

Document downloaded from:

<http://hdl.handle.net/10251/82090>

This paper must be cited as:

Reynoso Meza, G.; Sanchís Saez, J.; Blasco, X.; Freire, RZ. (2016). Evolutionary multi-objective optimisation with preferences for multivariable PI controller tuning. *Expert Systems with Applications*. 51:120-133. doi:10.1016/j.eswa.2015.11.028.



The final publication is available at

<http://doi.org/10.1016/j.eswa.2015.11.028>

Copyright Elsevier

Additional Information

Evolutionary Multi-Objective Optimisation with preferences for multivariable PI controller tuning

Gilberto Reynoso-Meza ^{b,*}, Javier Sanchis ^a, Xavier Blasco ^a, Roberto Z. Freire ^b

^a*Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, Camino de Vera s/n, Valencia 46022, Spain*

^b*Industrial and Systems Engineering Graduate Program (PPGEPS), Pontificia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, PR, Brazil*

Abstract

Multi-objective optimisation design procedures have shown to be a valuable tool for control engineers. They enable the designer having a close embedment of the tuning process for a wide variety of applications. In such procedures, evolutionary multi-objective optimisation has been extensively used for PI and PID controller tuning; one reason for this is due to their flexibility to include mechanisms in order to enhance convergence and diversity. Although its usability, when dealing with multi-variable processes, the resulting Pareto front approximation might not be useful, due to the number of design objectives stated. That is, a vast region of the objective space might be impractical or useless *a priori*, due to the strong degradation in some of the design objectives. In this paper preference handling techniques are incorporated into the optimisation process, seeking to improve the pertinency of the approximated Pareto front for multi-variable PI controller tuning. That is, the inclusion of preferences into the optimisation process, in order to seek actively for a pertinent Pareto front approximation. With such approach, it is possible to tune a multi-variable PI controller, fulfilling several design objectives, using previous knowledge from the designer on the expected trade-off performance. This is validated with a well-known benchmark example in multi-variable control. Control tests show the usefulness of the proposed approach when compared with other tuning techniques.

Keywords: Multi-objective optimisation, controller tuning, PI tuning, evolutionary multi-objective optimisation, preference handling, many-objective optimisation.

1. Introduction

Intelligent control is a subfield of control systems engineering of growing interest among researchers (Ruano, 2005; Ruano et al., 2014). Nowadays, the most accepted definition for intelligent control comprises using one or several tools from computational intelligence and

*Corresponding author.

Email addresses: g.reynosomeza@pucpr.br (Gilberto Reynoso-Meza), jsanchis@isa.upv.es (Javier Sanchis), xblasco@isa.upv.es (Xavier Blasco), roberto.freire@pucpr.br (Roberto Z. Freire)

soft computing for control engineering purposes. Such tools range from neural networks, fuzzy logic systems and evolutionary algorithms (Albertos, 2007; Ruano, 2007; Tzafestas, 2007) to rule-based and knowledge-based systems (Liao, 2005). Such techniques have shown to be useful in complex instances in control systems engineering (Ruano, 2005).

One of the fundamental tasks in intelligent control is the controller tuning problem (Jiménez et al., 2015; Mishra et al., 2015; Ponce et al., 2015; Sabzi et al., 2016). Such problem consists in finding suitable values for the tuneable parameters of a given control structure. With such parameters it is expected to fulfil some desired closed-loop specifications for a given process. Although there are several control structures, the PI-PID controller remains as a reliable and practical control solution for several industrial processes (Åström and Hägglund, 2001). One of the main advantages of PI-PID controllers is their ease of implementation, giving a good trade-off between simplicity and cost to implement (Tan et al., 2004; Stewart and Samad, 2011). Owing to this, seeking for new tuning techniques is an ongoing research topic (Åström and Hägglund, 2005); current research points to guarantee reasonable stability margins as well as a good overall performance for a wide variety of processes (Vilanova and Alfaro, 2011).

New tuning techniques are being focused on the fulfilment of several objectives and requirements, sometimes in conflict among them (Ang et al., 2005; Li et al., 2006). Some tuning procedures are based on optimisation statements (Ge et al., 2002; Toscano, 2005; Åström et al., 1998; Panagopoulos et al., 2002; Sanchez and Vilanova, 2013) and in some cases they are solved by means of stochastic optimisers. A recently popular approach consists on using Evolutionary Multi-objective Optimisation (EMO) for PI-PID controller tuning for single input, single output (SISO) and multiple input, multiple output (MIMO) processes (Reynoso-Meza et al., 2013b).

In EMO, a simultaneous optimisation approach is used in order to seek for a Pareto set approximation. This Pareto set comprises several solutions, where all they are Pareto-optimal, *i.e.* there is not a solution better than another in all the objectives, but a different trade-off between conflictive objectives. In order to approximate this Pareto set, Multi-Objective Evolutionary Algorithms (MOEAs) are used. Selecting or coding a MOEA is just a part of the overall process; from a practical point of view a multi-criteria decision making (MCDM) stage is required by the decision maker (DM or simply, the designer) in order to select a controller from the approximated Pareto front. Therefore, a Multi-objective Optimisation Design (MOOD) procedure for controller tuning is needed, where the multi-objective problem (MOP) definition, the optimisation process and the MCDM stage are integrated. This procedure has shown to be a valuable tool for control engineers (Reynoso-Meza et al., 2013b, 2014c); it may enable the designer to have a close embedment with the tuning process; with them it is possible to take into account each design objective individually; it also enables the design alternative comparison, to select a controller fulfilling the expected trade-off among conflictive objectives.

This MOOD procedure has been used with success in PI-PID tuning for MIMO processes (Herreros et al., 2002; Hung et al., 2008; Xue et al., 2010; Zhao et al., 2011; Reynoso-Meza et al., 2012). As noticed in (Reynoso-Meza et al., 2014c), mechanisms to improve

convergence, diversity and constraint handling have been included in MOEA for this purpose; the following step seems to be related with improving pertinency of solutions by means of the statement of designer's preferences. These mechanisms will enable the algorithm to approximate a Pareto front with pertinent solutions in the search process; furthermore, they could facilitate the DM's task of analysing and selecting a design alternative (Coello, 2000). This feature has not been widely exploited and could be helpful to solve efficiently many-objective optimisation instances (Ishibuchi et al., 2008) for multi-variable PI controller tuning. In such instances, each control loop and the overall system must fulfill several performance specifications.

The aim of this paper is twofold. On the one hand, stating a general *MOP/EMO* definition to deal efficiently with MIMO processes using designer's preferences; despite the fact that every process is different and the designer would prefer stating its own meaningful objectives, a general procedure could be valuable, where a pertinent Pareto front is provided for the further analysis in the MCDM stage. On the other hand, comparing preference handling techniques for EMO and evaluate their performance for PI controller tuning in MIMO processes. In both cases, this paper follow the assumption that the DM has already decided to use a MOOD procedure for controller tuning and the desired objectives have been selected.

The remainder of this paper is as follows: in Section 2 a basic background on multivariable PI control, EMO and preference handling will be provided; in Section 3 a *MOP/EMO* procedure for multivariable PI controller tuning will be stated; Section 4 will be dedicated to solve a benchmark setup based on the Wood & Berry distillation column (Berry, 1973; Wood and Berry, 1973); Finally, some concluding remarks are given.

2. Background

Some notions on multivariable PI control, multi-objective optimisation, and preferences handling are required. They are provided below.

2.1. Background on multivariable PI controller tuning

A basic control loop is depicted in Figure 1. It comprises transfer functions $P(s)$ and $C(s)$ of a process and a controller respectively. The objective of this control loop is to keep the desired output $Y(s)$ of the process $P(s)$ in the desired reference $R(s)$.

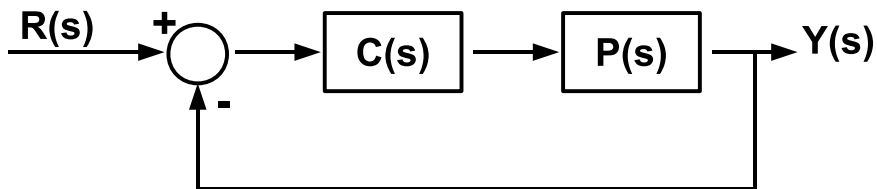


Figure 1: Basic control loop.

In the case of a $N \times N$ MIMO process, $P(s)$ is composed of several sub-processes P_{ij} with $i, j \in \{1, \dots, N\}$ and it has the following structure:

$$P(s) = \begin{bmatrix} P_{11}(s) & \dots & P_{1N}(s) \\ \vdots & \ddots & \vdots \\ P_{N1}(s) & \dots & P_{NN}(s) \end{bmatrix} \quad (1)$$

The complexity of a process like this is mainly due to its coupling effects between inputs and outputs. There are several alternatives to control a MIMO system, and the selection of one technique over another depends on the desired balance between complexity and tradeoff between design specifications. PI controllers are simple but successful solutions, and their performance can be improved with complementary techniques (Åström and Hägglund, 2005); because of this, they are used in this work. The decoupled PI controller $C(s)$ has N SISO PI controllers:

$$C(s) = \begin{bmatrix} C_1(s) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_N(s) \end{bmatrix} \quad (2)$$

Equation (3) shows the transfer function of the selected structure of the PI controller:

$$C_i(s) = k_p \left(1 + \frac{1}{T_i s} \right) \quad (3)$$

where $i \in \{1, \dots, N\}$, k_p is the proportional gain, T_i the integral time (s). The control problem consists in selecting proper gains k_p and $k_i = \frac{k_p}{T_i}$ for each one of the PI controllers $C_i(s)$ in order to achieve a desirable performance of the process $P(s)$ in the control loop as well as robust stability margins. This control problem is well known and it has been addressed with several techniques. Given the coupling effects among sub-processes $P_{ij}(s)$, conflictive objectives may appear, at least related with the performance of each individual control loop. For this reason, EMO techniques could be appealing for PI controller tuning.

2.2. Multi-objective optimisation statement

As referred in (Miettinen, 1998), a MOP with m objectives ¹, can be stated as follows:

$$\min_{\boldsymbol{\theta}} \mathbf{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), \dots, J_m(\boldsymbol{\theta})] \quad (4)$$

subject to:

$$\mathbf{K}(\boldsymbol{\theta}) \leq 0 \quad (5)$$

$$\mathbf{L}(\boldsymbol{\theta}) = 0 \quad (6)$$

$$\underline{\theta}_i \leq \theta_i \leq \overline{\theta}_i, i = [1, \dots, n] \quad (7)$$

¹A maximisation problem can be converted to a minimisation problem. For each of the objectives that have to be maximised, the transformation: $\max J_i(\boldsymbol{\theta}) = -\min(-J_i(\boldsymbol{\theta}))$ could be applied.

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$ is defined as the decision vector with $\dim(\boldsymbol{\theta}) = n$; $\mathbf{J}(\boldsymbol{\theta})$ as the objective vector and $\mathbf{K}(\boldsymbol{\theta})$, $\mathbf{L}(\boldsymbol{\theta})$ as the inequality and equality constraint vectors respectively; $\underline{\theta}_i, \bar{\theta}_i$ are the lower and the upper bounds in the decision space.

It has been noticed that there is not a single solution in MOPs, because there is not generally a better solution in all the objectives. Therefore, a set of solutions, the Pareto set, is defined. Each solution in the Pareto set defines an objective vector in the Pareto front. All the solutions in the Pareto front are a set of Pareto optimal and non-dominated solutions:

Definition 1. (*Pareto optimality (Miettinen, 1998)*): An objective vector $\mathbf{J}(\boldsymbol{\theta}^1)$ is Pareto optimal if there does not exist another objective vector $\mathbf{J}(\boldsymbol{\theta}^2)$ such that $J_i(\boldsymbol{\theta}^2) \leq J_i(\boldsymbol{\theta}^1)$ for all $i \in [1, 2, \dots, m]$ and $J_j(\boldsymbol{\theta}^2) < J_j(\boldsymbol{\theta}^1)$ for at least one j , $j \in [1, 2, \dots, m]$.

Definition 2. (*Dominance (Coello and Lamont, 2004)*): An objective vector $\mathbf{J}(\boldsymbol{\theta}^1)$ is dominated by another objective vector $\mathbf{J}(\boldsymbol{\theta}^2)$ iff $J_i(\boldsymbol{\theta}^2) \leq J_i(\boldsymbol{\theta}^1)$ for all $i \in [1, 2, \dots, m]$ and $J_j(\boldsymbol{\theta}^2) < J_j(\boldsymbol{\theta}^1)$ for at least one j , $j \in [1, 2, \dots, m]$. This is denoted as $\mathbf{J}(\boldsymbol{\theta}_2) \preceq \mathbf{J}(\boldsymbol{\theta}_1)$.

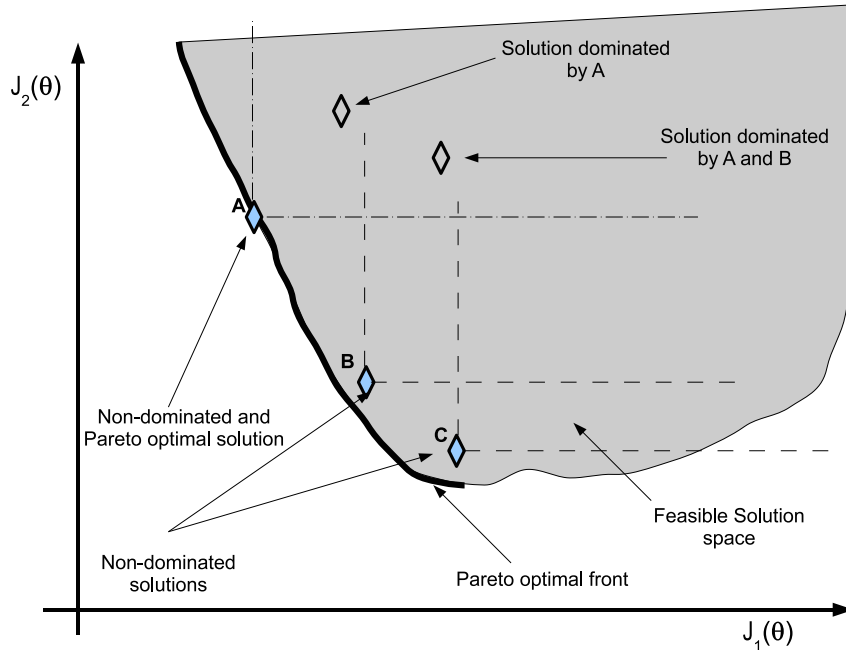


Figure 2: Pareto optimality and dominance concepts.

Within the context of the decoupled PI controller $C(s)$ of equation 2 for a multivariable process $P(s)$ (equation 1), the decision space is defined as:

$$\boldsymbol{\theta} = [k_{p1}, T_{i1}, \dots, k_{pN}, T_{iN}] \in \mathbb{R}^{2N} \quad (8)$$

It is important to notice that the Pareto front is usually unknown, and the DM can only rely on Pareto front approximations. It has been noticed that the MOOD procedure

should be holistic (Bonissone et al., 2009), nevertheless this paper would focus on the MOP definition and the EMO step. With this aim a preference handling procedure will be stated, that might support the holistic essence of this design procedure.

2.3. Background on preference handling in multi-objective optimisation

As suggested before, one potentially desirable characteristic of a MOEA is the mechanism for preference handling in order to calculate pertinent solutions. That is, the capacity to obtain a set of interesting solutions from the DM's point of view (Figure 3). Incorporating the DM's preferences into MOEAs has been suggested to improve the pertinency of solutions (Coello, 2000; Cvetkovic and Parmee, 2002) and hence, facilitating the MCDM task. For the specific application of PI controller tuning, seems to be the next step for EMO (Reynoso-Meza et al., 2014c).

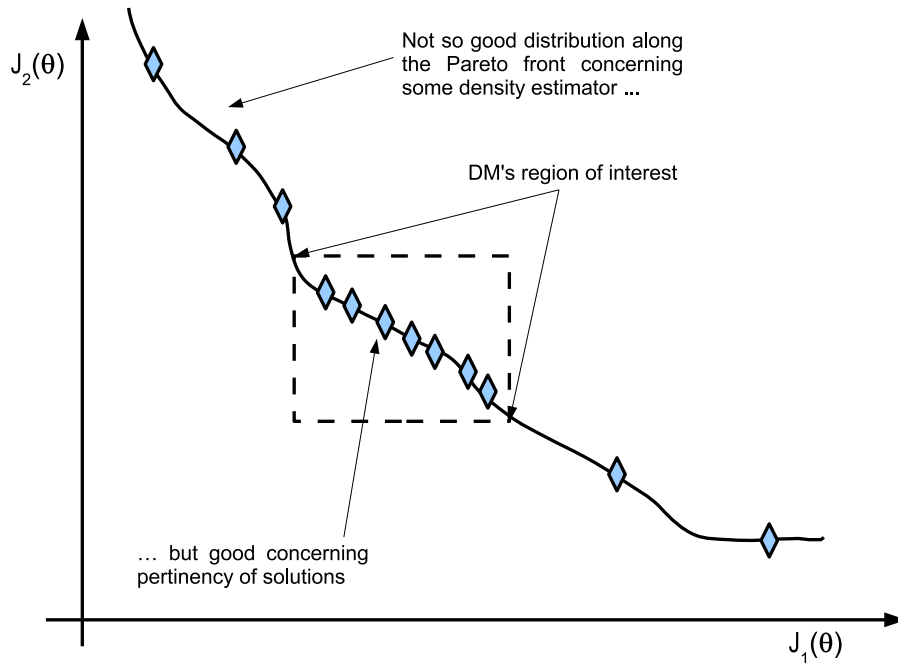


Figure 3: Pertinency concept of design alternatives.

The designer's preferences could be defined in the MOOD procedure in an *a priori*, *progressive*, or *a posteriori* fashion (Munro and Aouni, 2012).

- *A priori*: In such cases, the DM could be interested in using an algorithm that enables incorporating such preferences in the optimisation procedure.
- *Progressive*: the optimisation algorithm embeds the designer into the optimisation process to adjust or change his or her preferences *on the fly*.
- *A posteriori*: According to the set of solutions, the designer defines the preferences in order to select a preferable solution.

It is also possible to classify preference handling techniques into five classes with respect to the question: *what it is important for the designer?* (Reynoso-Meza et al., 2014a):

- *Dominance is essential*: it is important for the designer to calculate a set of solutions that dominate one or more *reference* objective vectors.
- *Objective against objective*: it is important for the designer to identify which objectives have priority over others through the EMO process.
- *Objective value against objective value*: it is important for the designer to identify when the value of a given objective has priority over the value of others.
- *Subset against subset*: identifying a combination of objectives and values that are preferred over others.

As noticed before, PI-PID controller are a common solution for industrial processes, and as consequence several tuning techniques and procedures are available for control engineers. Therefore, the tuning problem is not about finding a solution, but finding a solution with the desirable trade-off. Because of this, *a priori* techniques will be compatible with controller tuning within the MOOD context, given that usually it would be an initial solution available to work with. In this sense, *dominance is essential*, given that an initial solution is available, and it is necessary to improve it. Interactive techniques are outside this work and will be proposed for future work. It will be assumed that exists such reference case controller, which allows to have an idea about what it is important to optimise and get the desirable trade-off region for the designer.

3. Proposal: Methods and Tools

According to Reynoso-Meza et al. (2014c), while several works focus on PI-PID controller tuning using EMO, few of them deal with MIMO instances. Furthermore, few of them use some mechanism for pertinency improvement in many-objective optimisation statements for these problems. Therefore, it is justified to develop a proposal for such processes. In order to state a useful framework, a multi-objective problem definition, preferences statement and a sampling procedure are required.

3.1. Multi-objective problem definition

The non-convex optimisation problem stated in Åström et al. (1998) is used as guideline for the MOP problem definition. It states the optimisation of the integral gain $J_{k_i}(\boldsymbol{\theta}) = -\frac{k_p}{T_i}$ of a PI controller, subject to given values of the maximum sensitivity function $M_s = J_{M_s}(\boldsymbol{\theta})$ (Equation (9)) and/or the maximum complementary sensitivity function $M_p = J_{M_p}(\boldsymbol{\theta})$ (Equation (10)) for a SISO process.

$$J_{M_s}(\boldsymbol{\theta}) = \|(1 + P(s)C(s))^{-1}\|_{\infty} \quad (9)$$

$$J_{M_p}(\boldsymbol{\theta}) = \|P(s)C(s)(1 + P(s)C(s))^{-1}\|_{\infty} \quad (10)$$

According to this, a MOP statement for MIMO processes could be defined, using as control design objectives k_i , M_s and M_p (Reynoso-Meza et al., 2012); that is, it will be stated design objectives k_i , M_s and M_p for each control loop. Such objectives give the DM some insight regarding the tradeoff for robustness, load rejection, and set point response (Åström et al., 1998). It is well-known that certain practical limits to M_s and M_p values are needed to guarantee a minimum margin of robustness:

$$\begin{aligned} 1.2 &\leq M_s \leq 2.0 \\ 1 &\leq M_p \leq 1.5 \end{aligned} \quad (11)$$

In order to apply this tuning procedure in a multivariable process with success, an index to measure the overall MIMO system robustness is required. Here, the closed loop log modulus (L_{cm}) will be used as a robustness indicator. This index is used in the well-known Biggest Log Modulus Tuning (BLT) criterion for diagonal PID controllers in MIMO processes (Luyben, 1986). The index is defined as:

$$L_{cm} = 20 \log \left| \frac{W(s)}{1 + W(s)} \right| \leq L_{cm}^{max} \quad (12)$$

where $W(s) = -1 + \det(I + P(s)C(s))$. The BLT criterion proposes a *de-tuning* of the proportional gains of each controller obtained by the Ziegler-Nichols method, in order to fulfill a maximum value of the closed loop log modulus L_{cm}^{max} . It has been suggested that an empirical value of $L_{cm}^{max} = 2N$ is adequate for such purpose (Luyben, 1986).

Therefore, the MOP at hand is to find a tradeoff solution $\boldsymbol{\theta} = [k_{p1}, T_{i1}, \dots, k_{pN}, T_{iN}]$ such that:

$$\min_{\boldsymbol{\theta}} \mathbf{J}(\boldsymbol{\theta}) = [-k_{i1}, \dots, -k_{iN}, L_{cm}, M_{s1}, \dots, M_{sN}, M_{p1}, \dots, M_{pN}] \in \mathbb{R}^{3N+1} \quad (13)$$

subject to:

$$\begin{aligned} 1.2 &\leq M_{s1, \dots, N} \leq 2.0 \\ 1 &\leq M_{p1, \dots, N} \leq 1.5 \\ 0 &\leq L_{cm} \leq 2N \end{aligned} \quad (14)$$

Constraints concerning bounds on decision variables will be addressed in Section 3.3. The objective vector as defined by Equation (13), in spite of its usefulness (Reynoso-Meza et al., 2012), it leads to a many-objectives optimisation instance which could increase the complexity of the MCDM stage. A preference handling mechanism, in order to improve the pertinency of the Pareto front, will be useful to overcome such difficulties.

3.2. Preference Handling Mechanism

As noticed in Reynoso-Meza et al. (2014c), MOOD procedures for controller tuning should compete with several well established tuning procedures. In the case of PI controller tuning, different tuning rules are available. Therefore it is possible to assume that a solution already exists and it is possible to use it as a preferred solution that shall be improved. Therefore, among the schemes for preference handling discussed above, *a priori* techniques, within the classification *dominance is essential* can be used. A suitable mechanism for this is the global physical programming explained in Reynoso-Meza et al. (2014a), based on physical programming (Messac, 1996).

The physical programming (PP) method is a suitable technique for multi-objective engineering design since it formulates design objectives in an understandable and intuitive language for designers. PP is an aggregate objective function (AOF) technique (Mattson and Messac, 2005) for multi-objective problems that includes the available information in the optimisation phase. This enables the designer to express preferences relative to each objective function with more detail. Firstly, PP translates the designer's knowledge into classes² with previously defined ranges.³ This preference set reveals the DM's wishes using physical units for each of the objectives in the MOP. From this point of view, the problem is moved to a different range where all the variables are independent of the original MOP (see Figure 4).

For each objective and set of preferences \mathfrak{P} , a class function $\eta_q(\mathbf{J}(\boldsymbol{\theta}))|_{\mathfrak{P}}$, $q = [1, \dots, m]$ is built to translate each objective $J_q(\boldsymbol{\theta})$ to a new range where all the objectives are equivalent to each other. A PP index $J_{pp}(\mathbf{J}(\boldsymbol{\theta})) = \sum_{q=1}^m \eta_q(\mathbf{J}(\boldsymbol{\theta}))$ is then calculated.

In Reynoso-Meza et al. (2014a) the $J_{pp}(\mathbf{J}(\boldsymbol{\theta}))$ index is modified, and a global PP (GPP) index $J_{gpp}(\boldsymbol{\varphi})$ is defined for a given objective vector $\boldsymbol{\varphi}$ as:

$$J_{gpp}(\boldsymbol{\varphi}) = \sum_{q=1}^m \eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}} \quad (15)$$

Such index is helpful for pruning mechanism in MOEAs, in order to improve the pertinency of the solutions according to the predefined preferences. Preference ranges of Figure 4 are defined for the sake of flexibility as in Reynoso-Meza et al. (2014a) to evolve the population to a pertinent Pareto front. Besides, they will allow the following useful definitions (see Figure 5):

T_Vector: $\mathbf{J}^T = [J_1^3, J_2^3, \dots, J_m^3]$, *i.e.* the vector with the maximum value for each objective in the tolerable range.

²The original method states 4 classes: 1S (smaller is better); 2S (larger is better); 3S (a value is better); and 4S (a range is better)

³According to the original method: highly desirable (HD), desirable (D), tolerable (T), undesirable (U) and highly undesirable (HU)

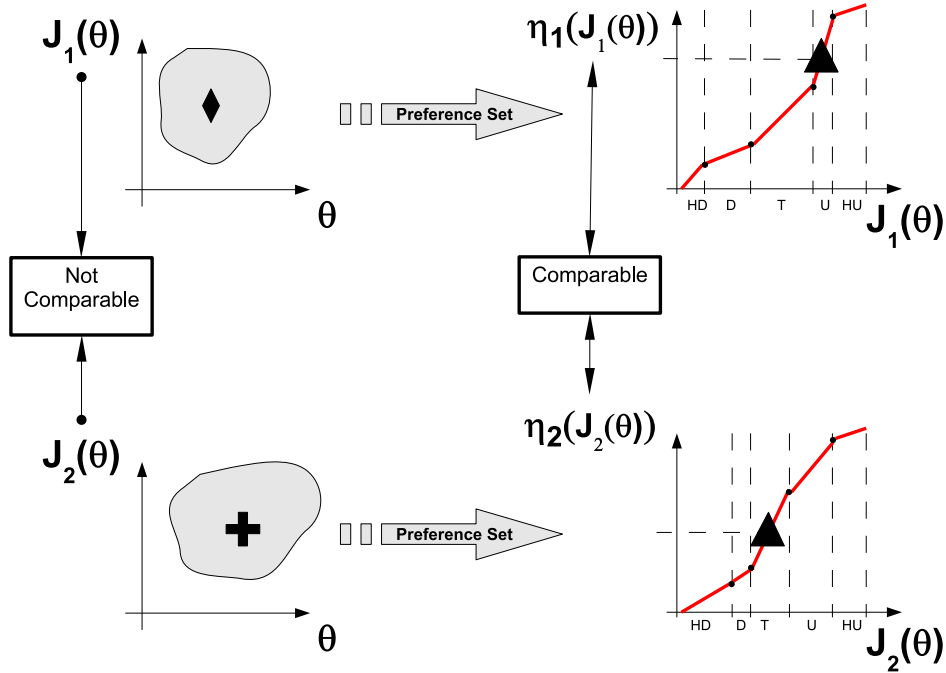


Figure 4: Physical programming (PP) notion. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

D_Vector: $\mathbf{J}^D = [J_1^2, J_2^2, \dots, J_m^2]$, *i.e.* the vector with the maximum value for each objective in the desirable range.

HD_Vector: $\mathbf{J}^{HD} = [J_1^1, J_2^1, \dots, J_m^1]$, *i.e.* the vector with the maximum value for each objective in the highly desirable range.

T_HypV: The hypervolume of the Pareto front approximation bounded by \mathbf{J}^T .

D_Hypv: The hypervolume of the Pareto front approximation bounded by \mathbf{J}^D .

HD_HypV: The hypervolume of the Pareto front approximation bounded by \mathbf{J}^{HD} .

From a practical point of view, the tolerable vector **T_Vector** could be defined as the performance of an available tuning procedure, and the **D_Vector**, **HD_Vector** as the following aspiration levels. This makes this proposal fully compatible with the classification *dominance is essential* and provides a path for the evolution process, in order to continue improving the pertinency of the Pareto front, according to the designer's preferences. Next, a proposal for the automatic statement of preferences for multivariable PI controllers will be provided.

3.3. Preference statement for MIMO processes

Although the designer could need stating specific preferences for a specific process, automatic guidelines would be also valuable, in the same way that general controller tuning rules

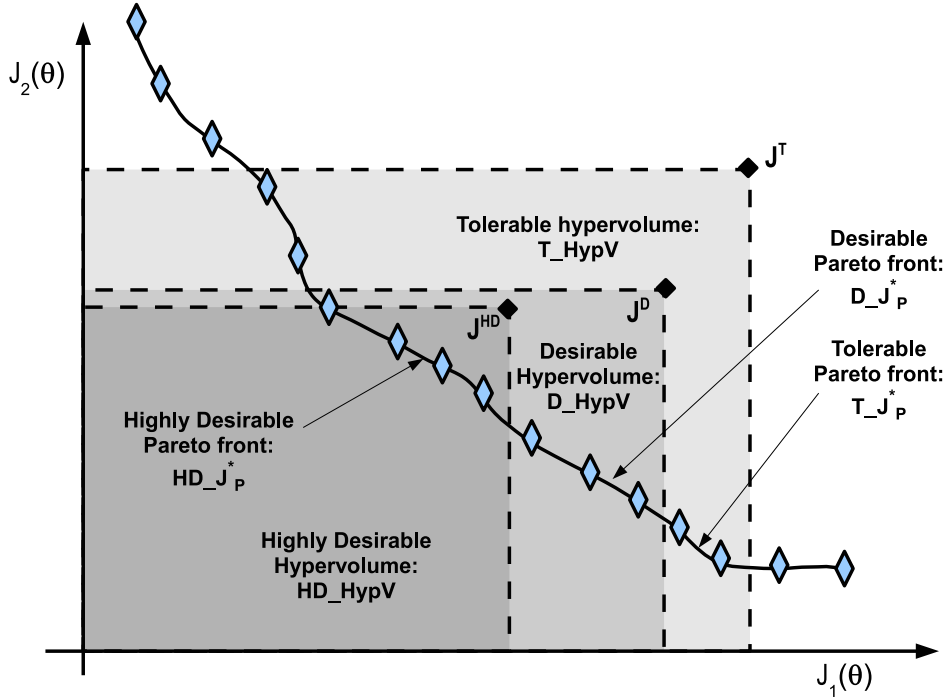


Figure 5: Graphical representation of the definitions stated.

are. That is, most of the times the control engineer needs to mind several control loops, and the availability of a general procedure will be helpful in order to adjust *automatically* control loops with less priority; consequently, the designer can focus in the remainder control loops (the ones with higher priority) which demand more control expertise and engineering thinking (Robinson, 1998).

The BLT procedure, easy to understand and to implement in MIMO processes (Johnson and Moradi, 2005), could provide a suitable *automatic* statement of preferences. Let it be θ_0 the gains of the multivariable PI controller $C(s)$ calculated with the BLT procedure for a given multivariable process $P(s)$. Thus, it is expected to achieve a solution with better $J_1(\theta_0), \dots, J_N(\theta_0)$ than the BLT (θ_0) solution, with the constraints $(M_{s1}, \dots, M_{sN}) < 2.0$, $(M_{p1}, \dots, M_{p2}) < 1.5$ and $L_{cm} < 2N$. According to this, it is possible to define the **T_Vector** as:

$$\mathbf{J}^T = [\overbrace{J_1(\theta_0), \dots, J_N(\theta_0)}^N, 2N, \overbrace{2, \dots, 2}^N, \overbrace{1.5, \dots, 1.5}^N] \quad (16)$$

From here, a preference range \mathfrak{P}_k , for objective $\mathbf{J}_k(\theta)$, $k \in 1, \dots, m$ is built according to Equation (17):

$$\mathfrak{P}_k = \mathbf{J}_k^T \cdot [a_{k1}, a_{k2}, a_{k3}, 1, a_{k4}, a_{k5}] \quad (17)$$

with $a_{k1} > a_{k2} > a_{k3} > 1 > a_{k4} > a_{k5}$ for objectives $J_1(\theta), \dots, J_N(\theta_0)$ ($-k_i < 0$ shall be minimised) and $a_{k1} < a_{k2} < a_{k3} < 1 < a_{k4} < a_{k5}$ for the others. According to these rules, a

Table 1: Preferences Set for multivariable PI controller tuning. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Set											
Objective	\leftarrow J_i^0	HD	$\rightarrow \leftarrow$ J_i^1	D	$\rightarrow \leftarrow$ J_i^2	T	$\rightarrow \leftarrow$ J_i^3	U	$\rightarrow \leftarrow$ J_i^4	HU	\rightarrow J_i^5
$J_1(\boldsymbol{\theta})$	$a_{1_1} \cdot J_1(\boldsymbol{\theta}_0)$		$a_{1_2} \cdot J_1(\boldsymbol{\theta}_0)$		$a_{1_3} \cdot J_1(\boldsymbol{\theta}_0)$		$J_1(\boldsymbol{\theta}_0)$		$a_{1_4} \cdot J_1(\boldsymbol{\theta}_0)$		$a_{1_5} \cdot J_1(\boldsymbol{\theta}_0)$
\vdots											
$J_N(\boldsymbol{\theta}_0)$	$a_{N_1} \cdot J_N(\boldsymbol{\theta}_0)$		$a_{N_2} \cdot J_N(\boldsymbol{\theta}_0)$		$a_{N_3} \cdot J_N(\boldsymbol{\theta}_0)$		$J_N(\boldsymbol{\theta}_0)$		$a_{N_4} \cdot J_N(\boldsymbol{\theta}_0)$		$a_{N_5} \cdot J_N(\boldsymbol{\theta}_0)$
$J_{Lcm}(\boldsymbol{\theta})$	$a_{N+1_1} \cdot (2N)$		$a_{N+1_2} \cdot (2N)$		$a_{N+1_3} \cdot (2N)$		$2N$		$a_{N+1_4} \cdot (2N)$		$a_{N+1_5} \cdot (2N)$
$J_{M_{s1}}(\boldsymbol{\theta})$	$2 \cdot a_{N+2_1}$		$2 \cdot a_{N+2_2}$		$2 \cdot a_{N+2_3}$		2		$2 \cdot a_{N+2_4}$		$2 \cdot a_{N+2_5}$
\vdots											
$J_{M_{sN}}(\boldsymbol{\theta})$	$2 \cdot a_{2N+1_1}$		$2 \cdot a_{2N+1_2}$		$2 \cdot a_{2N+1_3}$		2		$2 \cdot a_{2N+1_4}$		$2 \cdot a_{2N+1_5}$
$J_{M_{p1}}(\boldsymbol{\theta})$	$1.5 \cdot a_{2N+2_1}$		$1.5 \cdot a_{2N+2_2}$		$1.5 \cdot a_{2N+2_3}$		1.5		$1.5 \cdot a_{2N+2_4}$		$1.5 \cdot a_{2N+2_5}$
\vdots											
$J_{M_{pN}}(\boldsymbol{\theta})$	$1.5 \cdot a_{3N+1_1}$		$1.5 \cdot a_{3N+1_2}$		$1.5 \cdot a_{3N+1_3}$		1.5		$1.5 \cdot a_{3N+1_4}$		$1.5 \cdot a_{3N+1_5}$
$G_1(\boldsymbol{\theta})$	<0.0		<0.0		<0.0		0.0		1.0		2.0

preference set could be stated (see Table 1). An additional constraint $G_1(\boldsymbol{\theta}) = Re[\lambda]_{max} < 0$ for the eigenvalues λ of the closed loop is included, in order to guarantee internal stability of the multivariable control loop.

3.4. Controller sampling

In general, stochastic optimisers are characterised by the randomness used in the search process, which is helpful to avoid local minima. Evolutionary or nature inspired algorithms are very popular stochastic optimisers and as noticed before, they have been used extensively for PID-like controller tuning (Reynoso-Meza et al., 2013b). Nevertheless, in such case, this process has to be carefully performed; besides the problematic (and often uncommented) preliminary step of defining the boundaries in the search process, not all parameter combinations in a PID guarantees closed loop stability given a nominal process. It is stated that, given a subset $\mathbb{S} \in \mathbb{R}^3$ and a process $P(s)$, two important features for PID $C(s)$ parameter sampling from \mathbb{S} should be fulfilled:

1. Any sampled controller $C(s)$ from \mathbb{S} must stabilise the closed loop.
2. Any stabilising controller $C(s)$ of the process $P(s)$ must be contained in \mathbb{S} .

A common approach for feature 1 is to define bounds on the parameter which avoid all non-stable but also some stable PID parameters; therefore, feature 2 is not fulfilled. A second alternative, is to bound the search space with all stable PID parameters, but including non-stable parameters, which are verified while the algorithm is running. In such case, feature 1 is unfulfilled, and computational (CPU time) resources could be misspent.

According to Silva et al. (2002), the set $\mathbb{C} \subset \mathbb{R}^3$ of stabilising controllers for a first order plus deadtime (FOPDT) $P(s)$ process is given by subsets \mathbf{T} , $\mathbf{\Delta}$ and \mathbf{Q} (Figure 6).

The range of k_p values for which $P(s)$ is stable is given by:

$$-\frac{1}{k} < k_p < k_u \quad (18)$$

where k_u is usually known as the ultimate gain

$$k_u = \frac{1}{k} \left[\frac{T}{L} \alpha_1 \sin(\alpha_1) - \cos(\alpha_1) \right] \quad (19)$$

and α_1 is the solution of the equation

$$\tan(\alpha) = -\frac{T}{T+L} \alpha, \alpha \in [0, \pi] \quad (20)$$

The complete stabilisation region is given by:

1. For each $k_p \in [-\frac{1}{k}, \frac{1}{k})$, the corresponding region in (k_i, k_d) space is the quadrilateral \mathbf{Q} of Figure 6.
2. For $k_p = \frac{1}{k}$, the corresponding region in (k_i, k_d) space is the triangle $\mathbf{\Delta}$ of Figure 6.
3. For each $k_p \in [\frac{1}{k}, k_u)$, the corresponding region in (k_i, k_d) space is the trapezoid \mathbf{T} of Figure 6, where $k_u = \frac{1}{k} \left[\frac{T}{L} \alpha_1 \sin(\alpha_1) - \cos(\alpha_1) \right]$.

Where relevant variables m_j, b_j, w_j for $j = [1, 2]$ are calculated as follows:

$$m_j = \frac{L^2}{z_j^2} \quad (21)$$

$$b_j = -\frac{L}{k \cdot z_j} \left[\sin(z_j) + \frac{T}{L} z_j \cos(z_j) \right] \quad (22)$$

$$w_j = \frac{z_j (1 + k k_p) [1 + \cos(z_j)]}{kL \sin(z_j)} \quad (23)$$

with z_1, z_2 being the roots of

$$k \cdot k_p + \cos(z) - \frac{T}{L} z \sin(z) = 0 \quad (24)$$

Therefore, set \mathbb{C} is composed by subsets \mathbf{T} (for $k_p \in [-\frac{1}{k}, \frac{1}{k})$), $\mathbf{\Delta}$ (for $k_p = \frac{1}{k}$) and \mathbf{Q} (for $k_p \in [\frac{1}{k}, k_u)$).

To sample stabilising controllers fulfilling both aforementioned features, it is required to sample on \mathbb{C} . It is assumed that controllers matching $k_p > 0$, $k_i > 0$, $k_d > 0$ are the only accepted by the designer to stabilise $P(s)$. According to this an injective function $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ to map from the unitary cube to \mathbb{C} is defined (Reynoso-Meza et al., 2014b):

$$\begin{aligned} F(a, b, c) &= [k_p, k_i, k_d] \\ \{a, b, c\} &\in [0, 1] \end{aligned} \quad (25)$$

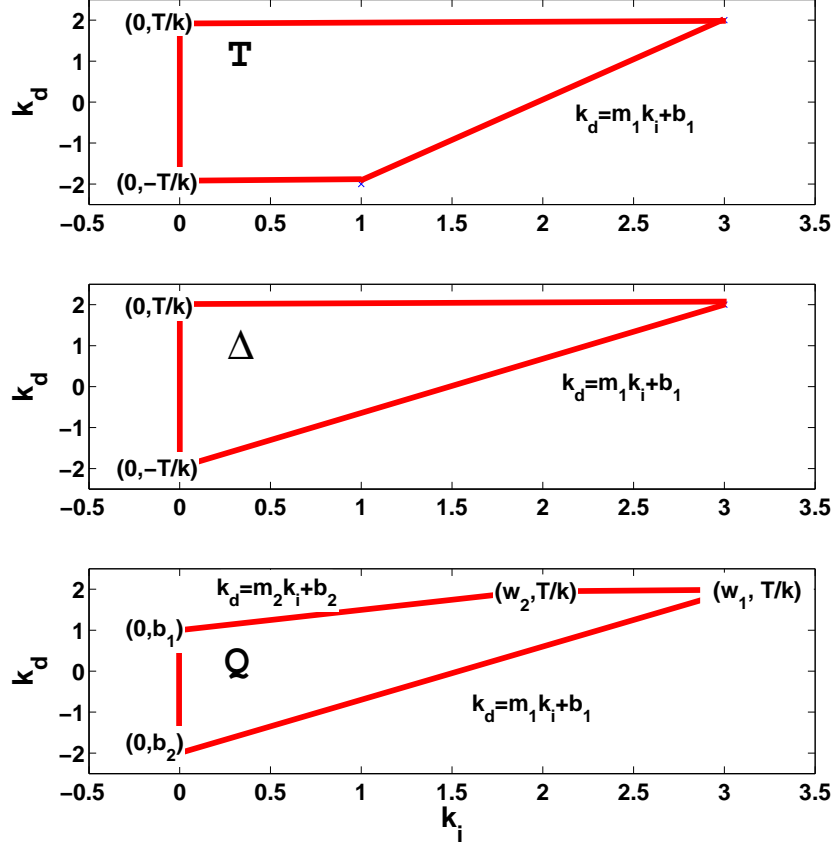


Figure 6: PID stable regions.

Where:

$$k_p = a \cdot k_u \quad (26)$$

$$k_d = c \cdot \frac{T}{K} \quad (27)$$

$$k_i = \begin{cases} b \cdot \frac{k_d - b_1}{m_1} & \text{if } k_p \leq 1/k \\ \frac{k_d - b_2}{m_2} + b \cdot \left[\frac{k_d - b_1}{m_1} - \frac{k_d - b_2}{m_2} \right] & \text{if } 1/k < k_p \end{cases} \quad (28)$$

As $c = 0$ (PI controllers), this leads to:

$$k_p = a \cdot k_u \quad (29)$$

$$k_i = \begin{cases} b \cdot \frac{-b_1}{m_1} & \text{if } k_p \leq 1/k \\ \frac{-b_2}{m_2} + b \cdot \left[\frac{-b_1}{m_1} - \frac{-b_2}{m_2} \right] & \text{if } 1/k < k_p \end{cases} \quad (30)$$

The algorithm to generate stabilising controllers is depicted in Algorithm 1. Simple analytic solutions using Newton-Raphson optimisation method could be employed. Line 1 and 2 could be calculated *off-line* if a fixed FOPDT is under consideration. Convergence properties of this coding have been evaluated for multiobjective optimisation in Reynoso-Meza et al. (2015).

- 1 Read values k, T, L ;
- 2 Calculate k_u (Equation (19));
- 3 Read values a, b, c ;
- 4 Calculate k_p (Equation (29));
- 5 Determine roots $z_j, j = [1, 2]$ (Equation (24));
- 6 Determine values $m_j, b_j, j = [1, 2]$ (Equations (21),(22));
- 7 Determine k_i (Equation (30));
- 8 Return k_p, k_i ;
- 9 Return $T_i = k_p/k_i$;

Algorithm 1: Stable PI stochastic sampling.

3.5. Overall procedure

The sp-MODE-II algorithm (Reynoso-Meza et al., 2014a) will be used because it is a MOEA already merged with the preference handling mechanism commented before⁴. It is a MOEA which incorporates a pruning mechanism to promote diversity in the Pareto front; for this purpose, it uses the physical programming index of Section 3.2 to evaluate the pertinency of a given solution, deciding if keeping or not in the Pareto front approximation. As evolutionary technique, it uses differential evolution algorithm (Storn and Price, 1997; Das and Suganthan, 2010). This algorithm, mixed with the MOP definition (Section 3.1), the automatic definition of preferences (Section 3.3) and the sampling mechanism (Section 3.4) build a general *MOP/EMO* approach with preferences for multivariable PI controller tuning.

4. Benchmark setup: the Wood & Berry distillation column process

The well-known distillation column model defined by Berry (1973); Wood and Berry (1973) will be used. In all instances, it is assumed that commonly used tuning techniques

⁴Frely available at: <http://www.mathworks.com/matlabcentral/fileexchange/47035>

don't fulfill all the designer's requirements and therefore, the MOOD procedure is employed. A standard CPU⁵ is used to calculate the Pareto front approximations.

The *ten times the number of objectives* thumb of rule for the quantity of solutions required in the approximated Pareto front commented in Reynoso-Meza et al. (2014a), based on Mattson and Messac (2005), is adopted. The aims of this example are:

- To provide a many-objective optimisation statement for MIMO processes, in order to validate the MOP/EMO proposal.
- To provide a comparison of the performance of different preference handling techniques to build a $\mathbf{T}\text{-}\mathbf{J}_P^*$, $\mathbf{D}\text{-}\mathbf{J}_P^*$ and $\mathbf{HD}\text{-}\mathbf{J}_P^*$ in many-objective optimisation instances.

Three different algorithms to approximate the $\mathbf{T}\text{-}\mathbf{J}_P^*$, $\mathbf{D}\text{-}\mathbf{J}_P^*$ and $\mathbf{HD}\text{-}\mathbf{J}_P^*$ are used:

Concept 1: sp-MODE-II algorithm of Reynoso-Meza et al. (2014a), with the *MOP/EMO* proposal defined in Section 3; relevant tuning parameters are shown in Table 2. Henceforth terms *MOP/EMO* approach and sp-MODE-II algorithm will be indistinctively used to make reference to the proposal of this work.

Concept 2: A DE algorithm with the diversity/pertinency mechanism described in Thiele et al. (2009), used for indicator based evolutionary algorithms (hereafter, IB-MODE). This technique is used because it is a state of the art technique capable of handling design preferences.

Concept 3: A stochastic sampling approach. This process is used for comparison purposes, since it has been noticed in Corne and Knowles (2007) that for many-objective optimisation instances, stochastic sampling procedures could be competitive with MOEAs.

For concepts 1 and 2, $Cr = 0.5$ and $F = 0.5$ are used in the DE mechanism (general values in accordance with Table 2). A budget of 2000 FEs is fixed in three instances. A total of 51 independent runs are carried out in each case. In all cases, the sampling mechanism and the MOP defined in Section 3 are used.

4.1. Design problem statement

As commented before, in order to show the applicability of the *MOP/EMO* approach for multivariable PI tuning, the well-known distillation column model defined by Wood and Berry will be used (Berry, 1973; Wood and Berry, 1973). It represents the control of the composition of two products (a mixture of methanol and water for this specific case), where the manipulated variables are the reflux and steam flows:

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \quad (31)$$

⁵DELL T1500 computer, Windows 7 system, processor Intel Core i7, 2.93 GHz with 8.00 GB RAM.

Table 2: Guidelines for spMODE-II's parameters tuning.

Parameter	Value
DE algorithm	
F (Scaling factor)	0.5
Cr (Crossover rate)	[0.5]
N_p (Population)	50
Pruning mechanism	
β_ϵ (Arcs)	70
Pertinency mechanism	
J_{gpp}^{max}	$J_{gpp}(\theta_0)$
$car(J_P^*)$	70

This process has been widely used to compare evolutionary algorithms (Iruthayarajan and Baskar, 2009, 2010; Coelho and Pessôa, 2011; Menhas et al., 2012) as well as MOEAs (Reynoso-Meza et al., 2012) performance. The process $P(s)$ (Equation (31)) will be controlled with a decentralized PI controller structure $C(s)$ (Equation (32)). The designer is interested in improving the performance of the initial solution $\theta_0 = [0.38, 8.29, -0.08, 23.6]$ (stated by the BLT methodology of Luyben, 1986).

$$C(s) = \begin{bmatrix} k_{p1} \left(1 + \frac{1}{T_{i1}s}\right) & 0 \\ 0 & k_{p2} \left(1 + \frac{1}{T_{i2}s}\right) \end{bmatrix} \quad (32)$$

The objective statement from Section 3.1 is used:

$$\min_{\theta} \mathbf{J}(\theta) = [-k_{i1}, -k_{i2}, L_{cm}, M_{s1}, M_{s2}, M_{p1}, M_{p2}] \in \mathbb{R}^7 \quad (33)$$

subject to:

$$\begin{aligned} 0.1 &\leq \theta_i \leq 1, i = [1, 3] \\ 0.0 &\leq \theta_i \leq 1, i = [2, 4] \\ 1.2 &\leq M_{s1, \dots, N} \leq 2.0 \\ 1 &\leq M_{p1, \dots, N} \leq 1.5 \\ 0 &\leq L_{cm} \leq 2N \\ G_1(\theta) &= Re[\lambda]_{max} < 0 \end{aligned} \quad (34)$$

where $k_{i1} = \frac{k_{p1}}{T_{i1}}$, $k_{i2} = \frac{k_{p2}}{T_{i2}}$ are the integral gains of controller $C_{11}(s)$ and $C_{22}(s)$ respectively; L_{cm} (Equation (12)) is the closed loop log modulus; M_{s1} and M_{s2} are the maximum

Table 3: Preferences Set for experimental setup of Section 4. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Set											
Objective	\leftarrow J_i^0	HD	$\rightarrow \leftarrow$ J_i^1	D	$\rightarrow \leftarrow$ J_i^2	T	$\rightarrow \leftarrow$ J_i^3	U	$\rightarrow \leftarrow$ J_i^4	HU	\rightarrow J_i^5
$J_1(\boldsymbol{\theta})$	$3.0 \cdot J_1(\boldsymbol{\theta}_0)$		$2.0 \cdot J_1(\boldsymbol{\theta}_0)$		$1.5 \cdot J_1(\boldsymbol{\theta}_0)$		$J_1(\boldsymbol{\theta}_0)$		$0.5 \cdot J_1(\boldsymbol{\theta}_0)$		$0.2 \cdot J_1(\boldsymbol{\theta}_0)$
$J_2(\boldsymbol{\theta})$	$6.0 \cdot J_2(\boldsymbol{\theta}_0)$		$4.0 \cdot J_2(\boldsymbol{\theta}_0)$		$3.0 \cdot J_2(\boldsymbol{\theta}_0)$		$J_2(\boldsymbol{\theta}_0)$		$0.8 \cdot J_2(\boldsymbol{\theta}_0)$		$0.6 \cdot J_2(\boldsymbol{\theta}_0)$
$J_3(\boldsymbol{\theta})$	1.0		2.0		3.0		4.0		5.0		10.0
$J_4(\boldsymbol{\theta})$	1.2		1.6		1.8		2.0		3.0		5.0
$J_5(\boldsymbol{\theta})$	1.2		1.6		1.8		2.0		3.0		5.0
$J_6(\boldsymbol{\theta})$	1.0		1.2		1.4		1.5		3.0		5.0
$J_7(\boldsymbol{\theta})$	1.0		1.2		1.4		1.5		3.0		5.0
$G_1(\boldsymbol{\theta})$	<0.0		<0.0		<0.0		0.0		1.0		2.0

values of the sensitivity function (Equation (9)) for individual control loops 1 and 2 respectively; M_{p1} and M_{p2} are the maximum values of the complementary sensitivity function (Equation (10)) for individual control loops 1 and 2, respectively. Variables θ_1, θ_3 are the coding values which correspond to proportional gains in the sampling procedure; Variables θ_2, θ_4 correspond to integral time values⁶. It is important to remember that it is expected to achieve a solution with better $J_1(\boldsymbol{\theta}_0)$, $J_2(\boldsymbol{\theta}_0)$ than the BLT ($\boldsymbol{\theta}_0$) solution. Also, limits of $(M_{s1}, M_{s2}) < 2.0$, $(M_{p1}, M_{p2}) < 1.5$ and $L_{cm} < 4.0$ are imposed.

For the sp-MODE-II (Concept 1) the preference matrix is depicted in Table 3. With this preference matrix, designer is stating that it is more interested in achieve a better response for second control loop. For the IB-MODE technique (Concept 2), two different approaches are evaluated: using as preference vector $J_{pref} = \mathbf{J}^T$ with $\rho = 0.001$ (named IB-MODE₁) and using $J_{pref} = \mathbf{J}^D$ with $\rho = 0.001$ (named IB-MODE₂). Since seven objectives are defined, 70 solutions are required by the DM in the approximated Pareto front. Therefore sp-MODE-II algorithm uses a spherical grid of 70 arcs and a population of 70 individuals; IBEA employs a population of 70 individuals.

4.2. Results and Discussions

In Table 4 the achieved **T_HypV** and **D_HypV** are depicted (in all cases, **HD_HypV** is null and thus omitted). Statistical significance was validated with the Wilcoxon test at 95% with Bonferroni correction (Derrac et al., 2011). In Figure 7 a visual representation of such statistics is shown. MOEAs were capable of reaching better **T_HypV** values than the stochastic approach. This validates the usage of MOEAs to approximate the **T_J_p*** in the many-objective optimisation instance of Equation (33).

Regarding MOEAs, on the one hand when comparing the sp-MODE-II and IB-MODE₁

⁶Notice that the constraints used to improve stability in the original statement (Reynoso-Meza et al., 2012) are no longer needed

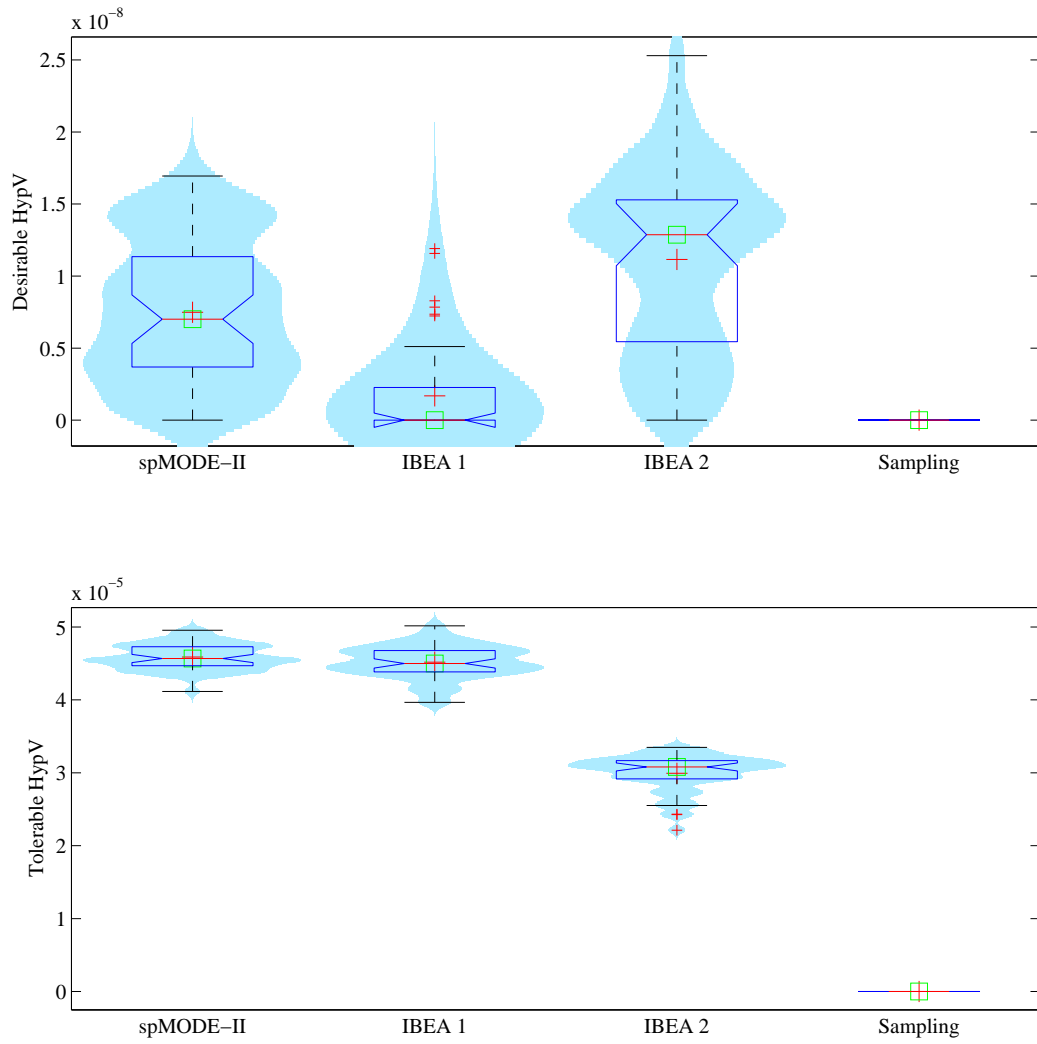


Figure 7: Pertinency level reached for benchmark setup 4. **T_HypV** and **D_HypV** achieved in 51 independent runs for each design concept.

Table 4: Pertinency level reached for benchmark setup 4. **T_HypV**, **D_HypV** and **HD_HypV** achieved in 51 independent runs for each design concept.

		sp-MODE-II algorithm (0%, 98%, 100%)	IB-MODE ₁ algorithm T_Vector , $\rho = 0.001$ (0%, 40%, 100%)	IB-MODE ₂ algorithm D_Vector , $\rho = 0.001$ (0%, 96%, 100%)	Random Sampling (0%, 0%, 55%)
D	Best	1.7E-008	1.2E-008	2.5E-008	0.0E+000
	Median	7.0E-009	0.00+000	1.3E-008	0.0E+000
	Worst	0.0E+000	0.0E+000	0.0E+000	0.0E+000
	Mean	7.5E-009	1.7E-009	1.1E-008	0.0E+000
	std	4.9E-009	3.0E-009	6.3E-009	0.0E+000
T	Best	4.9E-005	5.0E-005	3.4E-005	1.2E-005
	Median	4.6E-005	4.5E-005	3.1E-005	4.8E-008
	Worst	4.2E-005	4.0E-005	2.2E-005	0.0E+000
	Mean	4.6E-005	4.5E-005	3.0E-005	1.4E-006
	std	1.8E-006	2.3E-006	2.5E-006	2.3E-006
Time Performance		1.73	2.21	2.22	1.01

approaches, Wilcoxon test reveals that there are significant differences in the **D_HypV** covering, but there are not in the **T_HypV** covering. On the other hand, when comparing the sp-MODE-II and IB-MODE₂ approaches, Wilcoxon test reveals that there are significant differences in the **T_HypV** covering, but there are not in the **D_HypV** covering. This means that the sp-MODE-II algorithm achieves a better trade-off in covering both pertinent regions of the Pareto front in a single run; this happens because it has information about tolerable and desirable regions in a single run. It is important to remark that the aim here is to cover the surroundings of the pertinent region in the objective space with the information provided by the designer's preference statement. This is important because even if the designer has an idea about the expected trade-off, he/she doesn't know *a priori* the geometry of the Pareto front and thus, the exchange between conflictive objectives. Thus, it is important to guarantee the covering of this pertinent region in order to provide useful solutions for the MCDM stage.

Regarding the computational cost, a time base of 1,180 seconds is used (median value of the time required to evaluate 2000 FEs, with the available CPU resources, in 51 runs). Afterwards, the time required to execute each algorithm with the same number of FEs for each run is calculated and divided by the time base. This lead to the median values of time ratio depicted at the end of the Table 4. The dominance filter is the algorithm with the lower time performance (less than 1% of additional computational cost), while for the IB-MODE algorithm has the highest (more than 100% of additional computational cost).

In Figure 8 a LD / $\left\{ \widetilde{J}_{p1}^{*50\%}, \widetilde{J}_{p2}^{*50\%}, \widetilde{J}_{p3}^{*50\%} \right\} / Q(J^i(\theta^i), J_{p0}^*)$ visualisation of attainment surfaces of IB-MODE₁ (J_{p1} , green +), IB-MODE₂ (J_{p2} , blue x) and sp-MODE-II (J_{p3} , red \diamond) is depicted. A Pareto front approximation with all the solutions from all executions of each design concept evaluated has been used as reference front. The set of solutions close to

1 indicate⁷ that the algorithm approximates better the $\mathbf{T}\text{-}\mathbf{J}_P^*$ and $\mathbf{D}\text{-}\mathbf{J}_P^*$ Pareto fronts.

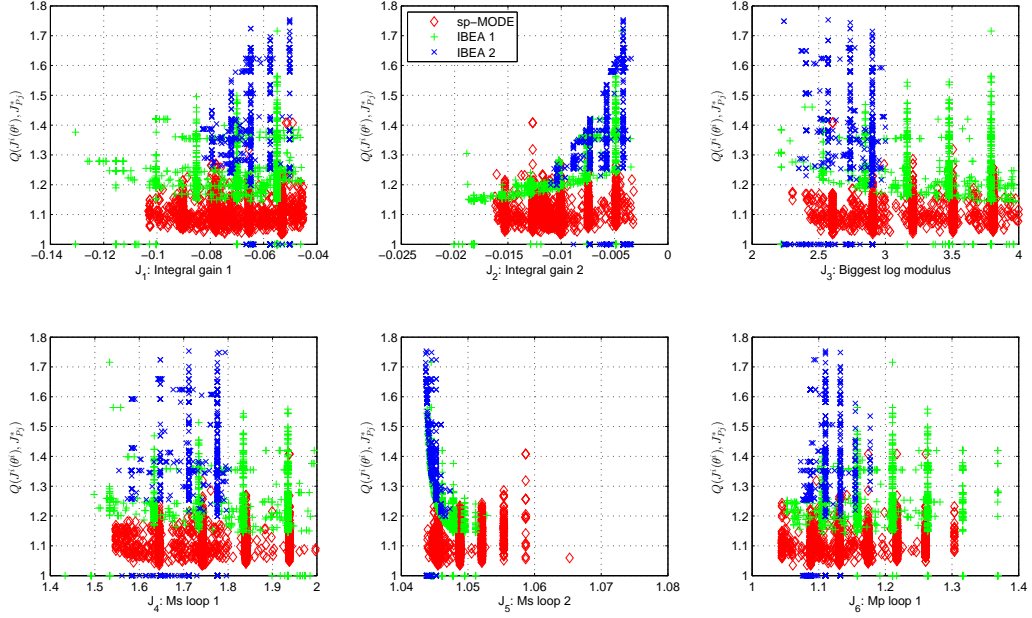


Figure 8: $\text{LD} / \left\{ \widetilde{\mathbf{J}}_{p1}^* 50\%, \widetilde{\mathbf{J}}_{p2}^* 50\%, \widetilde{\mathbf{J}}_{p3}^* 50\% \right\} / Q(\mathbf{J}^i(\theta^i), \mathbf{J}_{p0}^*)$ visualisation of attainment surfaces of IB-MODE₁ (J_{p1} , green +), IB-MODE₂ (J_{p2} , blue x) and sp-MODE-II (J_{p3} , red \diamond) for the benchmark setup of Section 4. Design objective 7 was intentionally omitted, since all algorithms achieved values close to 1.

4.3. Control test and validation

In order to validate the proposal, in Figure 9, the Pareto front approximation for the median value $\mathbf{T}\text{-}\mathbf{HypV}$ of the $\mathbf{T}\text{-}\mathbf{J}_P^*$ for sp-MODE-II algorithm is presented. From here, the DM will evaluate and analyse the approximated Pareto front to select a solution according to his/her needs. For this case, lets say that controller θ_{DM} (depicted with a star in Figure 9) satisfies the required performance. Main differences between this solution and the minimum GPP are the improvement on both integral gains with an affordable exchange on the robustness measures.

In Figure 10 ,this controller is compared with other multi-variable PI-tuning procedures for the Wood and Berry benchmark. Test under considerations are a set point response in the variable Y1 at 0 min and a set point response in variable Y2 at 100 min. In Table 5 some additional indicators are provided, as well as the parameters values of the corresponding controllers. As it can be notice, the controller is competitive with other tuning procedures, with

⁷According to que comparison methods developed for this visualisation in Reynoso-Meza et al. (2013a).

a smoother set-point response and control action, which might be a desirable characteristic for most industrial processes.

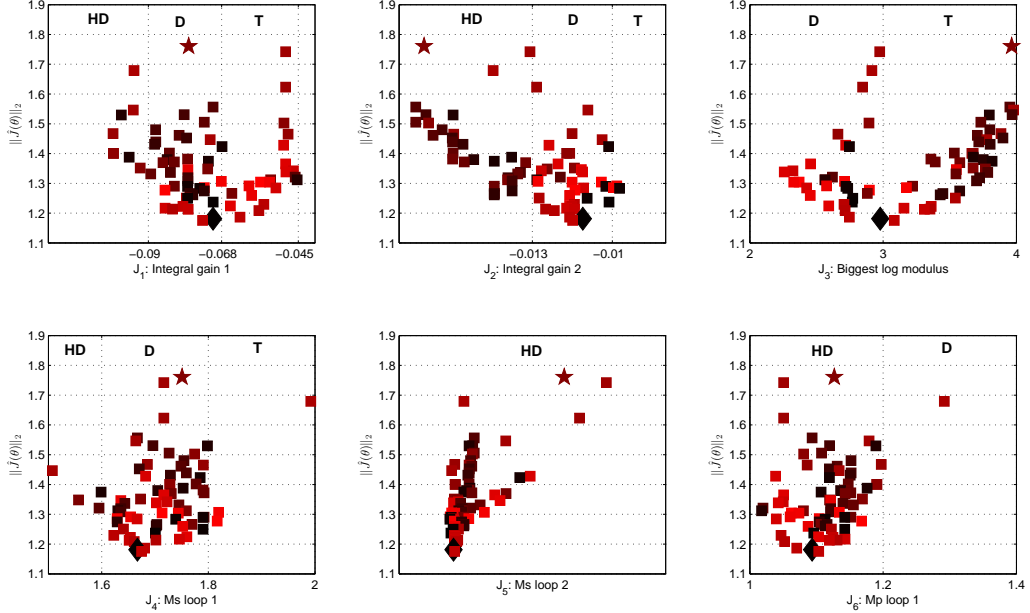
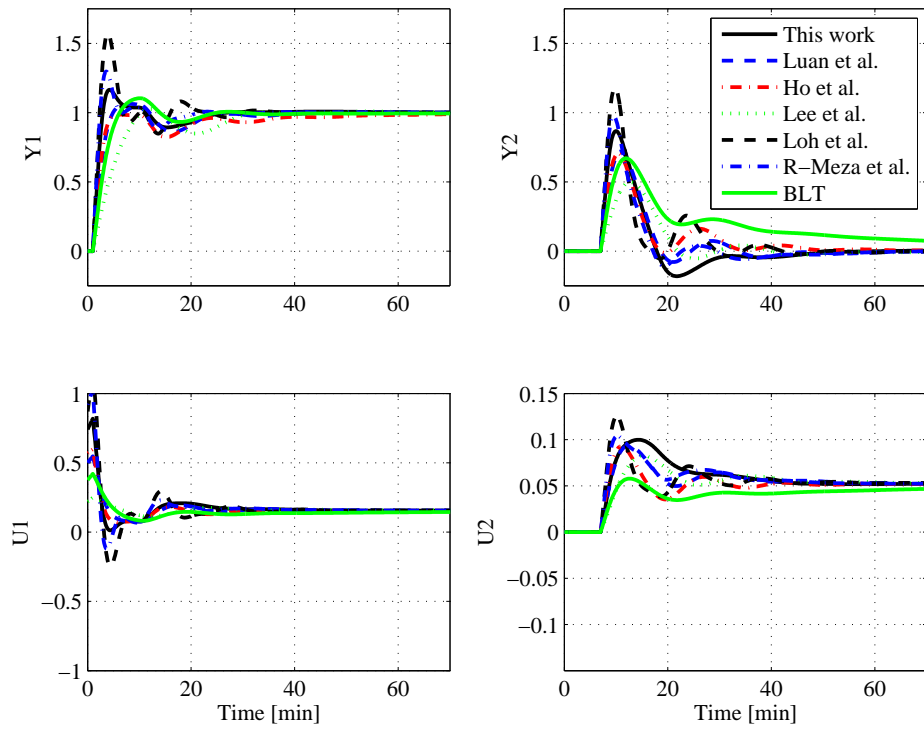


Figure 9: Pareto front approximation with sp-MODE-II algorithm of the benchmark setup of Section 4. The darker the solutions, the better the GPP index. Minimum GPP solution (diamond) and selected solution (star) are depicted. Design objective 7 was intentionally omitted, since all algorithms achieved values close to 1.

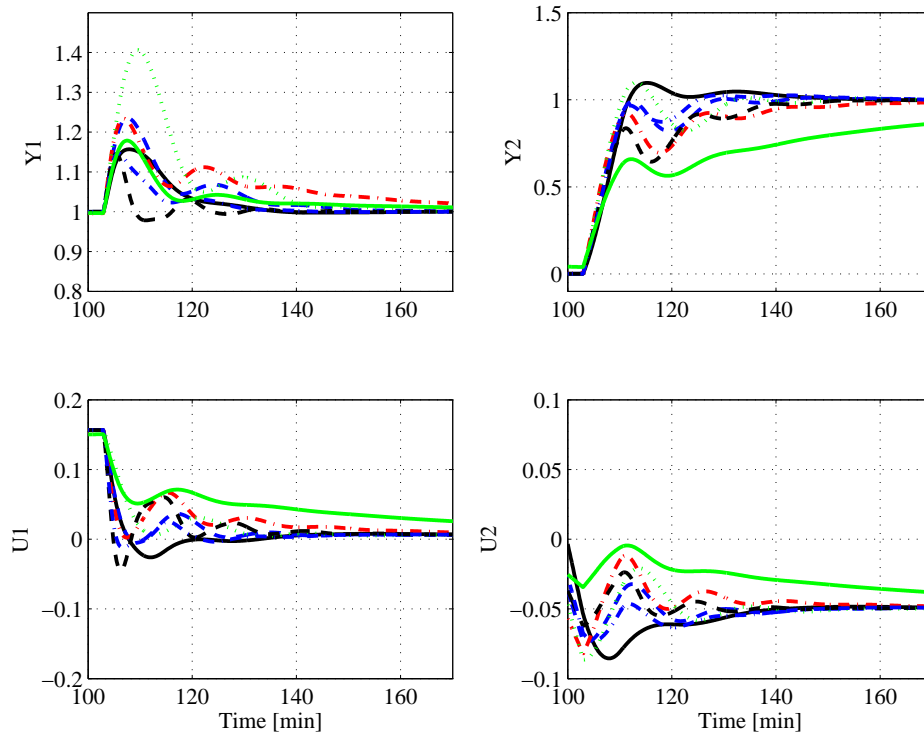
5. Conclusion

Intelligent control techniques have shown to be valuable tools for control engineers; particularly evolutionary multi-objective optimisation is a useful tool for controller tuning tasks. It allows the control engineer to perform a multi-criteria analysis between controller alternatives, in order to compare their trade-off. In this paper, a MOP definition and an EMO process have been proposed, in order to handle designer's preferences in the PI controller tuning problem for multi-variable processes. Importance of such proposal relies in the inclusion of the control engineer preferences into the optimisation process. In this way, the optimisation algorithm will seek actively for the pertinent region in the objective space, guaranteeing a useful set of solutions (reflecting designer's preferences) and thus facilitating the decision making process.

According to the example provided, such proposal is useful in multi-variable processes, common in industry, where individual loops and the overall system must fulfil a list of performance requirements. The methodology presented is competitive when compared with



(a) Setpoint change in variable $Y1$



(b) Setpoint change in variable $Y2$

Figure 10: Performance comparison of the selected controller with other tuning techniques for a given setpoint change for benchmark setup of Section 4.

Table 5: Performance comparison with other controllers. Integral fo the aboslute value of the error (IAE) and total variation (TV) of contral action are shown.

Reference	Parameters				Test 1				Test 2			
	kp_1	Ti_1	kp_2	Ti_2	IAE Y1	IAE Y2	TV U1	TV U2	IAE Y1	IAE Y2	TV U1	TV U2
This work	0.74	9.50	-0.06	3.33	3.62	7.49	1.14	0.21	2.03	9.03	0.23	0.12
Vu and Lee (2010)	0.50	10.54	-0.09	7.32	4.14	6.11	0.72	0.26	3.18	8.78	0.25	0.12
Ho et al. (1997)	0.57	20.70	-0.11	12.88	5.62	6.20	0.77	0.31	5.21	11.93	0.30	0.16
Lee et al. (2004)	0.22	8.35	-0.10	7.45	6.11	5.48	0.29	0.24	5.26	8.76	0.22	0.16
Loh et al. (1993)	0.87	3.25	-0.09	10.40	4.36	7.09	2.64	0.37	0.84	11.93	0.41	0.12
Reynoso-Meza et al. (2012)	0.92	8.74	-0.08	5.81	3.22	6.43	1.81	0.26	1.46	8.84	0.25	0.10
BLT tuning	0.38	8.29	-0.08	23.6	4.44	14.65	0.49	0.17	2.90	26.57	0.18	0.08

other multi-variable tuning techniques. This proposal focus only in PI controller structure and multi-variable process modelled as first order plus time delay. This represents an obvious limitation, since on the one hand, it is possible to find more complex processes; on the other hand, more advanced control structures might be used. Nevertheless, as PI control is still the first option for process control in industry, this proposal might be useful for industrial practitioners.

Regarding preference handling techniques, future work points towards developing techniques for *progressive* preference statement; that is, stating preferences *on the fly* while the optimisation algorithm is running. For such purpose machine learning or bayesian inference might be a useful tool, in order to catch the designer preferences which will be dynamically adapted while the optimisation is running. With regard to the EMO process, additional mechanisms to deal with the many-objectives optimisation instance and with multi-variable process are needed. As commented previously, given the amount of work published, EMO has been shown to be useful for single input single output process; nevertheless, few works focus on such multi-variable process, very common in industrial applications. Therefore, more practical proposals in this area might have an impact for such applications, increasing the usability of computational intelligence tools and intelligent control in the industry.

Acknowledgment

This work was partially supported by the fellowship BJT-304804/2014-2 from the National Council of Scientific and Technologic Development of Brazil (CNPq) and by EVO-CONTROL project (ref. PROMETEO/2012/028, Generalitat Valenciana - Spain).

Albertos, P., 2007. Lights and shadows of the intelligent control. In: Control Conference (ECC), 2007 European. IEEE, pp. 4440–4441.

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

Åström, K., Hägglund, T., 2001. The future of PID control. Control Engineering Practice 9 (11), 1163 – 1175.

- Åström, K., Panagopoulos, H., Hägglund, T., 1998. Design of PI controllers based on non-convex optimization. *Automatica* 34 (5), 585 – 601.
- Åström, K. J., Hägglund, T., 2005. *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709.
- Berry, M. W., 1973. Terminal composition control of a binary distillation column. Master's thesis, Department of Chemical and Petroleum Engineering, University of Alberta., Edmonton, Alberta.
- Bonissone, P., Subbu, R., Lizzi, J., 2009. Multicriteria decision making (MCDM): a framework for research and applications. *IEEE Computational Intelligence Magazine* 4 (3), 48 –61.
- Coelho, L. d. S., Pessôa, M. W., 2011. A tuning strategy for multivariable pi and pid controllers using differential evolution combined with chaotic zaslavskii map. *Expert Systems with Applications* 38 (11), 13694–13701.
- Coello, C., 2000. Handling preferences in evolutionary multiobjective optimization: a survey. In: *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. Vol. 1. pp. 30 –37.
- Coello, C. A. C., Lamont, G. B., 2004. *Applications of Multi-Objective evolutionary algorithms*, advances in natural computation vol. 1 Edition. World scientific publishing.
- Corne, D. W., Knowles, J. D., 2007. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. GECCO '07. ACM, New York, NY, USA, pp. 773 – 780.
- Cvetkovic, D., Parmee, I., 2002. Preferences and their application in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 6 (1), 42 –57.
- Das, S., Suganthan, P. N., 2010. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation* PP (99), 1 –28.
- Derrac, J., García, S., Molina, D., Herrera, F., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1 (1), 3 – 18.
- Ge, M., Chiu, M.-S., Wang, Q.-G., 2002. Robust PID controller design via LMI approach. *Journal of process control* (12), 3 – 13.
- Herreros, A., Baeyens, E., Perán, J. R., 2002. Design of PID-type controllers using multiobjective genetic algorithms. *ISA Transactions* 41 (4), 457 – 472.

- Ho, W. K., Lee, T. H., Gan, O. P., 1997. Tuning of multiloop proportional-integral-derivative controllers based on gain and phase margin specifications. *Industrial & engineering chemistry research* 36 (6), 2231–2238.
- Hung, M.-H., Shu, L.-S., Ho, S.-J., Hwang, S.-F., Ho, S.-Y., 2008. A novel intelligent multi-objective simulated annealing algorithm for designing robust PID controllers. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 38 (2), 319–330.
- Iruthayarajan, M. W., Baskar, S., 2009. Evolutionary algorithms based design of multivariable PID controller. *Expert Systems with Applications* 36 (5), 9159–9167.
- Iruthayarajan, M. W., Baskar, S., 2010. Covariance matrix adaptation evolution strategy based design of centralized PID controller. *Expert Systems with Applications* 37 (8), 5775–5781.
- Ishibuchi, H., Tsukamoto, N., Nojima, Y., 2008. Evolutionary many-objective optimization: A short review. In: *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*. IEEE Congress on. pp. 2419–2426.
- Jiménez, T., Merayo, N., Andrés, A., Durán, R. J., Aguado, J. C., de Miguel, I., Fernández, P., Lorenzo, R. M., Abril, E. J., 2015. An auto-tuning pid control system based on genetic algorithms to provide delay guarantees in passive optical networks. *Expert Systems with Applications* 42 (23), 9211–9220.
- Johnson, M. A., Moradi, M. H., 2005. *PID control*. Springer.
- Lee, M., Lee, K., Kim, C., Lee, J., 2004. Analytical design of multiloop pid controllers for desired closed-loop responses. *AIChE journal* 50 (7), 1631–1635.
- Li, Y., Ang, K. H., Chong, G., 2006. *PID control system analysis and design*. IEEE Control Systems.
- Liao, S.-H., 2005. Expert system methodologies and applications: a decade review from 1995 to 2004. *Expert systems with applications* 28 (1), 93–103.
- Loh, A. P., Hang, C. C., Quek, C. K., Vasnani, V. U., 1993. Autotuning of multiloop proportional-integral controllers using relay feedback. *Industrial & engineering chemistry research* 32 (6), 1102–1107.
- Luyben, W. L., 1986. Simple method for tuning SISO controllers in multivariable systems. *Industrial and Engineering Chemistry Process Design* (25), 654–660.
- Mattson, C. A., Messac, A., 2005. Pareto frontier based concept selection under uncertainty, with visualization. *Optimization and Engineering* 6, 85–115.
- Menhas, M. I., Wang, L., Fei, M., Pan, H., 2012. Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design. *Expert Systems with Applications* 39 (4), 4390–4401.

- Messac, A., 1996. Physical programming-effective optimization for computational design. *AIAA journal* 34 (1), 149–158.
- Miettinen, K. M., 1998. Nonlinear multiobjective optimization. Kluwer Academic Publishers.
- Mishra, P., Kumar, V., Rana, K., 2015. A fractional order fuzzy pid controller for binary distillation column control. *Expert Systems with Applications* 42 (22), 8533–8549.
- Munro, M., Aouni, B., 2012. Group decision makers’ preferences modelling within the goal programming model: An overview and a typology. *Journal of Multi-Criteria Decision Analysis* 19 (3-4), 169–184.
- Panagopoulos, H., Åström, K., Hägglund, T., 2002. Design of PID controllers based on constrained optimisation. *Control Theory and Applications, IEE Proceedings - 149* (1), 32 – 40.
- Ponce, H., Ponce, P., Bastida, H., Molina, A., 2015. A novel robust liquid level controller for coupled-tanks systems using artificial hydrocarbon networks. *Expert Systems with Applications* 42 (22), 8858–8867.
- Reynoso-Meza, G., Blasco, X., Sanchis, J., Herrero, J. M., 2013a. Comparison of design concepts in multi-criteria decision-making using level diagrams. *Information Sciences* 221, 124 – 141.
- Reynoso-Meza, G., Blasco, X., Sanchis, J., Martínez, M., 2013b. Evolutionary algorithms for PID controller tuning: Current trends and perspectives (in spanish). *Revista Iberoamericana de Automática e Informática Industrial* 10 (3), 251 – 268.
- Reynoso-Meza, G., dos Santos Coelho, L., Freite, R. Z., 2015. Efficient sampling of pi controllers in evolutionary multiobjective optimization. In: *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*. ACM, pp. 1263–1270.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., García-Nieto, S., 2014a. Multiobjective evolutionary algorithms for multivariable PI controller tuning. *Applied Soft Computing* 24, 341 – 362.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J. M., 2012. Multiobjective evolutionary algorithms for multivariable PI controller tuning. *Expert Systems with Applications* 39, 7895 – 7907.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J. M., August 2014b. A stabilizing PID controller sampling procedure for stochastic optimizers. In: *Memories of the 19th World Congress IFAC 2014*.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Martínez, M., 2014c. Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice* 1, 58 – 73.

- Robinson, J. A., 1998. Engineering thinking and rhetoric. *Journal of Engineering Education* 87 (3), 227–229.
- Ruano, A. E., 2005. Intelligent control systems using computational intelligence techniques. Vol. 70. Iet.
- Ruano, A. E., 2007. Intelligent control-the road ahead. In: *Control Conference (ECC), 2007 European*. IEEE, pp. 4442–4443.
- Ruano, A. E., Ge, S. S., Guerra, T. M., Lewis, F. L., Principe, J. C., Colnarič, M., 2014. Computational intelligence in control. *Annual Reviews in Control* 38 (2), 233 – 242.
- Sabzi, H. Z., Humberson, D., Abudu, S., King, J. P., 2016. Optimization of adaptive fuzzy logic controller using novel combined evolutionary algorithms, and its application in diez lagos flood controlling system, southern new mexico. *Expert Systems with Applications* 43, 154–164.
- Sanchez, H. S., Vilanova, R., 2013. Multiobjective tuning of pi controller using the nmc method: Simplified problem definition and guidelines for decision making. In: *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*. IEEE, pp. 1–8.
- Silva, G., Datta, A., Bhattacharyya, S., 2002. New results on the synthesis of PID controllers. *Automatic Control, IEEE Transactions on* 47 (2), 241–252.
- Stewart, G., Samad, T., 2011. Cross-application perspectives: Application and market requirements. In: Samad, T., Annaswamy, A. (Eds.), *The Impact of Control Technology*. IEEE Control Systems Society, pp. 95 – 100.
- Storn, R., Price, K., 1997. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341 – 359.
- Tan, W., Liu, J., Fang, F., Chen, Y., 2004. Tuning of PID controllers for boiler-turbine units. *ISA Transactions* 43 (4), 571 – 583.
- Thiele, L., Miettinen, K., Korhonen, P. J., Molina, J., 2009. A preference-based evolutionary algorithms for multi-objective optimization. *Evolutionary Computation* (3), 411 – 436.
- Toscano, R., 2005. A simple robust PI/PID controller design via numerical optimization approach. *Journal of process control* (15), 81 – 88.
- Tzafestas, S. G., 2007. The global goal and abilities of intelligent control. In: *Control Conference (ECC), 2007 European*. IEEE, pp. 4444–4446.
- Vilanova, R., Alfaro, V. M., 2011. Robust PID control: an overview (in spanish). *Revista Iberoamericana de Automática e Informática Industrial* 8 (3), 141 – 158.

- Vu, T. N. L., Lee, M., 2010. Independent design of multi-loop pi/pid controllers for interacting multivariable processes. *Journal of Process control* 20 (8), 922–933.
- Wood, R. K., Berry, M. W., 1973. Terminal composition control of a binary distillation column. *Chemical Engineering Science* 28 (9), 1707 – 1717.
- Xue, Y., Li, D., Gao, F., 2010. Multi-objective optimization and selection for the PI control of ALSTOM gasifier problem. *Control Engineering Practice* 18 (1), 67 – 76.
- Zhao, S.-Z., Iruthayarajan, M. W., Baskar, S., Suganthan, P. N., 2011. Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization. *Information Sciences* 181 (16), 3323 – 3335.