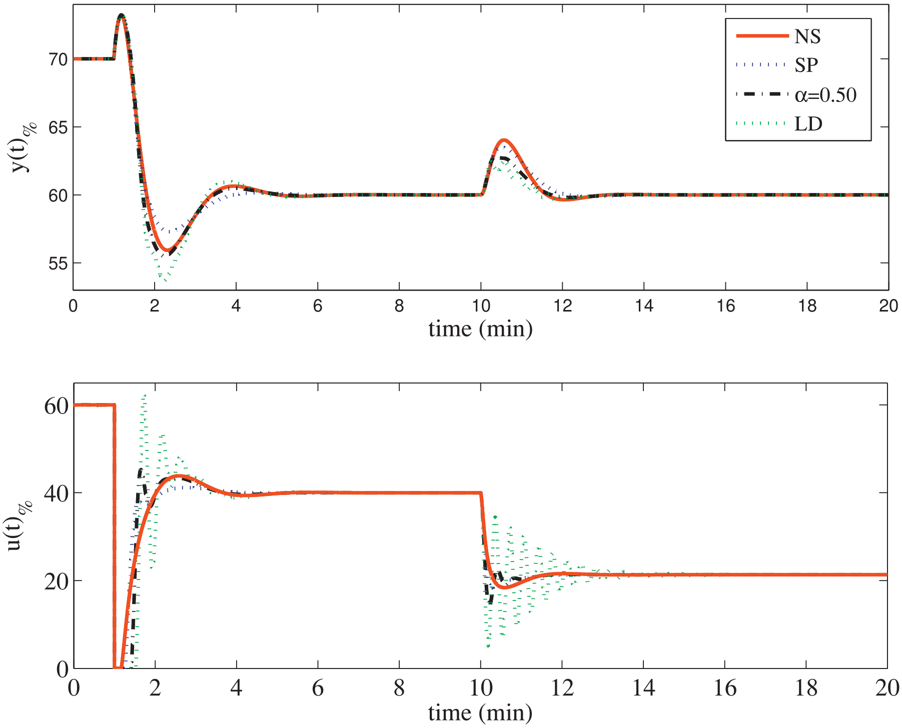


Fig. 17. Process output for the nonlinear control system operating in both operation modes.

7. Conclusions

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

This methodology addresses two different trade-offs: that between the performance and robustness and that between the servo and regulatory modes. It has been shown that, in particular, the robustness of the system can be a critical issue and therefore it has to be included explicitly in the optimization procedure.

The tuning rules developed in this paper are based on the Nash solution and they can be easily implemented in standard industrial controllers. Through the presented examples it can be observed that the devised tuning rules offers a balance between both operation modes with a smooth response and acceptable values of performance and robustness. As the application of the Nash selection provides a good compromise between the set-point following and load disturbance attenuation operation, the work will be now continued by addressing the tuning of second order plus time delay process models. In this case, the derivation of tuning rules is not so immediate as the parameterization in terms of the process model parameters involve more terms and the dependencies are more sensitive and difficult to handle.

Acknowledgment

This work was partially supported by the Spanish Ministry of Economy and Competitiveness program under grants DPI2013-47825-C3-1-R, DPI2016-77271-R and PRX16/00271 and by MINECO and FEDER through the project CICYT HARCRICS (DPI2014-58104-R). This work was carried out during the stay of R. Vilanova as Visiting Professor at the University of Brescia.

References

[1] S. Alcántara, C. Pedret, R. Vilanova, On the model matching approach to PID design: analytical perspective for robust servo/regulator tradeoff tuning, J. Econ. Theory 20 (5) (2010) 596–608.

[2] V.M. Alfaro, Low-order models identification from process reaction curve, Ciencia y Tecnología 24 (2) (2006) 197–216.

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

[4] J. Arévalo, Gradual Nash bargaining with endogenous agenda: a path-dependent model, Colomb. Econ. J. 2 (2004) 189–212.

[5] K.J. Åström, T. Hägglund, PID Controllers: Theory, Design and Tuning, ISA Press, Research Triangle Park, NC, 1995.

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

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

[8] R.J. Aumann, S. Hart, Handbook of Game Theory with Economic Applications, Elsevier, 1994.

[9] C.C. Coello, Evolutionary multi-objective optimization: a historical view of the field, IEEE Comput. Intell. Mag. 1 (1) (2006) 28–36.

[10] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (1998) 631–657.

[11] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of the-Art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.

[12] J.G.V. de Vusse, Plug-flow type reactor versus tank reactor, Chem. Eng. Sci. 19 (1964) 964.

[13] M. Dorigo, T. Stützle, Ant Colony Optimization: Overview and Recent Advances, Springer, US, Boston, MA, 2010, pp. 227–263.

[14] O. Garpinger, T. Hägglund, Software-based optimal PID design with robustness and noise sensitivity constraints, J. Process Control 33 (2015) 90–101.

[15] M. Ge, M. Chiu, Q. Wang, Robust PID controller design via LMI approach, J. Process control 12 (2002) 3–13.

[16] E.N. Goncalves, R.M. Palhares, R.H.C. Takahashi, A novel approach for H_2/H_∞ robust PID synthesis for uncertain systems, J. Process Control 18 (2008) 19–26.

[17] M.-H. Hung, L.-S. Shu, S.-J. Ho, S.-F. Hwang, S.-Y. Ho, A novel intelligent multiobjective simulated annealing algorithm for designing robust PID controllers, IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. 38 (2) (2008) 319–330.

[18] J.F. Nash Jr., The bargaining problem, Econom. J. Econom. Soc. 18 (1950) 155–162.

[19] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. 42 (1) (2014) 21–57.

[20] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.

[21] A. Konak, D.W. Coit, A.E. Smith, Multi-objective optimization using genetic algorithms: a tutorial, Reliab. Eng. Syst. Saf. 91 (9) (2006) 992–1007.

[22] C. Kravaris, P. Daoutidis, Nonlinear state feedback control of second-order nonminimum-phase nonlinear systems, Comput. Chem. Eng. 14 (4) (1990) 439–449.

[23] A. Messac, A. Ismail-Yahaya, C.A. Mattson, The normalized normal constraint method for generating the Pareto frontier, Struct. Multidiscip. Optim. 25 (2) (2003) 86–98.

[24] A. Messac, C.A. Mattson, Generating well-distributed sets of Pareto points for engineering design using physical programming, Optim. Eng. 3 (4) (2002) 431–450.

[25] E. Mezura-Montes, M. Reyes-Sierra, C.A. Coello, Multi-objective Optimization Using Differential Evolution: A Survey of the State-of-the-Art, Springer, Berlin Heidelberg, 2008, pp. 173–196.
[26] K.M. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, 1998.
[27] A. O'Dwyer, Handbook of PI and PID Controller Tuning Rules, second, Imperial College Press, London, UK, 2006.
[28] A. Okada, The Nash bargaining solution in general n-person cooperative games, J. Econ. Theory 145 (6) (2010) 2356–2379.
[29] H. Panagopoulos, K.J. Åström, T. Hägglund, Design of PID controllers based on constrained optimisation, IEE Proc. Control Theory Appl. 149 (1) (2002) 32–40.
[30] V. Pareto, Manuale di Economia Politica, vol. 13, Societa Editrice, 1906.
[31] G. Reynoso-Meza, J. Sanchis, X. Blasco, J.M. Herrero, Multiobjective evolutionary algortihms for multivariable PI controller tuning, Expert Syst. Appl. 39 (2012) 7895–7907.
[32] G. Reynoso-Meza, J. Sanchis, X. Blasco, M. Martínez, Evolutionary algorithms for PID controller tuning: current trends and perspectives (in Spanish), Revista Iberoamericana de Automática e Informática Industrial 10 (3) (2013) 251–268.
[33] G. Reynoso-Meza, J. Sanchis, X. Blasco, M. Martínez, Controller tuning using evolutionary multi-objective optimisation: current trends and applications, Control Eng. Pract. 28 (2014) 58–73.
[34] H. Sánchez, R. Vilanova, Optimality comparison of 2DoF PID implementations, in: Proceedings of the 18th IEEE International Conference on System Theory, Control and Computing, Sinaia, Romania, 2014.
[35] H. Sánchez, A. Visioli, R. Vilanova, Nash tuning for optimal balance of the servo/regulation operation in robust PID control, in: Proceedings of the 23rd Mediterranean Conference on Control and Automation, Torremolinos, Spain, 2015, pp. 715–721.
[36] M. Srinivas, L.M. Patnaik, Genetic algorithms: a survey, Computer 27 (6) (1994) 17–26.
[37] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[38] R. Toscano, A simple robust PI/PID controller design via numerical optimization approach, J. Process Control 15 (2005) 81–88.
[39] R. Vilanova, PID controller tuning rules for robust step response of first-order-plus-dead-time models, in: Proceedings of the American Control Conference, Minneapolis, USA, 2006, pp. 21–27.
[40] R. Vilanova, V.M. Alfaro, O. Arrieta, C. Pedret, Analysis of the claimed robustness for PI/PID robust tuning rules, in: Proceedings of the 18th Mediterranean Conference on Control and Automation, Marraketch, Morrocco, 2010, pp. 658–662.
[41] R. Vilanova, A. Visioli, PID Control in the Third Millennium, Springer, 2012.
[42] A. Visioli, Practical PID Control, Springer, London, UK, 2006.
[43] G.-Q. Zeng, J. Chen, Y.-X. Dai, L.-M. Li, C.-W. Zheng, M.-R. Chen, Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization, Neurocomputing 160 (2015) 173–184.
[44] S.Z. Zhao, M.W. Iruthayarajan, S. Baskar, P.N. Suganthan, Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization, Inf. Sci. 181 (16) (2011) 3323–3335.
[45] A. Zhou, B.-Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evol. Comput. 1 (1) (2011) 32–49.
[46] M. Zhuang, D.P. Atherton, Automatic tuning of optimum PID controllers, IEE Proc. Control Theory Appl. 140 (1993) 216–224.
[47] J.G. Ziegler, N.B. Nichols, Optimum settings for automatic controllers, Trans. ASME 64 (11) (1942) 759–768.