Preference Driven Multi-Objective Optimisation Design Procedure for Industrial Controller Tuning

Gilberto Reynoso-Meza [b,*], Javier Sanchis [a], Xavier Blasco [a], Miguel Martínez [a]

[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), Pontifical Catholic University of Parana (PUCPR), 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. These procedures could be used by designers when (1) it is difficult to find a reasonable trade-off for a controller tuning fulfilling several requirements; and (2) if it is worthwhile to analyse design objectives exchange among design alternatives. Despite the usefulness of such methods for describing trade-offs among design alternatives (tuning proposals) with the so called Pareto front, for some control problems finding a pertinent set of solutions could be a challenge. That is, some control problems are complex in the sense of finding the required trade-off among design objectives. In order to improve the performance of MOOD procedures for such situations, preference handling mechanisms could be used to improve pertinency of solutions in the approximated Pareto front. In this paper an overall MOOD procedure focusing in controller tuning applications using designer's preferences is proposed. In order to validate such procedure, a benchmark control problem is used and reformulated into a multi-objective problem statement, where different preference handling mechanisms in the optimisation process are evaluated and compared. The obtained results validate the overall proposal as a potential tool for industrial controller tuning.

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

1. Introduction

Multi-objective Optimisation Design (MOOD) procedures using Evolutionary Multi-objective Optimisation (EMO) have shown to be a valuable tool for controller tuning applications [41]. They enable the designer or decision maker (DM) having a close embedment into the design process; since it is possible to take into account each design objective individually; they also enable comparing design alternatives (*i.e.* tuning proposals), in order to select a controller fulfilling the expected trade-off among conflicting objectives. This MOOD

[*] Corresponding author.
Email address: g.reynosomeza@pucpr.br (Gilberto Reynoso-Meza)

procedure comprises at least, three fundamental steps: the multi-objective problem (MOP) definition, the EMO process and the multicriteria decision making (MCDM) step.

Such procedures have been used with success when (1) it is difficult to find a reasonable trade-off for a controller tuning fulfilling several requirements; and (2) if it is worthwhile analysing design objectives exchange among design alternatives. Despite the usefulness of such methods for describing trade-offs among design alternatives by the so called Pareto front, for some control problems finding a pertinent set of solutions could be a challenge. In such instances, finding pertinent solutions could be difficult due to complexity of the process and/or complexity of the MOP statement.

The former case refers when the process complexity makes difficult finding desirable solutions, even for 2 or 3 design objectives. That is, the region in the decision (search) space which fulfils designer's preferences could be difficult to find. In the latter case, designer commonly face the problem of fulfilling several performance objectives and requirements. If the number of design objectives is more than 3, it is said that the designer is dealing with a many-objective optimisation instance. This could increase the complexity of the EMO process and the MCDM step, since diversity and convergence properties of a given algorithm usually conflict each other in the Pareto front approximation.

An alternative to overcome the above mentioned issues, is the inclusion of preferences in the EMO process. The inclusion of preferences is exploited by algorithms in order to provide an interesting (useful) Pareto front approximation for designers [7]. This information could be used in the same way to deal effectively with many-objective optimisation instances [21], since the algorithm could be able to focus in the interesting regions of the objective space. Furthermore, preferences could be used to bridge any gap between problem definition, optimisation and decision making process [41, 34] leading to a holistic design procedure. With this tool, designer could address more effectively with complex processes, in the sense of complex to find a desirable trade-off among conflicting objectives.

The aim of this paper is twofold. On the one hand, proposing a MOOD procedure taking into account preference handling for controller tuning, in order to improve pertinency of solutions when it is difficult to find a desirable (required) trade-off. On the other hand, through the example provided, stating a controller tuning benchmark in the multi-objective optimisation context. The lack of formal benchmarks definitions for multi-objective optimisation in the control context was noticed in [41]; the statement of such benchmark will enable the comparison among techniques, methodologies and algorithms in the MOOD procedure context. This situation can motivate further developments of the MOOD procedure in controller tuning applications.

The remainder of this paper is as follows: in Section 2 a brief background on MOOD procedures and preference handling is commented; in Section 3 the proposal of this paper is presented; Section 4 is devoted to validate the MOOD procedure with preferences for controller tuning; with this aim, two different instances are stated using the Boiler Control benchmark problem of [28]: a univariable and a multivariable statement. Finally, some concluding remarks are given.

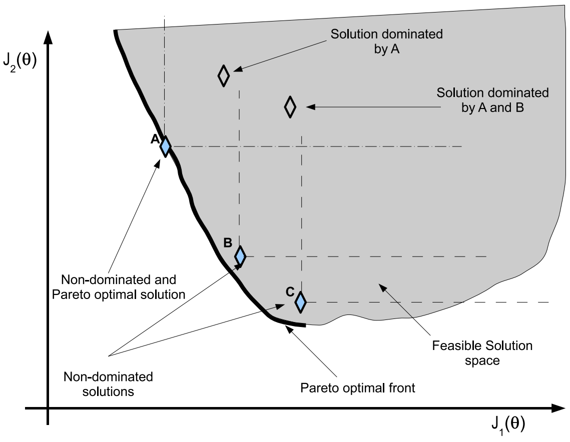


Figure 1: Pareto optimality and dominance concepts.

2. Theorical Background on Multi-objective Optimisation: a controller tuning context

Some notions on multi-objective optimisation and preference handling techniques are required. They are provided below, within the controller tuning framework.

2.1. Multi-objective optimisation statement

As referred in [26], a MOP [1], can be stated as follows:

$$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x}) = [J_1(\boldsymbol{x}), \ldots, J_m(\boldsymbol{x})] \quad (1)$$

subject to:

$$\boldsymbol{K}(\boldsymbol{x}) \leq 0 \quad (2)$$

$$\boldsymbol{L}(\boldsymbol{x}) = 0 \quad (3)$$

$$\underline{x_i} \leq x_i \leq \overline{x_i}, i = [1, \ldots, n] \quad (4)$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ is defined as the decision vector; $\boldsymbol{J}(\boldsymbol{x})$ as the objective vector and $\boldsymbol{K}(\boldsymbol{x})$, $\boldsymbol{L}(\boldsymbol{x})$ as the inequality and equality constraint vectors respectively; $\underline{x_i}, \overline{x_i}$ are the lower and upper bounds in the decision space.

It has been pointed out 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 $\boldsymbol{\Theta}_P$, is defined. Each solution in the Pareto set defines an objective vector in the Pareto front

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

$\boldsymbol{J}_P$. All the solutions in the Pareto front conforms a set of Pareto optimal and non-dominated solutions:

Definition 1. *(Pareto optimality [26]): An objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is Pareto optimal if there is not another objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ such that $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$.*

Definition 2. *(Dominance [8]): An objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is dominated by another objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ iff $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$.*

It is important to notice that the Pareto front is usually unknown, and the DM can only rely on a Pareto front approximation $\boldsymbol{J}_P^*$. In order to successfully embed the multi-objective optimisation concept into a design process, three fundamental steps [39] are (at least) required: the MOP definition (measure); the multi-objective optimisation process (search); and the multi-criteria decision making (MCDM) step (decision making). This procedure will be named Multi-objective Optimisation Design (MOOD) procedure (Figure 2). This procedures have been used for controller tuning applications with success, as commented below.

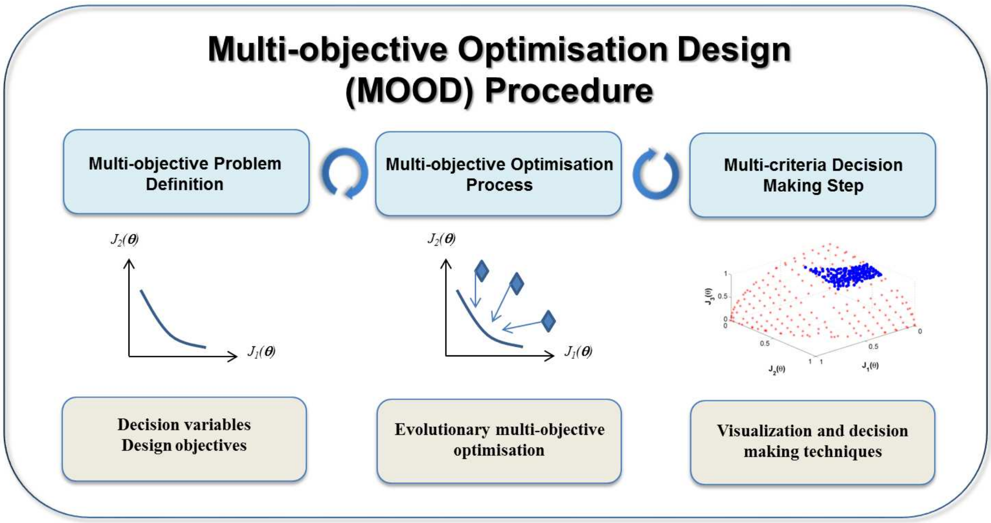

Multi-objective Optimisation Design (MOOD) Procedure
Multi-objective Problem Definition
Multi-objective Optimisation Process
Multi-criteria Decision Making Step
$J_2(\boldsymbol{\theta})$
$J_1(\boldsymbol{\theta})$
$J_2(\boldsymbol{\theta})$
$J_1(\boldsymbol{\theta})$
Decision variables
Design objectives
Evolutionary multi-objective optimisation
Visualization and decision making techniques

Figure 2: Multi-objective Optimisation Design (MOOD) procedure.

2.2. Background on controller tuning

When addressing a controller tuning problem by means of the MOOD procedure, the two following questions are important:

- *Is it difficult finding a controller with a reasonable balance among design objectives?*

- *Is it worthwhile analysing the trade-off among controllers (design alternatives)?*

The MOOD procedure is used in controller tuning applications not because controllers are difficult to optimise nor to find, but it might be a complex task to find a reasonable trade-off. Although there a lot of well established tuning techniques, the MOOD procedure is an alternative which focus on providing a reasonable trade-off solution in exchange of expending (investing) more time in the EMO process and the MCDM step. A basic control loop is depicted in Figure 3. 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 process output $Y(s)$ in the desired reference $R(s)$. The control problem consists in selecting proper tuning parameters for controller $C(s)$ in order to achieve a desirable performance of the process output as well as robust stability margins. This control problem is well known and it has been addressed with several techniques.

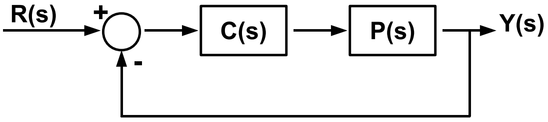


Figure 3: Basic control loop.

Classic techniques [26] to calculate this set of solutions have been used (such as varying weighting vectors, ϵ-constraint, and goal programming methods) as well as specialized algorithms (normal boundary intersection method [12] and normal constraint method [25] for example). Nevertheless sometimes these problems could be complex, non-linear and highly constrained, situation which makes difficult to find a useful Pareto set approximation. According to this, another way to approximate the Pareto set is by means of Evolutionary Multi-objective Optimisation (EMO), which is useful due to the flexibility of Multi-objective Evolutionary Algorithms (MOEAs) in dealing with non-convex and highly constrained functions [9, 8]. Such algorithms have been successfully applied in several control engineering [17, 41] and engineering design areas [45]. For this reason, MOEAs will be used in this work and hereafter the optimisation process will be performed by means of EMO in the MOOD procedure.

2.3. Background on preference handling in multi-objective optimisation

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. Incorporating the DM's preferences into MOEAs has been suggested to improve the pertinency of solutions (see for example [7, 11]).

The designer's preferences could be defined in the MOOD procedure in an *a priori*, *progressive*, or *a posteriori* fashion [29].

- *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, he or she defines the preferences in order to select a preferable solution.

It is also possible to classify preference handling techniques into five classes [38] with respect to the question: *what is it important for the designer?*:

- *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.

Some popular techniques include ranking procedures [10, 49, 33], goal attainment, and fuzzy relations [7]. In any case, some desirable characteristics of the preference handling mechanism have been stated in [14]:

- Managing multiple preference conditions simultaneously.
- Approximating Pareto-optimal solutions for each preference condition.
- The handling mechanism should be indifferent to the shape of the Pareto front.
- It should be capable of handling many-objective optimisation instances.

In [38] the following feature was also included:

- It should enable the DM to decide how many solutions are required in the Pareto front approximation, which will be analysed in the MCDM step.

In the case of controller tuning, capabilities to assuring the *dominance is essential* feature will be highly appreciated. Several tuning techniques and procedures are available for control engineers. Therefore and as noticed before, 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 exist an initial solution available to work with. In this sense, *dominance is essential*, given that an initial solution is usually available, and it is required to improve (at least) its overall performance. It will be assumed that exists such reference case controller, which allows to have an idea about what it is important to optimise and which might be the desirable trade-off region for the designer. Therefore, this work will focus on *a priori* preferences definitions where *dominance is essential*.

2.4. Preference conditions handling

A preference handling mechanism should be useful through the entire MOOD procedure. That is, regarding the MOP definition, it should be easy to code such preferences; regarding the EMO, it should be helpful to find a balance between diversity and pertinency mechanisms; finally, in the MCDM step, it should help to provide a useful Pareto front approximation for the designers. According to this, three different *a priori* mechanisms will be analysed: reference points [14], indicator-based with preferences [52] and global physical programming [38].

In the event that for the designer it is only important to prioritize some of the objectives (*Objective against objective*), fitter mechanisms such as the one proposed in [49] would be used; or if the designer is willing to refine a given set of preferences *on the fly* (progressive scheme), proposals as [22],[5],[6],[43] will be more appropriate. In both cases, they are out of the scope of this work.

2.4.1. Reference points

Mechanisms like the one defined by [14] are useful to cover the mentioned characteristics. It defines a proximity index to a previously defined set of reference points $\mathcal{R}$.

Definition 3. *The weighted euclidean distance to a set of reference points $\mathcal{R}$ is defined as:*

$$d(\boldsymbol{x}, \mathcal{R}) = \min_{\boldsymbol{R}_j \in \mathcal{R}} \left(\sqrt{\sum_{i=1}^{m} w \left(\frac{J_i(\boldsymbol{x}) - R_{j_i}}{J_i^{max} - J_i^{min}} \right)} \right) \tag{5}$$

It has been used as auxiliary criteria within the NSGA-II algorithm, to decide which solutions will be pruned (discarded) and which ones will remain in the Pareto front approximation.

2.4.2. Binary epsilon indicator

To evaluate the relative performance between two Pareto fronts (or design concepts), the $I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2})$ binary indicator [53] is used. This indicator shows the factor $\alpha = I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2})$ by which an approximation set $\boldsymbol{J}^*_{p1}$ is worse than another set $\boldsymbol{J}^*_{p2}$ with respect to all the objectives. As detailed in [53] and is useful to determine if two Pareto fronts are incomparable, equal, or if one is better than the other (see Table 1).

Definition 4. *The binary ϵ-indicator $I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2})$ [53] for two Pareto front approximations $\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}$ is defined as:*

$$I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}) = \max_{\boldsymbol{J}^2(\boldsymbol{x}^2) \in \boldsymbol{J}^*_{p2}} \epsilon_{\boldsymbol{J}^2(\boldsymbol{x}^2)} \tag{6}$$

where

$$\epsilon_{\boldsymbol{J}^2(\boldsymbol{x}^2)} = \min_{\boldsymbol{J}^1(\boldsymbol{x}^1) \in \boldsymbol{J}^*_{p1}} \epsilon_{\boldsymbol{J}^1(\boldsymbol{x}^1), \boldsymbol{J}^2(\boldsymbol{x}^2)} \tag{7}$$

$$
\begin{aligned}
\epsilon_{\boldsymbol{J}^1(\boldsymbol{x^1}),\boldsymbol{J}^2(\boldsymbol{x^2})} &= \max_{1\leq l\leq m} \frac{J^1(\boldsymbol{x}^1)_l}{J^2(\boldsymbol{x}^2)_l}, \\
\forall \boldsymbol{J}^1(\boldsymbol{x}^1) \in \boldsymbol{J}^*_{p1} \quad &, \quad \boldsymbol{J}^2(\boldsymbol{x}^2) \in \boldsymbol{J}^*_{p2}
\end{aligned}
\tag{8}
$$

Table 1: Interpretations for the I_ϵ indicator.

$I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}) < 1$	→	Every $\boldsymbol{J^2}(\boldsymbol{x^2}) \in \boldsymbol{J}^*_{p2}$ is strictly dominated by at least one $\boldsymbol{J^1}(\boldsymbol{x^1}) \in \boldsymbol{J}^*_{p1}$.
$I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}) = 1 \wedge I_\epsilon(\boldsymbol{J}^*_{p2}, \boldsymbol{J}^*_{p1}) = 1$	→	$\boldsymbol{J}^*_{p1} = \boldsymbol{J}^*_{p2}$.
$I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}) > 1 \wedge I_\epsilon(\boldsymbol{J}^*_{p2}, \boldsymbol{J}^*_{p1}) > 1$	→	Neither $\boldsymbol{J}^*_{p1}$ weakly dominates $\boldsymbol{J}^*_{p2}$ nor $\boldsymbol{J}^*_{p2}$ weakly dominates $\boldsymbol{J}^*_{p1}$.

The IBEA algorithm is an indicator based MOEA [52], which uses the $I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2})$ indicator [53] to evolve the entire Pareto front approximation. In each iteration the Pareto front approximation which optimises the indicator is selected; that is, the population which describes a Pareto front approximation $\boldsymbol{J}^*_{p1}$ which is better to the previous $\boldsymbol{J}^*_{p2}$ according to this index, is selected. If a fixed reference point $\boldsymbol{R}$ is used instead of a previous approximation $\boldsymbol{J}^*_{p2}$, then the population will evolve towards the region which dominates such reference point.

2.4.3. Global physical programming index

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 [24] for multi-objective problems that includes the *a priori* 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 classes[2] with previously defined ranges[3] according to a matrix of preferences. This matrix 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 its range of preferences in a matrix of preferences $\mathfrak{P}$, a class function $\eta_q(\boldsymbol{J}(\boldsymbol{x}))|_{\mathfrak{P}}$, $q = [1, \ldots, m]$ is built to translate each $J_q(\boldsymbol{x})$ to a new range where all

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

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

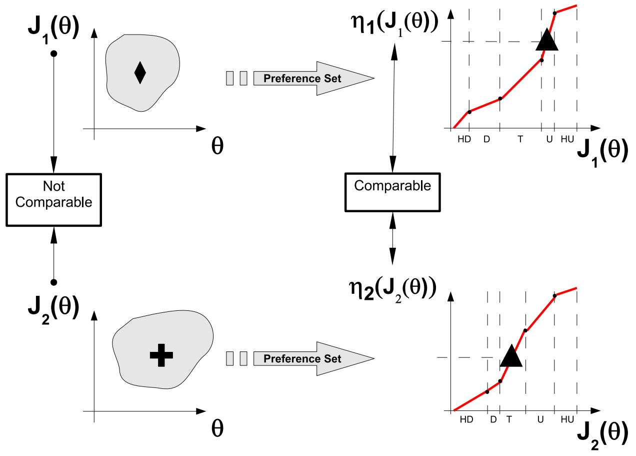


Figure 4: Physical programming (PP) notion. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

the objectives are equivalent to each other. A PP index $J_{pp}(\boldsymbol{J}(\boldsymbol{x})) = \sum\limits_{q=1}^{m} \eta_q(\boldsymbol{J}(\boldsymbol{x}))$ is then calculated.

In [38] the $J_{pp}(\boldsymbol{J}(\boldsymbol{x}))$ index is modified, and a global PP (GPP) index $J_{gpp}(\boldsymbol{\varphi})$ is defined for a given objective vector $\boldsymbol{\varphi}$. Main difference between both is that the latter uses linear functions to built the class functions, while the former uses splines with several requirements to maintain convexity and continuity; the former fits better for local optimisation algorithms, while the latter for (global) stochastic and evolutionary techniques. Furthermore, this GPP index enables encoding K preferences conditions, by defining a set of preferences $\mathfrak{K}$:

$$J_{gpp}(\boldsymbol{\varphi}) = \min_{\mathfrak{P}_{\mathfrak{k}} \in \mathfrak{K}, k=[1,\cdots,K]} \left(\sum_{q=1}^{m} \eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}_{\mathfrak{k}}} \right) \qquad (9)$$

Such index is helpful for pruning mechanism in MOEAs, in order to improve the pertinency of the solutions according to the predefined preferences. A typical preference matrix is shown in Table 2. Next, a preference handling mechanism will be selected, ir order to state the preference driven MOOD procedure of this work.

3. MOOD procedure proposal for controller tuning applications using preferences.

In this section, a preference driven MOOD procedure for controller tuning is proposed. Firstly, an analysis on the above commented preference handling mechanisms is presented, in order to select one of them for the overall procedure. Afterwards, in order to guarantee the successful implementation of the preference driven MOOD procedure, its three fundamental steps will be stated for controller tuning purposes: the MOP definition, the EMO process, and the MCDM stage.

Table 2: Preferences Set for the benchmark setup. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Matrix											
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{x})$ [-]	J_1^0		J_1^1		J_1^2		J_1^3		J_1^4		J_1^5
$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$
$J_m(\boldsymbol{x})$ [-]	J_m^0		J_m^1		J_m^2		J_m^3		J_m^4		J_m^5

3.1. Analysis on preference handling mechanisms

In Figure 5, a visual comparison of the landscapes generated by the preference handling techniques for a bi-objective problem are depicted. Five successive reference points have been used in each one of them.

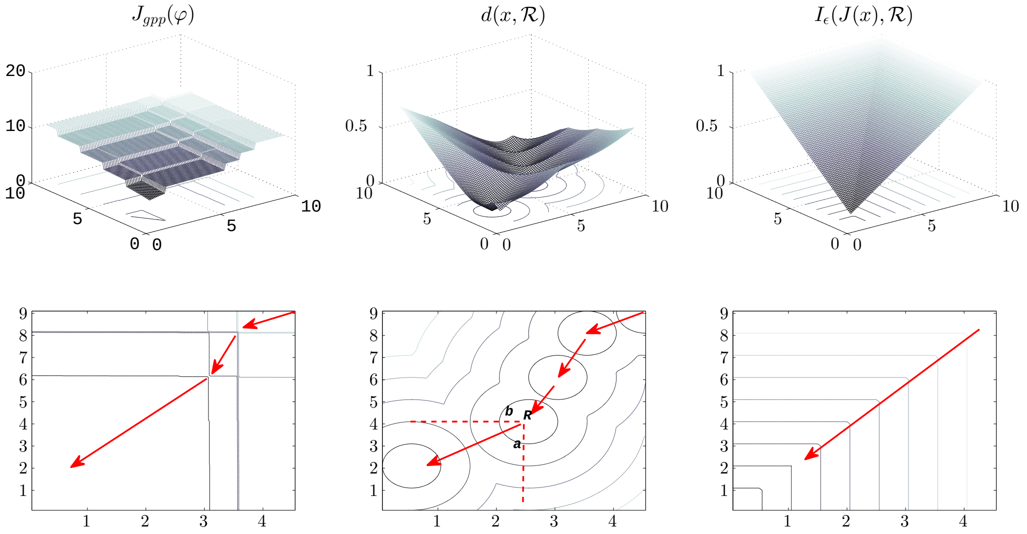


Figure 5: Comparison of path for successive aspiration levels (arrows) for $J_{gpp}(\boldsymbol{\varphi})$ index, $d(\boldsymbol{x}, \mathcal{R})$ and I_ϵ indicator.

On the one hand, the reference points can deal with different reference conditions and successive dominated aspirations levels. Nevertheless, a situation like the one depicted in Figure 5 can arise through the evolution process: a point b is preferred over point a since it is closer to reference point R. This situation does not preserve the *dominance is essential* feature. On the other hand, the I_ϵ indicator while it is effective dealing with multiple reference points, it doesn't handle efficiently the successive dominated aspiration levels. That is, the improvement path for successive aspiration levels is always the same.

According to the above commented, GPP index seems to be a practical option, since it can deals with multiple reference conditions and it provides a more flexible path for successive dominated aspiration levels. Furthermore, linguistic labels used in GPP are helpful for the designer, because they could provide not only a meaning for tolerability on design objectives, but also defines successive hypervolumes of desirability. Let's define the following vectors and values with these five preferences ranges of Table 2 (see Figure 6):

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

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

HD_Vector: $\boldsymbol{J}^{HD} = [J_1^1, J_2^1, \cdots, J_m^1]$, *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 $\boldsymbol{J}^T$.

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

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

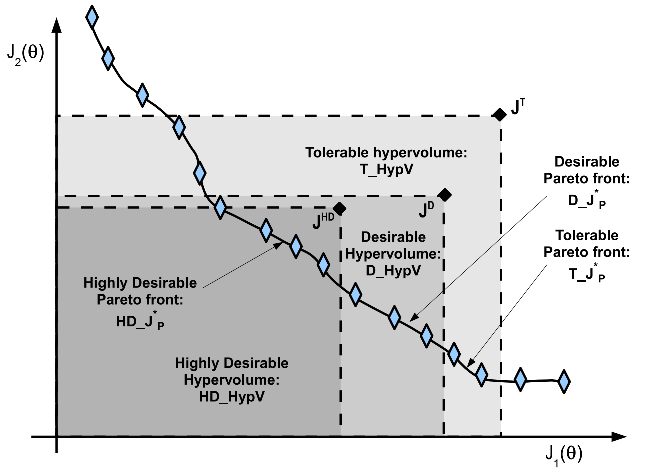

$J_2(\theta)$
J^T
Tolerable hypervolume: T_HypV
Desirable Pareto front: $D\_J^*_P$
J^{HD}
J^D
Desirable Hypervolume: D_HypV
Tolerable Pareto front: $T\_J^*_P$
Highly Desirable Pareto front: $HD\_J^*_P$
Highly Desirable Hypervolume: HD_HypV
$J_1(\theta)$

Figure 6: Graphical representation of the definitions stated.

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 it provides a path for the evolution process. Thus enabling a further improvement of the pertinency of the approximated Pareto front, according to the designer's preferences. In order to state the preferences ranges of Table 2, it is fundamental to have an understanding of the objectives to define the preference ranges. Nevertheless, if the DM has no idea on such values, it could be an indicative of a perfunctory or precipitate selection of the design objectives. Therefore, perhaps the DM should ponder the design objectives stated.

For above commented reasons, the GPP handling mechanism will be used as a pivotal tool in the overall procedure. Its flexibility and benefits compensate the fact of defining an overall preference matrix, as shown in [38].

3.2. The multiobjective problem definition

According to the basic control loop of Figure 3, some common choices in controller tuning [41] for design objectives are:

- Maximum value of sensitivity function

$$J_{M_s}(\boldsymbol{x}) = \left\| (I + P(s)C(s))^{-1} \right\|_\infty \tag{10}$$

- Maximum value of the complementary sensitivity function

$$J_{M_p}(\boldsymbol{x}) = \left\| P(s)C(s)(I + P(s)C(s))^{-1} \right\|_\infty \tag{11}$$

- Noise sensitivity

$$J_{M_u}(\boldsymbol{x}) = \left\| C(s)(I + P(s)C(s))^{-1} \right\|_\infty \tag{12}$$

- Integral of the absolute error value

$$J_{IAE}(\boldsymbol{x}) = \int\limits_{t=t_0}^{T_f} |r(t) - y(t)| \, dt \tag{13}$$

- Integral of the time weighted absolute error value

$$J_{ITAE}(\boldsymbol{x}) = \int\limits_{t=t_0}^{T_f} t \, |r(t) - y(t)| \, dt \tag{14}$$

- Total variation of control action

$$J_{TV}(\boldsymbol{x}) = \int\limits_{t=t_0}^{T_f} \left| \frac{du}{dt} \right| \tag{15}$$

where $r(t)$, $y(t)$, $u(t)$ are the reference, measured variable and control action in time t.

For frequency domain objectives, there are empirical relationships and limits, which are helpful to provide them with meaning. For example, it is known that practical limits for SISO processes exist ($1.2 \leq J_{M_s}(\boldsymbol{x}) \leq 2.0$ and $1.0 \leq J_{M_p}(\boldsymbol{x}) \leq 1.5$ [1, 31]). In the cases of time performance indexes, it is possible to provide meaning to them by using a reference case and normalizing those indicator [28]. Other indexes, as stabilizing time, overshoot, among others provide more meaning, but they are tied to the specific process at hand; *i.e* they strongly depend on the features of the process to be controlled. In any case, it has been noticed that using both kind of objectives could lead to more pertinent Pareto front approximations for the designer [19].

As it has been noticed before, MOOD procedure could be valuable for designers when it is difficult to find a desirable trade-off. In controller tuning, they are several and well established tuning techniques for different control loop strategies. Therefore, any advantage that the MOOD procedure could provide to controller tuning, is related to dominating a reference controller (or its surroundings), which is not fulfilling the desired specifications.

3.3. The EMO process

As explained before, the GPP index enables to state several preference conditions and it could handle successive dominated aspiration levels. Preference ranges of Figure 4 are defined for the sake of flexibility (as in [38]) to evolve the population to a pertinent Pareto front. According to this, the typical preference matrix of Table 2 is defined.

The GPP index can be merged with pruning techniques to search actively for the pertinent Pareto front approximation. Furthermore, it can be used to differentiate design objectives for the optimisation process from design objectives for the decision making. That is, perhaps the DM is interested in approximate a Pareto front for the most meaningful design objectives (objectives for decision making), but the designer would like to mind other design objectives (in the optimisation stage). For example, in Figure 7, the DM is interested to perform a decision with design objectives $J_1(\boldsymbol{x})$ and $J_2(\boldsymbol{x})$, but it is also interested in taking into account $J_3(\boldsymbol{x})$ in the optimisation. With an appropriate preference matrix, the GPP index will prefer the ■ solution over the • solution. Even if it appears to be a sub-optimal Pareto solution in the Pareto front approximation for $J_1(\boldsymbol{x})$ and $J_2(\boldsymbol{x})$, it is Pareto optimal for the $J_1(\boldsymbol{x})$, $J_2(\boldsymbol{x})$ and $J_3(\boldsymbol{x})$ objective space. If for example, a pruning mechanism which keeps one solution for each spherical sector is used, when performing the approximation with two design objectives the algorithm will keep only the • solution; if it is used for three design objectives, it will keep both solutions, but this could potentially increase (unnecessarily) the size of the Pareto front approximation. The GPP index as a pruning mechanism to keep one solution in each spherical sector is an alternative solution in-between, where a Pareto front is approximated in a reduced objective subspace, but taking into account all design objectives.

The spMODE-II algorithm [38][4] will be used because this is an implementation using the GPP index to improve the pertinency of solutions in the approximated Pareto front. It is a

[4] Available at `www.mathworks.com/matlabcentral/fileexchange/authors/289050`.

Differential Evolution (DE) based MOEA, which uses a spherical grid to maintain diversity in the approximated Pareto front. For each spherical sector, usually a norm is used in order to keep just one design alternative (original spMODE version [37]); nevertheless, this norm is substituted by the GPP index in the new version of the algorithm.

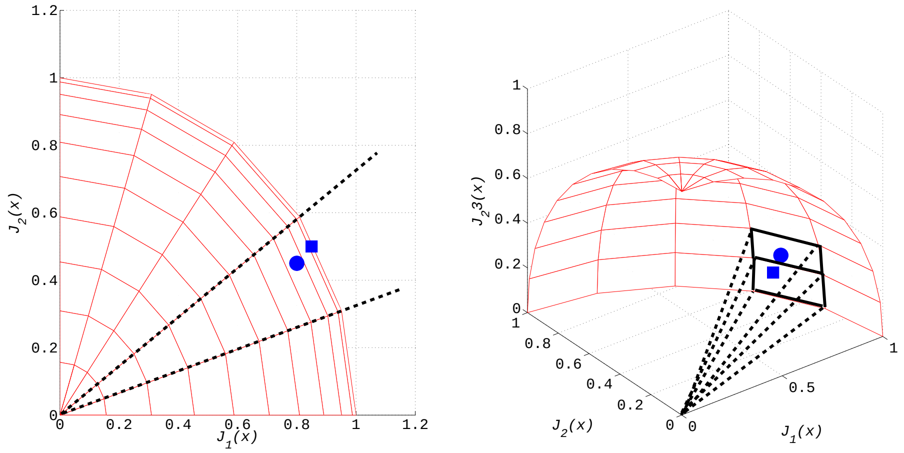


Figure 7: Difference between design objective for optimisation and for decision making. The GPP index as a pruning mechanism to keep one solution in each spherical sector is an intermediate solution, where a reduced Pareto front is approximated, both taking into account more design objectives.

3.4. MCDM stage

The *ten times the number of objectives* thumb of rule for the quantity of solutions required in the approximated Pareto front based on [24], is adopted. Also a clear distinction among design objectives and design objectives for decision making is stated, as commented earlier. That is, in which subspace the DM would like to perform a decision making analysis, by identifying objectives that should be minimised and minded in the search process, but they are not meant to be used for decision making.

Different alternatives could be used by practitioners [20, 4, 13, 48], nevertheless Level Diagrams (LD) will be used due to their capabilities to depict m-dimensional Pareto fronts [3] and for design concepts comparison [35]. The taxonomy to identify the visualisations is adopted from [35] [5].

4. Benchmark setup: the Boiler Control problem

The process under consideration is the benchmark for PID control 2012 described by [28]. It proposes a boiler control problem [27, 16] based on the work of [32]. This work improves

[5] *LD/front/measure*. For example, LD/$\boldsymbol{J}_p^*/\|\hat{\boldsymbol{J}}(\boldsymbol{x})\|_2$, means that a visual representation of Pareto front approximation $\boldsymbol{J}_p^*$ with 2-norm in LD is presented.

the model provided in [2] by adding a non-linear combustion equation with a first order lag to model the excess oxygen in the stack and the stoichiometric *air-to-fuel* ratio for complete combustion. The non-linear explicit model is described by the following equations:

$$
\begin{aligned}
\dot{\mathbf{x_1}}(t) &= c_{11}\mathbf{x_4}(t)\mathbf{x_1}^{\frac{9}{8}} + c_{12}\mathbf{u_1}(t-\tau_1) - c_{13}\mathbf{u_3}(t-\tau_3) && (16)\\
\dot{\mathbf{x_2}}(t) &= c_{21}\mathbf{x_2}(t) \\
&\quad + \frac{c_{22}\mathbf{u_2}(t-\tau_2) - c_{23}\mathbf{u_1}(t-\tau_1) - c_{24}\mathbf{u_1}(t-\tau_1)\mathbf{x_2}(t)}{c_{25}\mathbf{u_2}(t-\tau_2) - c_{26}\mathbf{u_1}(t-\tau_1)} && (17)\\
\dot{\mathbf{x_3}}(t) &= -c_{31}\mathbf{x_1}(t) - c_{32}\mathbf{x_4}(t)\mathbf{x_1}(t) + c_{33}\mathbf{u_3}(t-\tau_3) && (18)\\
\dot{\mathbf{x_4}}(t) &= -c_{41}\mathbf{x_4}(t) + c_{42}\mathbf{u_1}(t-\tau_1) + c_{43} + \mathbf{n_5}(t) && (19)\\
\mathbf{y_1}(t) &= c_{51}\mathbf{x_1}(t-\tau_4) + \mathbf{n_1}(t) && (20)\\
\mathbf{y_2}(t) &= c_{61}\mathbf{x_1}(t-\tau_5) + \mathbf{n_2}(t) && (21)\\
\mathbf{y_3}(t) &= c_{70}\mathbf{x_1}(t-\tau_6) + c_{71}\mathbf{x_3}(t-\tau_6) \\
&\quad + c_{72}\mathbf{x_4}(t-\tau_6)\mathbf{x_1}(t-\tau_6) + c_{73}\mathbf{u_3}(t-\tau_3-\tau_6) \\
&\quad + c_{74}\mathbf{u_1}(t-\tau_1-\tau_6) \\
&\quad + \frac{[c_{75}\mathbf{x_1}(t-\tau_6) + c_{76}]\,[1 - c_{77}\mathbf{x_3}(t-\tau_6)]}{\mathbf{x_3}(t-\tau_6)\,[\mathbf{x_1}(t-\tau_6) + c_{78}]} \\
&\quad + c_{79} + \mathbf{n_3}(t) && (22)\\
\mathbf{y_4}(t) &= [c_{81}\mathbf{x_4}(t-\tau_7) + c_{82}]\,\mathbf{x_1}(t-\tau_7) + \mathbf{n_4}(t) && (23)
\end{aligned}
$$

Where $\mathbf{x_1}(t)$, $\mathbf{x_2}(t)$, $\mathbf{x_3}(t)$, $\mathbf{x_4}(t)$ are the states of the system; $\mathbf{y_1}(t)$, $\mathbf{y_2}(t)$, $\mathbf{y_3}(t)$, $\mathbf{y_4}(t)$ the outputs; $\mathbf{u_1}(t)$, $\mathbf{u_2}(t)$, $\mathbf{u_3}(t)$ the inputs; c_{ij}, τ_i and $\mathbf{n_i}$ are non-linear coefficients, time constants and noise models, respectively, determined to improve the accuracy on the model.

It is assumed that commonly used tuning techniques don't fulfill all the designer's requirements and therefore, the MOOD procedure is employed. Furthermore, that it is complex to find a desirable trade-off and therefore, the MOOD proposal with preferences is used. Two different instances will be evaluated: the univariable and multivariable versions of the benchmark.

4.1. Univariable benchmark setup: multiple preference conditions

The reduced *single input, single output* (SISO) version of the benchmark stated in [28] is used in this example. This reduced version comprises the steam pressure control by means of manipulating fuel flow. The identified nominal model[6] $G(s)$ to be used in the optimisation process is:

$$G(s) = \frac{0.3934}{1 + 45.6794s} e^{-3.42s} \tag{24}$$

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

In order to control this process, a proportional-interal-derivative controller with derivative filter (PIDn) is proposed. Its structure is as follows:

$$C(s) = k_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{T_d/N \cdot s + 1}\right) \tag{25}$$

Where k_p is the proportional gain; T_i, T_d are the integral and derivative time values; and N the derivative filter. Aims of this example are:

- Defining a controller tuning MOP statement with multiple preference conditions.
- Evaluating and comparing the performance of the preference handling approach, in order to validate its selection.

4.1.1. Design problem statement

For this example the MOP definition is:

$$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x}) \quad = \quad [J_1(\boldsymbol{x}), \ldots, J_3(\boldsymbol{x})] \tag{26}$$

subject to:

$$0 \leq x_i \quad \leq \quad 1, i = [1, 2, 3, 4] \tag{27}$$

$$L_1(\boldsymbol{x}) = Re[\lambda]_{max} \quad < \quad 0 \tag{28}$$

Where $\boldsymbol{x}$ are the tuning parameters of the PIDn controller. In order to bound the controller parameters, the stochastic sampling described in [40] for stabilizing PID controllers is implemented, and therefore bound constraints are $x_i \in [0, 1]$ for $i = 1, \cdots, 3$; the derivative filter is bound in the interval $N = x_4 \in [3, 20]$ (practical values recognized for industrial applications). The design objectives stated are:

$J_1(\boldsymbol{x})$: IAE performance for a unitary step reference change (Equation (13)).

$J_2(\boldsymbol{x})$: Maximum value of sensitivity function M_s for control loop (Equation (10)).

$J_3(\boldsymbol{x})$: Maximum value of M_u for noise rejection in the control loop (Equation (12)).

Two preference matrix for the design objectives stated are defined (Table 3). Preference Set A promotes performance, while the preference set B promotes robustness. Design objectives to perform a MCDM stage are $J_1(\boldsymbol{x}), J_2(\boldsymbol{x})$. That means that 20 solutions are required and the objective space is just partitioned in two dimensions. It is important to notice that all three objectives are used in the optimisation process in order to calculate the GPP index, since they are included in the preference matrix.

Table 3: Preferences for the univariable benchmark setup. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

			Preference Set A			
	← HD	→← D	→← T	→← U	→← HU	→
Objective	J_i^0	J_i^1	J_i^2	J_i^3	J_i^4	J_i^5
$J_1(\boldsymbol{x})$ [-]	0.0	30.0	60.0	120.0	500.0	800.0
$J_2(\boldsymbol{x})$ [-]	1.0	1.2	1.5	2.0	3.0	4.0
$J_3(\boldsymbol{x})$ [-]	0.0	10.0	50.0	100.0	110.0	120.0
			Preference Set B			
	← HD	→← D	→← T	→← U	→← HU	→
Objective	J_i^0	J_i^1	J_i^2	J_i^3	J_i^4	J_i^5
$J_1(\boldsymbol{x})$ [-]	0.0	10.0	20.0	120.0	500.0	800.0
$J_2(\boldsymbol{x})$ [-]	1.0	1.6	1.8	2.0	3.0	4.0
$J_3(\boldsymbol{x})$ [-]	0.0	40.0	80.0	100.0	110.0	120.0

Table 4: Parameters used for IB-MODE, RP-spMODE and spMODE-II in univariable benchmark setup. Further details in [34]

spMODE-II	RP-spMODE	IB-MODE	**Parameter**	**Value**
			Evolutionary mechanism	
			F (Scaling factor)	0.5
			Cr (Crossover rate)	0.5
			N_p (Population)	20
			Pruning mechanism	
			$\boldsymbol{\beta_\epsilon}$ (Arcs)	100
			Pertinency mechanism	
			J_{gpp}^{max}	$J_{gpp}([120, 2.0, 100.0])$
			$car(J_P^*)$	30

4.1.2. Results and Discussions on the EMO process

In order to test the usefulness of the preference handling mechanism, two different preferences handling mechanisms will be also evaluated, using as main evolutionary mechanism the DE algorithm:

- An IBEA [47] using a basic DE algorithm, hereafter denoted as IB-MODE. This strategy is selected because it is a *state of the art* technique for handling simultaneous preference conditions. It uses the binary epsilon indicator previously explained in Section 2.4.2.
- A spMODE algorithm using the reference point based multi-objective optimisation technique described in [14] (and commented in Section 2.4.1), hereafter denoted as RP-spMODE. This technique is selected because it is consistent with the spMODE algorithm structure, and it could be used as a *customized* norm in its pruning mechanism.

In all cases, parameters used are depicted in Table 7 for each case. Additionally, the following *control experiment* is included:

- A pure stochastic sampling approach, using the same function evaluations budget. This is used as a base test to evaluate the usability of the approaches above.

The aim of this analysis is to evaluate the capabilities of such approaches to approximate a Pareto front in the **T_HypV**, **D_HypV** and **HD_HypV** in order to validate the usefulness of the spMODE-II[7] algorithm for these applications. A standard CPU[8] is used to calculate the Pareto front approximations for this benchmark.

In Figure 8, distribution plots of the attained **HD_HypV**, **D_HypV** are depicted; in Table 5 numerical values for the best, worst, median, mean and standard deviation of the attained hypervolumes are shown. Statistical significance has been validated using the Wilcoxon test at 95% with Bonferroni correction [15].

With the sampling approach it can be noticed that the **HD_HypV** for the preference Set B is most difficult to achieve. Also, according with the provided data, the GPP approach is able to attain the **HD_HypV** better both preference sets simultaneously. This, therefore, justify its usability for preference driven MOOD procedure for controller tuning.

4.2. Multivariable benchmark setup: a many-objectives optimisation statement

For the secod instance the reduced *two inputs, two outputs* (TITO) version of the benchmark stated in [28] is used:

[7] Hereafter, the GPP approach for EMO and spMODE-II will be used indistinctively

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

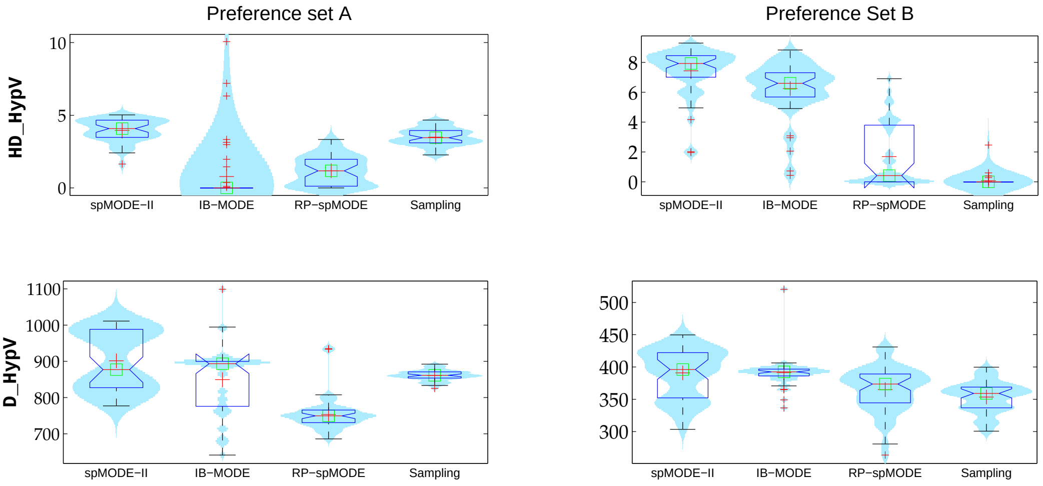


Figure 8: Distribution plots for the hypervolumen attained by the different approaches used in the univariable benchmark setup.

Table 5: Hypervolume achieved in the univariable benchmark setup (51 runs). Statistical significance has been evaluated according to the Wilcoxon and it is indicated for each case.

		Preference Set A				Preference Set B			
		spMODE-II✓	IB-MODE†	RP-spMODE‡	Sampling §	spMODE-II✓	IB-MODE†	RP-spMODE‡	Sampling §
HD_HypV	Best	5.0280	10.0660	3.3333	4.6682	9.2954	8.8309	6.9039	2.4625
	Median	**4.0845**†, ‡	0.0000✓, §	1.1781✓, §	3.4513†, ‡	**7.9294**‡, §	6.6029‡, §	0.4179✓, †	0.0000✓, †
	Worst	1.6416	0.0000	0.0000	2.2763	1.9594	0.4360	0.0000	0.0000
	Mean	3.9631	0.7838	1.1847	3.4987	7.4335	6.2478	1.6839	0.0744
	std	0.7621	2.0332	0.9856	0.6074	1.5662	1.7796	2.0977	0.3572
D_HypV	Best	1.01e+03	1.10e+03	0.94e+03	0.89e+03	0.45e+05	0.52e+03	0.43e+03	0.40e+03
	Median	0.87e+03§	**0.89e+03**§	0.75e+03§	0.86e+03✓, †, ‡	**0.40e+05**‡, §	0.39e+03‡, §	0.37e+03✓, †	0.36e+03✓, †
	Worst	0.77e+03	0.64e+03	0.69e+03	0.83e+03	0.30e+05	0.30e+03	0.26e+03	0.30e+03
	Mean	0.90e+03	0.85e+03	0.75e+03	0.86e+03	0.39e+05	0.39e+03	0.36e+03	0.35e+03
	std	0.08e+03	0.09e+03	0.05+e03	0.01e+03	0.04e+05	0.02e+03	0.03e+03	0.02e+03

$$
\begin{aligned}
\begin{bmatrix} Y_1(s) \\ Y_3(s) \end{bmatrix} &= \begin{bmatrix} P_{11}(s) & P_{13}(s) \\ P_{31}(s) & P_{33}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_3(s) \end{bmatrix} \\
&+ \begin{bmatrix} P_{1d}(s) \\ P_{3d}(s) \end{bmatrix} D(s) \qquad (29)
\end{aligned}
$$

Where the inputs are fuel flow $U_1(s)$ [%], air flow $U_2(s)$ [%] and water flow $U_3(s)$ [%], while the outputs are steam pressure $Y_1(s)$ [%], oxygen level $Y_2(s)$ [%] and water level $Y_3(s)$ [%]. $D(s)$ is a measured load disturbance. This is a verified model, useful to propose, evaluate and compare different kinds of tuning/control techniques ([18, 44, 30, 46, 42]).

For the sake of simplicity a proportional-integral (PI) controller for a multiple input, multiple output (MIMO) instance will be used. According to [41], while several works focus on PI-like 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 test the MOOD procedure with the proposals contained in this paper. In all instances, it is assumed that commonly used tuning techniques don't fulfill all the designer's requirements and therefore, the MOOD procedure is employed. EA's and MOEA's in PI controller tuning is still an ongoing research field [50, 51, 36]. The proposed multivariable PI controller structure is:

$$
C(s) = \begin{bmatrix} k_{p1}\left(1 + \frac{1}{T_{i1}s}\right) & 0 \\ 0 & k_{p2}\left(1 + \frac{1}{T_{i2}s}\right) \end{bmatrix} \qquad (30)
$$

Where k_{p1}, k_{p2} are the proportional gains, and T_{i1}, T_{i2} are the integral time values. Aims of this example are:

- Providing a many-objective optimisation statement for MIMO processes under *quasi-real* conditions.
- Validating the overall preference driven MOOD procedure for controller tuning under *quasi-real* conditions.

Quasi-real conditions makes reference to the following steps:

1. Consider the (original) non-linear model simulation as the *real process.*
2. Step tests are used to obtain simplified linear models from the *real process.*
3. Controllers are tuned using the aforementioned approximated models.
4. The selection procedure will be made according to experiments on the approximated models.
5. The selected controller will be implemented in the *real process.*

4.2.1. Design problem statement

The identified reduced model[9] at the operating point is shown in Equations (31), (32) and depicted in Figure (9).

$$\begin{aligned} P(s) &= \begin{bmatrix} P_{11}(s) & P_{13}(s) \\ P_{31}(s) & P_{33}(s) \end{bmatrix} \\ &= \begin{bmatrix} \frac{0.3727e^{-3.1308s}}{55.68s+1} & \frac{-0.1642}{179.66s+1} \\ \frac{0.0055\cdot(166.95s-1)}{31.029s^2+s} & \frac{0.0106e^{-9.28s}}{s} \end{bmatrix} \end{aligned} \tag{31}$$

$$D(s) = \begin{bmatrix} P_1(s) \\ P_3(s) \end{bmatrix} = \begin{bmatrix} \frac{-0.78266e^{-17.841s}}{234.69s+1} \\ \frac{-0.0014079e^{-7.1872s}}{7.9091s^2+s} \end{bmatrix} \tag{32}$$

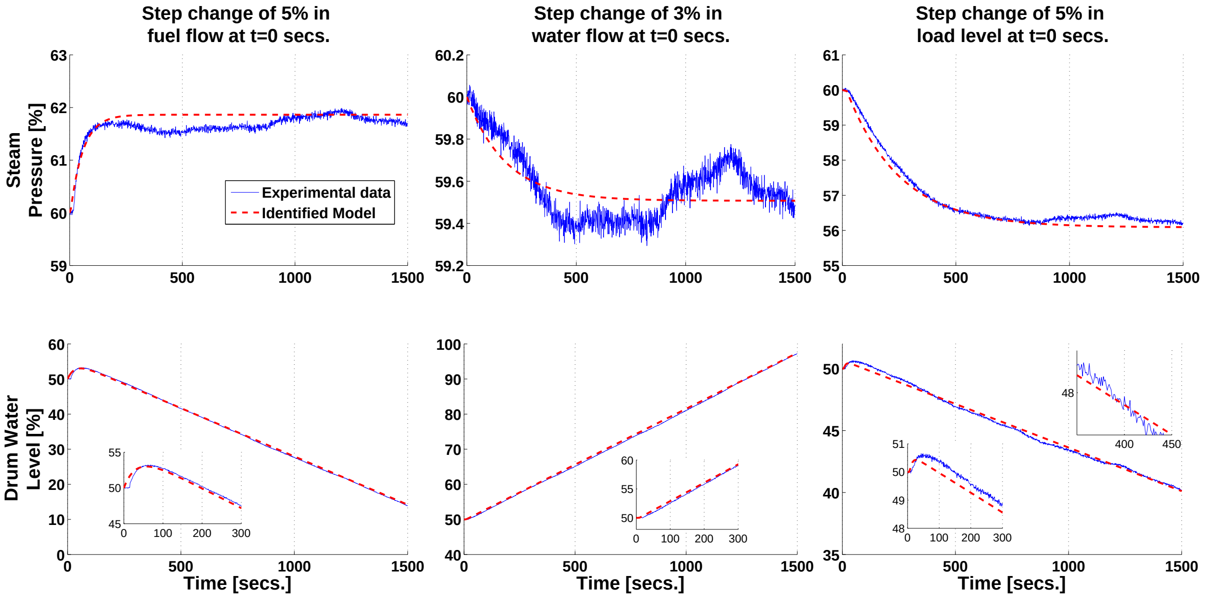


Figure 9: Identified reduced model of the Boiler process).

For this example the MOP definition is:

$$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x}) \quad = \quad [J_1(\boldsymbol{x}), \ldots, J_5(\boldsymbol{x})] \tag{33}$$

subject to:

[9] Nominal linear models have been identified using simple step tests with the Matlab© identification toolbox.

$$0 \leq x_i \quad \leq \quad 1, i = [1, 2, 3, 4] \tag{34}$$

$$L_1(\boldsymbol{x}) = Re[\lambda]_{max} \quad < \quad 0 \tag{35}$$

Where $\boldsymbol{x}$ are the proportional gains and integral time values of the PI controllers. In order to bound the controller parameters, the stochastic sampling described in [40] for stabilizing PI controllers is implemented, and therefore bound constraints are $x_i \in [0, 1]$ for $i = 1, \cdots, 4$; since such coding does not implies the overall stability for a MIMO system, constraint $G_1(\boldsymbol{x})$ for the eigenvalues λ of the overall system is included. The design objectives stated are:

$J_1(\boldsymbol{x})$: Stabilizing time for $Y_1(s)$ at presence of a step Load disturbance $D(s)$.

$J_2(\boldsymbol{x})$: Stabilizing time for $Y_2(s)$ at presence of a step Load disturbance $D(s)$.

$J_3(\boldsymbol{x})$: Biggest log modulus for overall robustness [23]. The criterion is defined as:

$$L_{cm} = 20 \log \left| \frac{W(s)}{1 + W(s)} \right| \leq L_{cm}^{max} \tag{36}$$

where $W(s) = -1 + \det(I + P(s)C(s))$. This criterion proposes a *de-tuning* of the proportional gains of each controller, in order to fulfill a maximum value of the closed loop log modulus L_{cm}^{max}.

$J_4(\boldsymbol{x})$: Maximum value of sensitivity function M_s for loop 1 (Equation (10)).

$J_5(\boldsymbol{x})$: Maximum value of sensitivity function M_s for loop 2 (Equation (10)).

The preference matrix for the design objectives stated is depicted in Table 6. Design objectives to perform a MCDM stage are $J_1(\boldsymbol{x}), J_2(\boldsymbol{x}), J_3(\boldsymbol{x})$. That means that 30 solutions are required and the objective space is just partitioned in three dimensions. It is important to notice that all five objectives are used in the optimisation process in order to calculate the GPP index, since they are included in the preference matrix.

4.2.2. Results and Discussions on the EMO process

In order to test the usefulness of this approach, two different preferences handling mechanisms will evaluated anew:

- The previously defined IB-MODE approach.
- The RP-spMODE algorithm using reference points.

In all cases, parameters used are depicted in Table 7 for each case. Additionally, the following *control experiment* is included:

Table 6: Preferences set for the multivariable benchmark setup. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Set											
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{x})$ [s]	600		700		800		900		1500		2000
$J_2(\boldsymbol{x})$ [s]	600		900		1000		1600		1800		2000
$J_3(\boldsymbol{x})$ [-]	0.0		1		3		6		8		16
$J_4(\boldsymbol{x})$ [-]	0.0		1.2		1.4		1.8		1.9		2.0
$J_5(\boldsymbol{x})$ [-]	0.0		1.2		1.4		1.8		1.9		2.0
$L_1(\boldsymbol{x})$ [-]	<0.0		<0.0		<0.0		0.0		1.0		2.0

- A pure stochastic sampling approach, using the same function evaluations budget. This is used as a base test to evaluate the usability of the approaches above. This is because it has been remarked that sampling procedures could be more effective in many-objective optimisation statements [10].

The aim of this analysis is to evaluate the capabilities of such approaches to approximate a Pareto front in the **T_HypV**, **D_HypV** and **HD_HypV** in order to validate the usefulness of the spMODE-II algorithm for these applications. Same standard CPU is used to calculate the Pareto front approximations for this benchmark.

Table 7: Parameters used for IB-MODE, RP-spMODE and spMODE-II in the multivariable benhcmark setup. Further details in [34]

spMODE-II	RP-spMODE	IB-MODE	Parameter	Value
			Evolutionary mechanism	
			F (Scaling factor)	0.5
			Cr (Crossover rate)	0.5
			N_p (Population)	30
			Pruning mechanism	
			$\boldsymbol{\beta_\epsilon}$ (Arcs)	30
			Pertinency mechanism	
			$\underline{J_{gpp}^{max}}$	$J_{gpp}([900, 1600, 6, 1.8, 1.8, 0])$
			$car(J_P^*)$	30

In Figure 10, distribution plots of the attained **T_HypV** are depicted; in Table 8 numerical values for the best, worst, median, mean and standard deviation of the attained

T_HypV are shown. Statistical significance has been validated using the Wilcoxon test at 95% with Bonferroni correction [15]. As it can be noticed, the approach using the GPP approximates better the **T_HypV**, when compared with the reference point, indicator based and stochastic sampling approaches.

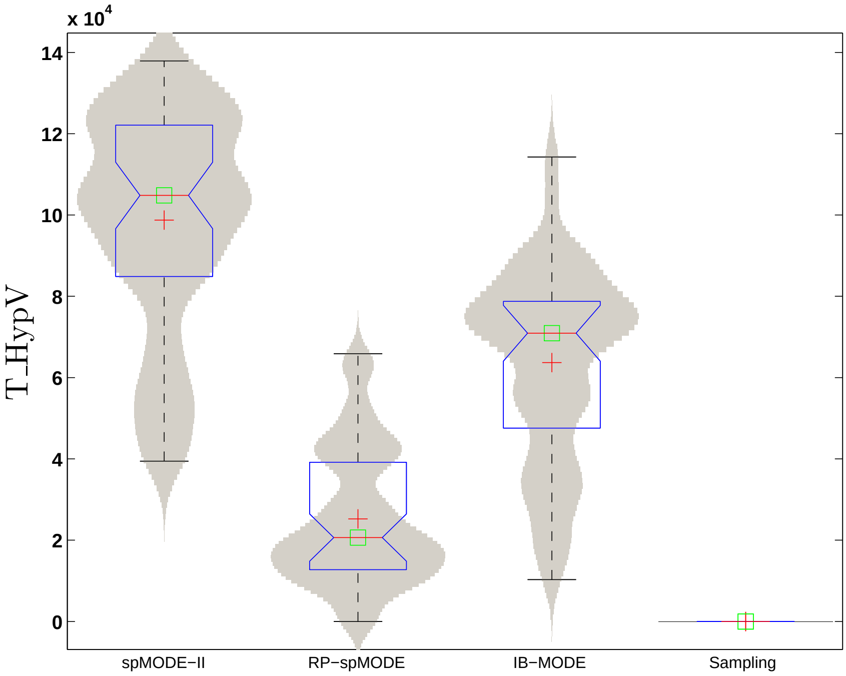
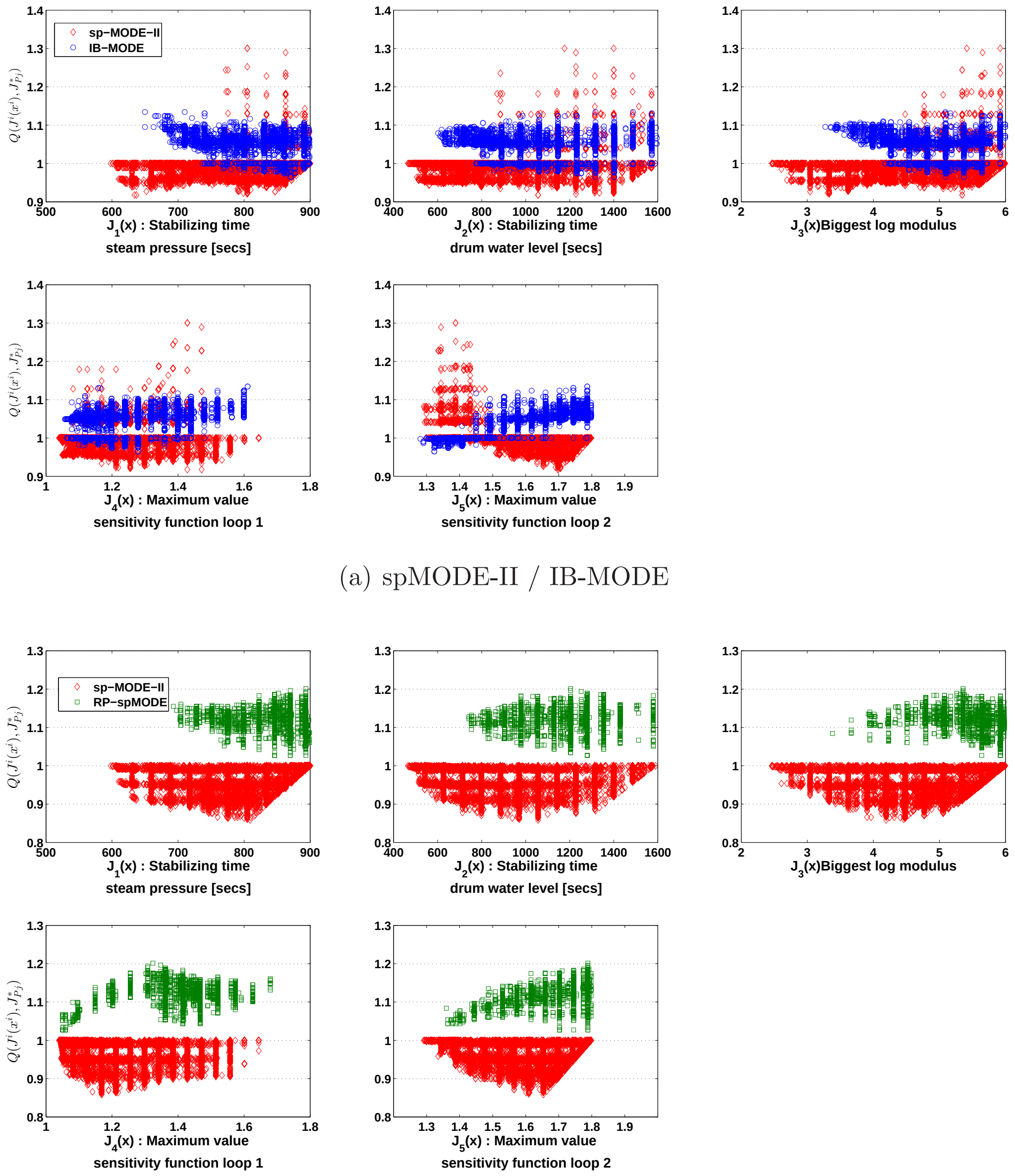


Figure 10: Distribution plots for the tolerable hypervolumen attained by the different approaches used in the benchmark setup.

In Figure 11 attainment surfaces at 50% are compared using level diagrams followings the guidelines of [35] for design concepts comparison. In such visualisation any surface of an approximated Pareto front J^*_{p1} above 1 is dominated by a surface below 1 of the other Pareto front approximation J^*_{p2} and vice versa. In Figure 11a, the GPP approach is compared with the IB-MODE; in such figure, it is possible to appreciate that the main differences between approaches is in the covering of J_5, where each one dominates a portion of the other. In the case of Figure 11b, the GPP approach consistently dominates the RP-MODE.

4.2.3. Results and Discussions on the MOOD procedure

The approximated Pareto front with the median value of the **T_HypV** in the spMODE-II approach is used as representative solution for further analysis in the MCDM step. In Figure 12 such approximation is presented. It is important noting the compactness of the approximated Pareto front towards the **D_HypV**. This facilitates the DM procedure to



(a) spMODE-II / IB-MODE

(b) spMODE-II / RP-spMODE

Figure 11: Attainment surfaces comparison using Level Diagrams for the different approaches used in the benchmark setup.

Table 8: Hypervolume achieved in 51 runs. Statistical significance has been evaluated according to the Wilcoxon.

		spMODE-II	RP-MODE	IB-MODE	Sampling
T_HypV	Best	**1.37e+05**	0.66e+04	1.14e+05	0.00e+00
	Median	**1.04e+05**	0.21e+04	0.71e+05	0.00e+00
	Worst	**0.39e+05**	0.00e+00	0.10e+05	0.00e+00
	Mean	**0.99e+05**	0.25e+05	0.63e+05	0.00e+00
	std	7.16e+08	2.81e+08	5.62+e08	0.00e+00

select a solution to implement. Also the simulation response of the approximated design alternatives is depicted in Figure 13.

The preference set stated (Table 6) to calculate the GPP index has been useful to incorporate the designer's preferences among conflicting objectives. Nevertheless, although this index has been useful to evolve the evolutionary process towards the pertinent Pareto front, it is not capable to describe by itself the trade-off among objectives. That is, since the Pareto front is unknown *a priori*, it is impossible to predict its shape by just analysing the lowest GPP index obtained. Therefore, the MCDM stage is still a necessary step for the designer. By using the spMODE-II, it has been assured to have a pertinent approximation of the Pareto front, with an interesting set of solutions from the point of view of the designer. By consequence, it facilitates the decision making process.

Two solutions are marked in the Figure 12: the design alternative with the lowest GPP index (▲ in the approximated Pareto front) and a solution which sacrifices a desirable value in $J_5(\boldsymbol{x})$ (individual loop performance) in order to get a better overall performance $J_3(\boldsymbol{x})$ (★ in the approximated Pareto front). The differences between both design alternatives can be appreciated in Figure 13; the latter has a smooth response in the drum water level when compared with the former. Given the above, the latter design alternative is preferred over the former with the minimum GPP index.

This design alternative has been implemented in the *real process*, and the performance index defined by the benchmark $I_{benchmark}(C_e, C_r, \omega)$ is shown in Table 9. Such index is an aggregate objective function, which combines ratios of the IAE (13), ITAE (14) and TV[10] (15). In order to evaluate a controller C_e, indexes are referred to a base case controller C_r where a weighting factor ω for the ratios of the control action values ($\omega \dot{T}V$) is included. Further details are available in [28].

In the original benchmark, two PI controllers $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2}] = [2.5, 50, 1.25, 50]$ are used as C_r, and the weighting factor is proposed as $\omega = 0.25$. Two different tests are proposed:

Test 1: Performance when the system had to attend a time variant load level.

Test 2: Performance when the system had to attend a sudden change in the steam pressure set-point.

[10] also known as IADU

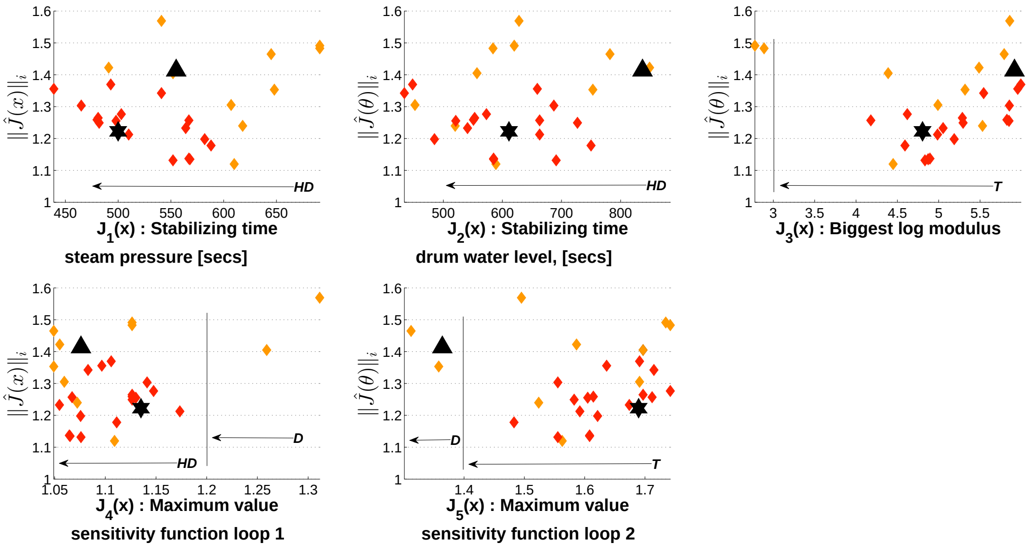


Figure 12: Pareto Front approximation of the benchmark setup. Design alternative with the lowest GPP index (▲) and the selected design alternative (★) are depicted.

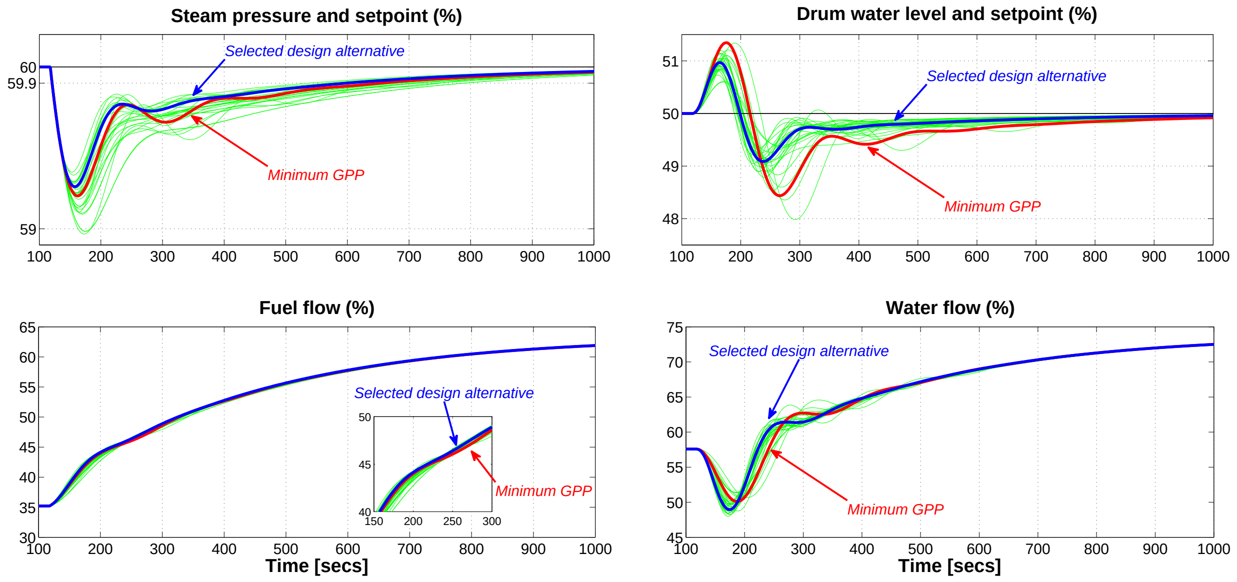


Figure 13: Simulation performance of the approximated Pareto front from Figure 12.

Table 9: Performance achieved of the selected design alternative (no filter) for the benchmark setup. Ratios of IAE (RIAE), ITAE (RITAE) e IADU (RIADU) with respect to the PI reference case are depicted.

	RIAE1	RIAE2	RIAE3	RITAE1	RITAE3	RIADU1	RIADU2	$I_{benchmark}(C_e, C_r, 0.25)$
Test1	0.8032	0.9991	0.5581	0.0000	0.0000	1.8736	9.8487	1.5117
Test2	0.7882	0.9647	0.2644	0.7137	0.0000	2.4576	4.8029	1.0169

Table 10: Performance achieved of the selected design alternative ($\tau_f = 10$) for the benchmark setup. Ratios of IAE (RIAE), ITAE (RITAE) e IADU (RIADU) with respect to the PI reference case are depicted.

	RIAE1	RIAE2	RIAE3	RITAE1	RITAE3	RIAVU1	RIAVU2	$I_{benchmark}(C_e, C_r, 0.25)$
Test1	0.8042	0.9991	0.5614	0.0000	0.0000	0.9637	1.2320	0.8325
Test2	0.7093	0.9673	0.3307	0.7085	0.0000	0.7574	1.4825	0.7279

Firstly, the selected controller without filtering the measured signal is employed. Notice that performance indexes related with ratios of IAE and ITAE are better than the reference controller. Nevertheless, the performance indicator $I_{benchmark}(C_e, C_r, \omega)$ is worse in the case of Test 1 and it has almost the same performance than Test 2 (see Table 9). This is due to the weighting factor used for the control action; the design alternative selected is more sensitive to noise and therefore, the IADU ratio is bigger.

Using a first order filter with $\tau_f = 10$ for the measured signal, the performance related to the control action is improved[11] and as a consequence, the overall index $I_{benchmark}(C_e, C_r, \omega)$ (Figures 15 and 17 and Table 10). Therefore, the proposed PI controllers have a performance (regarding this metric) which is better than the reference controllers.

In summary, the methodology is effective, bringing a controller fulfilling all the requirements with a better performance than the reference controller. A comparison in equal conditions with other control solutions dealing with the boiler benchmark is not possible. In [18] a feedforward mechanism is used that is not included in this proposal; in [44] a data driven approach is used (*i.e.* a *on the fly* tuning technique); in [30] a 2x2 PI controller matrix is proposed; and finally in [46] and [42] results reported are not evaluated under the benchmark guidelines.

5. Conclusion

A MOOD procedure for controller tuning was presented in this paper. Such approach uses preferences, in order to focus the evolutionary search towards the interested region of the Pareto front. The MOOD procedure is a powerful tool to analyse objective exchanges and select a preferable solution. Nevertheless, it is a valuable tool when (1) it is difficult to find a controller with a reasonable balance among design objectives; (2) it is worthwhile

[11] Such filter still guarantees overall stability in the control loop for the nominal process of Equation (31)

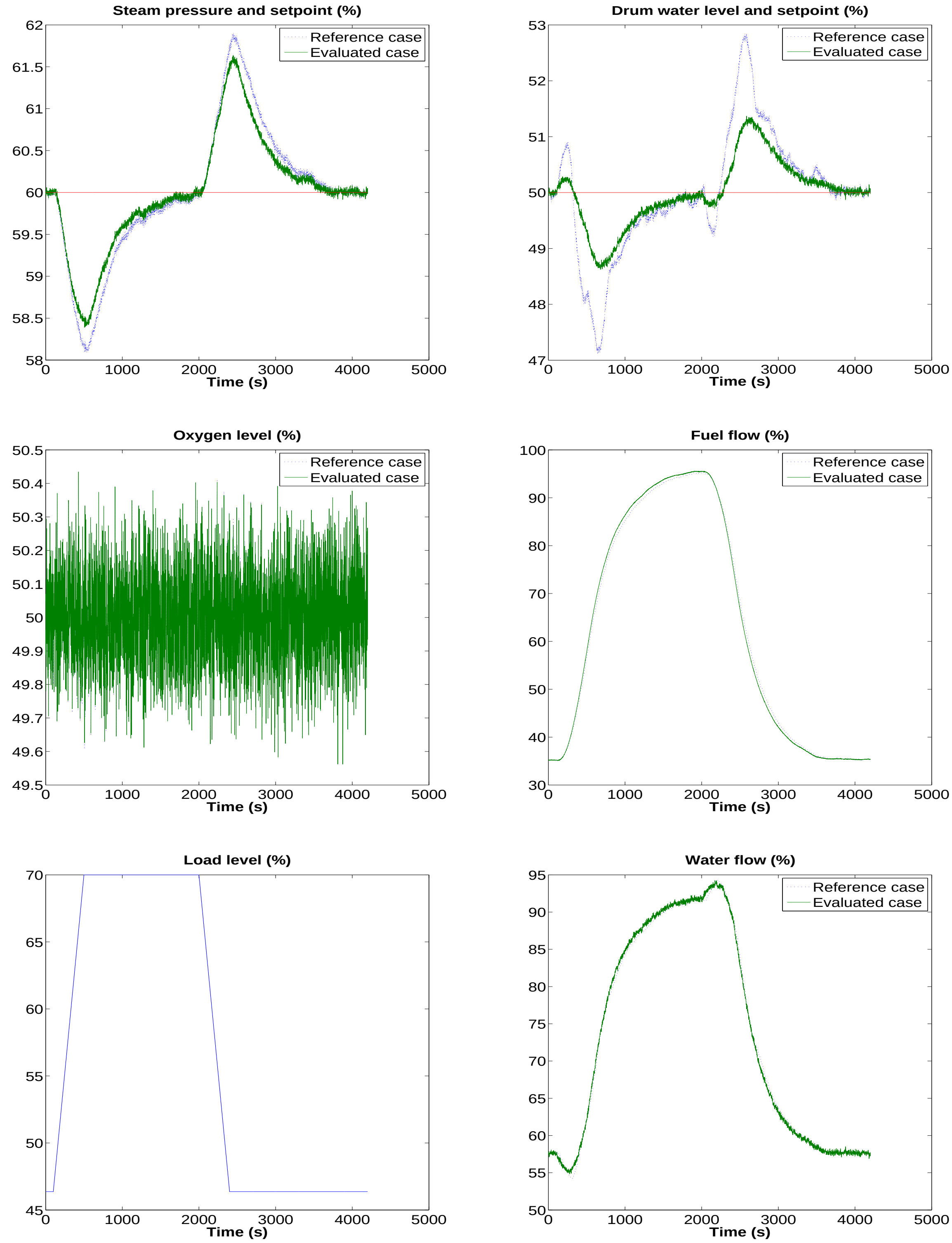


Figure 14: Performance for the Test 1 of the PI controller $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [1.533, 29.549, 5.315, 125.778]$ (without filter) and its comparison with the reference case $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2}] = [2.5, 50, 1.25, 50]$ in the benchmark setup.

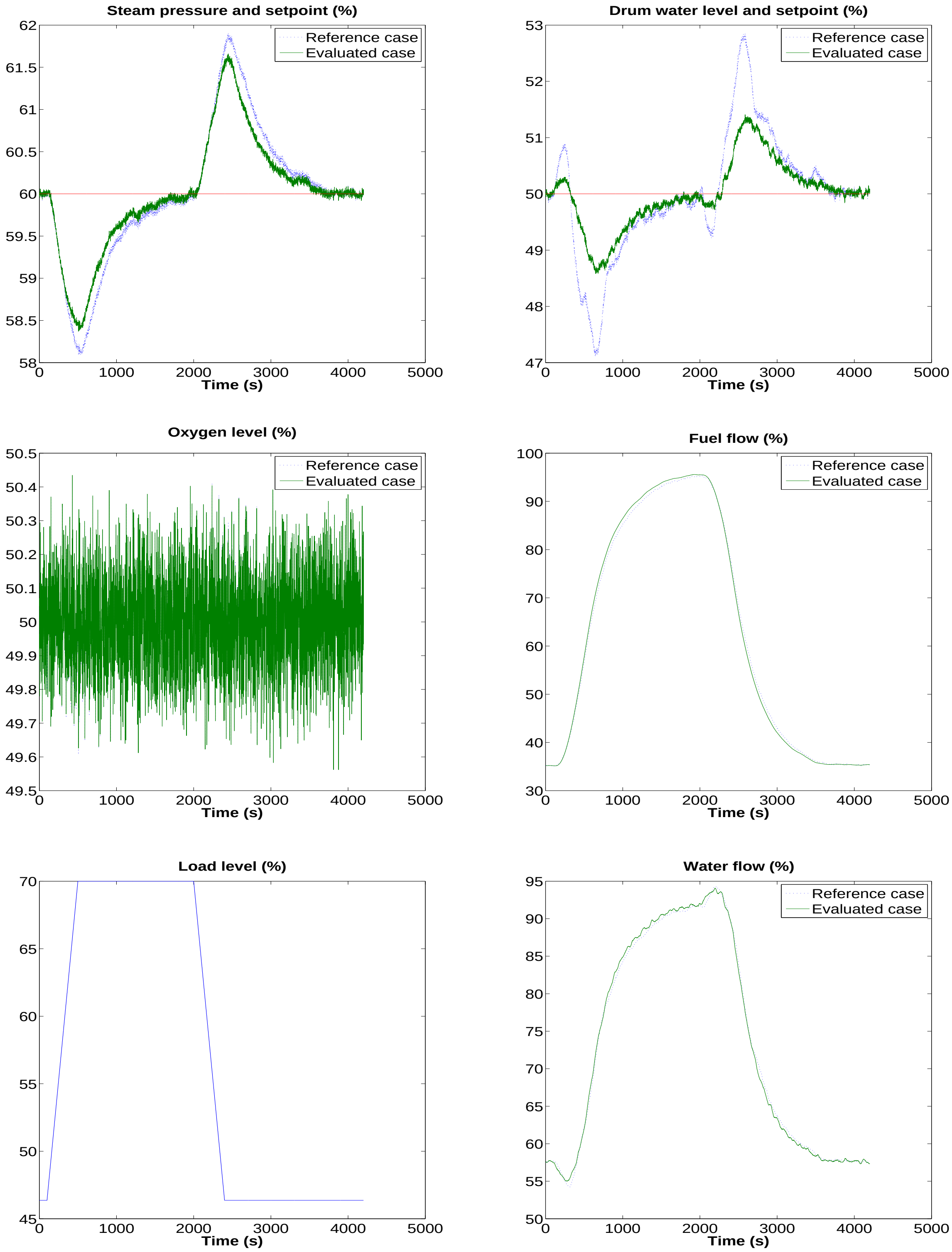


Figure 15: Performance for the Test 1 of the PI controller $[k_p, T_i, k_p, T_i] = [1.533, 29.549, 5.315, 125.778]$ ($\tau_f = 10$) and its comparison with the reference case $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2}] = [2.5, 50, 1.25, 50]$ in the benchmark setup.

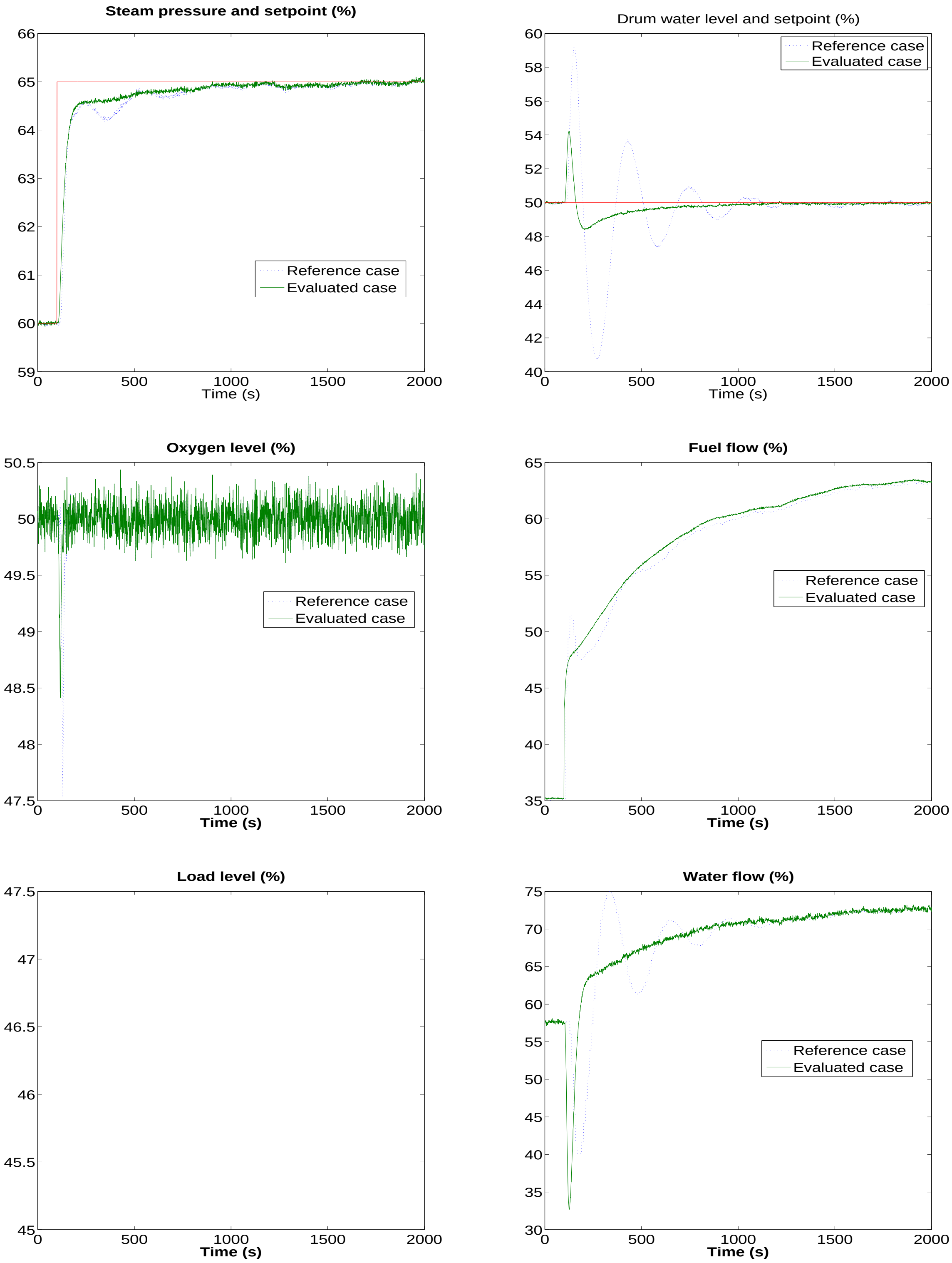


Figure 16: Performance for the Test 2 of the PI controller $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [1.533, 29.549, 5.315, 125.778]$ (without filter) and its comparison with the reference case $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [2.5, 50, 1.25, 50]$ in the benchmark setup.

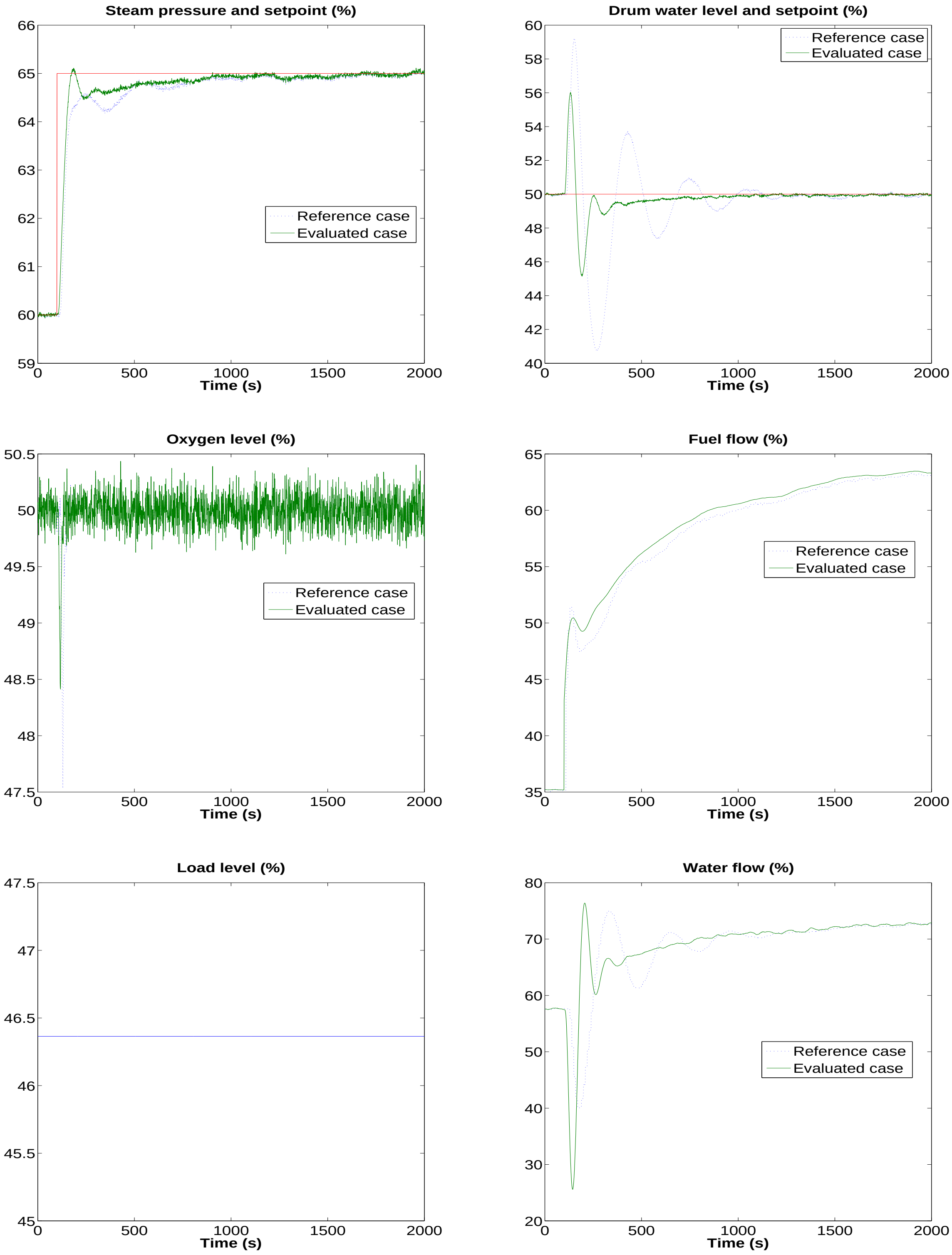


Figure 17: Performance for the Test 2 of the PI controller $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [1.533, 29.549, 5.315, 125.778]$ ($\tau_f = 10$) and its comparison with the reference case $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [2.5, 50, 1.25, 50]$ in the benchmark setup.

analysing the trade-off among controllers (design alternatives).

Three key points for its success are the following:

1. Approximating a compact set of solutions. Ten times the number of objectives seem to be a reasonable size for a Pareto front approximation.
2. Beyond approximating a compact set, approximating a compact and pertinent set. For this, it is fundamental to have an understanding of the objectives to define a preference range.[12] If the DM has no idea on such values, that could be an indicative of a perfunctory or precipitate selection of the design objectives. Therefore, perhaps the DM should ponder the design objectives stated.
3. Deciding where to perform the DM stage. In addition to design objectives and constraints, a third category is included in this work: objectives that should be minimised and taken into account during the search process, but which are not meant to be used for decision making.

Limitations and future work are:

- A strategy to design feedforward compensators should be included in the overall MOOD procedure for controller tuning.
- More complex control tuning strategies should be evaluated and compared.
- It should be worthwhile to evaluate different optimisation instances for multivariable controller tuning as multidisciplinary or reliable based statements.
- While the LD visualisation is a powerful tool to analyse an m-dimensional Pareto front, it was also required to incorporate information from the time response of the approximated (and pertinent) Pareto front. Therefore, it seems to be a promising area for development to build visualisations approaches for the specific application of controller tuning.

Acknowledgment

This work was partially supported by projects TIN2011-28082, ENE2011-25900 from the Spanish Ministry of Economy and Competitiveness. First author gratefully acknowledges the partial support provided by the postdoctoral fellowship BJT-304804/2014-2 from the National Council of Scientific and Technologic Development of Brazil (CNPq) for the development of this work.

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

[12] Or at least, the tolerable values for design objectives.

[2] Bell, R., Åström, K. J., jun 1987. Dynamic models for boiler-turbine alternator units: Data logs and parameter estimation for a 160 MW unit". Technical Report ISRN LUTFD2/TFRT--3192--SE, Department of Automatic Control, Lund University, Sweden.

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

[4] Cela, R., Bollaín, M., 2012. New cluster mapping tools for the graphical assessment of non-dominated solutions in multi-objective optimization. Chemometrics and Intelligent Laboratory Systems 114 (0), 72 – 86.

[5] Chica, M., Cordón, Ó., Damas, S., Bautista, J., 2014. Interactive preferences in multiobjective ant colony optimisation for assembly line balancing. Soft Computing, 1–13.

[6] Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K., 2015. An interactive simple indicator-based evolutionary algorithm (i-sibea) for multiobjective optimization problems. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C. C. (Eds.), Evolutionary Multi-Criterion Optimization. Vol. 9018 of Lecture Notes in Computer Science. Springer International Publishing, pp. 277–291.

[7] 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.

[8] Coello, C. A. C., Lamont, G. B., 2004. Applications of Multi-Objective evolutionary algorithms, advances in natural computation vol. 1 Edition. World scientific publishing.

[9] Coello, C. A. C., Veldhuizen, D. V., Lamont, G., 2002. Evolutionary algorithms for solving multi-objective problems. Kluwer Academic press.

[10] 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.

[11] Cvetkovic, D., Parmee, I., 2002. Preferences and their application in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation 6 (1), 42 –57.

[12] Das, I., Dennis, J., 1998. Normal-boundary intersection: a new method for generating the Pareto surface in non-linear multicriteria optimization problems. SIAM Journal on Optimization 8, 631 – 657.

[13] de Freitas, A. R., Fleming, P. J., Guimar$\tilde{a}$es, F. G., 2015. Aggregation trees for visualization and dimension reduction in many-objective optimization. Information Sciences 298, 288 – 314.

[14] Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S., 2006. Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research 2 (3), 273–286.

[15] 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.

[16] Fernández, I., Rodríguez, C., Guzman, J., Berenguel, M., 2011. Decoupled predictive control algorithm with disturbance compensation for the benchmark of 2009-2010 (in spanish). Revista Iberoamericana de Automática e Informática Industrial Apr. 8 (2), 112 – 121.

[17] Fleming, P., Purshouse, R., 2002. Evolutionary algorithms in control systems engineering: a survey. Control Engineering Practice (10), 1223 – 1241.

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

[19] Huang, L., Wang, N., Zhao, J.-H., 2008. Multiobjective optimization for controller design. Acta Automatica Sinica 34 (4), 472 – 477.

[20] Inselberg, A., 1985. The plane with parallel coordinates. The Visual Computer 1, 69–91.

[21] 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.

[22] Kaliszewski, I., Miroforidis, J., Podkopaev, D., 2012. Interactive multiple criteria decision making based on preference driven evolutionary multiobjective optimization with controllable accuracy. European Journal of Operational Research 216 (1), 188–199.

[23] Luyben, W. L., 1986. Simple method for tuning SISO controllers in multivariable systems. Industrial and Engineering Chemistry Process Design (25), 654 – 660.

[24] Mattson, C. A., Messac, A., 2005. Pareto frontier based concept selection under uncertainty, with visualization. Optimization and Engineering 6, 85–115.

[25] Messac, A., Ismail-Yahaya, A., Mattson, C., 2003. The normalized normal constraint method for generating the Pareto frontier. Structural and Multidisciplinary Optimization (25), 86 – 98.

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

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

[28] Morilla, F., 2012. Benchmark for PID control based on the boiler control problem. Available at `http://www.dia.uned.es/~fmorilla/benchmark09_10/`, internal report, UNED Spain.

[29] 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.

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

[31] 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.

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

[33] Qu, B., Suganthan, P., 2010. Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection. Information Sciences 180 (17), 3170 – 3181, including Special Section on Virtual Agent and Organization Modeling: Theory and Applications.

[34] Reynoso-Meza, G., June 2014. Controller tuning by means of evolutionary multiobjective optimization: a holistic multiobjective optimization design procedure. Ph.D. thesis, Universitat Politècnica de València.

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

[36] Reynoso-Meza, G., Blasco, X., Sanchis, J., Martínez, M., 2013. Evolutionary algorithms for PID controller tuning: Current trends and perspectives (in spanish). Revista Iberoamericana de Automática e Informática Industrial 10 (3), 251 – 268.

[37] Reynoso-Meza, G., Sanchis, J., Blasco, X., 2009. An adaptive parameter for the differential evolution algoritm. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J. M.

(Eds.), Bio-inspired systems: computational and ambient intelligence. Vol. LNCS 5517. Springer-Verlag, pp. 375 – 382.

[38] Reynoso-Meza, G., Sanchis, J., Blasco, X., García-Nieto, S., 2014. Physical programming for preference driven evolutionary multi-objective optimization. Applied Soft Computing 24, 341 – 362.

[39] Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J. M., 2012. Multiobjective evolutionary algortihms for multivariable PI controller tuning. Expert Systems with Applications 39, 7895 – 7907.

[40] Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J. M., August 2014. A stabilizing PID controller sampling procedure for stochastic optimizers. In: Memories of the 19th World Congress IFAC 2014.

[41] Reynoso-Meza, G., Sanchis, J., Blasco, X., Martínez, M., 2014. Controller tuning using evolutionary multi-objective optimisation: current trends and applications. Control Engineering Practice 1, 58 – 73.

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

[43] Ruiz, A., Luque, M., Miettinen, K., Saborido, R., 2015. An interactive evolutionary multiobjective optimization method: Interactive wasf-ga. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C. C. (Eds.), Evolutionary Multi-Criterion Optimization. Vol. 9019 of Lecture Notes in Computer Science. Springer International Publishing, pp. 249–263.

[44] Saeki, M., Ogawa, K., Wada, N., March 2012. Application of data-driven loop-shaping method to multi-loop control design of benchmark PID 2012. In: Proceedings of the IFAC Conference on Advances in PID Control (PID'12).

[45] Saridakis, K., Dentsoras, A., 2008. Soft computing in engineering design - a review. Advanced Engineering Informatics 22 (2), 202 – 221, network methods in engineering.

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

[47] Thiele, L., Miettinen, K., Korhonen, P. J., Molina, J., 2009. A preference-based evolutionary algorithms for multi-objective optimization. Evolutionary Computation (3), 411 – 436.

[48] Tušar, T., Filipič, B., 2015. Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method 19 (2), 225 – 245.

[49] Wang, Y., Yang, Y., 2009. Particle swarm optimization with preference order ranking for multi-objective optimization. Information Sciences 179 (12), 1944 – 1959, special Section: Web Search.

[50] Zhang, J., Zhuang, J., Du, H., Wang, S., 2009. Self-organizing genetic algorithm based tuning of PID controllers. Information Sciences 179 (7), 1007 – 1018.

[51] Zhao, S.-Z., Iruthayarajan, M. W., Baskar, S., Suganthan, P., 2011. Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization. Information Sciences 181 (16), 3323 – 3335.

[52] Zitzler, E., Künzli, S., 2004. Indicator-based selection in multiobjective search. In: Yao, X. e. A. (Ed.), Parallel Problem Solving from Nature - PPSN VIII. Vol. 3242 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 832–842, 10.1007/978-3-540-30217-9 84.

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