Preprints, 4th IFAC Conference on
Advances in Proportional-Integral-Derivative Control
Almería, Spain | June 12-14, 2024

Controller tuning with multi-objective optimization: benchmark definitions for SISO linear systems: A PI controller case ⋆

Gilberto Reynoso-Meza *,** **Ricardo Massao Kagami** *

* *Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Paraná (PUCPR). (e-mail: g.reynosomeza@pucpr.br.*
** *Control Systems Optimization Laboratory (LOSC), Pontifícia Universidade Católica do Paraná (PUCPR).*

Abstract: Multi-objective optimization techniques are practical techniques for controller tuning purposes. A wide variety of papers use such an approach or propose new algorithms to approximate the Pareto front. Nevertheless, despite the expressive volume of works dealing with it, there is no standard benchmark testbed that could be used as a baseline comparison among different techniques. This work addresses such a gap, with a first step proposing a linear-SISO testbed with guidelines and rules and providing a basic example for a PI controller. Such a benchmark is expected to promote further research on the topic.

Keywords: Multi-objective optimization, Parametric optimization, Evolutionary Algorithms.

1. INTRODUCTION

Controller tuning is a multi-objective task. That means several requirements, constraints, and specifications must be fulfilled. Such demands are usually in conflict, and control engineers must propose trade-off solutions.

For single-input, single-output (SISO) processes, it is usual to achieve a trade-off between performance and robustness. That is because the tuning process requires a model (M) of the process. Therefore, such balance is relevant to guarantee a successful implementation of the controller. For the multi-input, multi-output (MIMO) case, besides such a trade-off, it is usual to encounter that, given coupled dynamics, an improvement in one of the outputs implies the degradation in a different output.

Multi-objective optimization is a suitable way to deal with such a situation. A design objective vector is stated instead of defining a single cost function. Then, all design objectives are optimized simultaneously. Consequently, a set of design alternatives with different trade-offs is calculated instead of a single solution. Such solutions are Pareto optimal: it is only possible to improve an objective by worsening another. Such an approach has been used with success in control systems engineering (Reynoso-Meza et al., 2014; Moarref et al., 2016; Lim et al., 2017; Cao et al., 2017).

Different reviews collect and summarise the advantages of such approaches. Nevertheless, there needs to be more standardization to evaluate the performance of a given algorithm. Benchmarks allow researchers to have a higher degree of reproducibility and comparability of diverse control techniques(Kroll and Schulte, 2014). Even if it is possible to find literature on control engineering benchmarks (Dixon and Pike, 2006; Bejarano et al., 2017; Kroll and Schulte, 2014; Mercader et al., 2019; Romero and Sanchis, 2011; Fernández et al., 2011; Atanasijevic-Kunc et al., 2010; Eriksson et al., 2019) and solutions involving multi-objective techniques (Xue et al., 2010; Kagami et al., 2019), there is yet to be a specific benchmark dedicated to evaluating the performance of multi-objective optimization algorithms.

This paper aims to provide a first framework to test multi-objective optimization algorithms and PI controllers. The remainder of this paper is as follows: Section 2 provides a brief background on multi-objective optimization techniques. In Section 3, the benchmark proposal is presented; in Section 4, an example using a PI controller is provided. Finally, some concluding remarks and future work are discussed.

2. BACKGROUND

In this section, a brief background on multi-objective optimization techniques is provided.

2.1 Multi-objective optimization

There is not a single solution for a multi-objective problem (MOP) because there is not generally a better solution for all the objectives. Therefore, a Pareto set $\boldsymbol{X_P}$ is approximated. Each solution in this set defines an objective vector in the Pareto front $\boldsymbol{J_P}$ (See Figure 1).

Successful implementation of multi-objective optimization techniques requires (at least) three steps: the MOP statement, the multi-objective optimization (MOO) process,

⋆ This work was partially supported by the Conselho Nacional de Pesquisa e Desenvolvimento (CNPq) - Brazil - Finance Codes: [310195/2022-5-PQ2], [UNIVERSAL 4408164/2021-2].

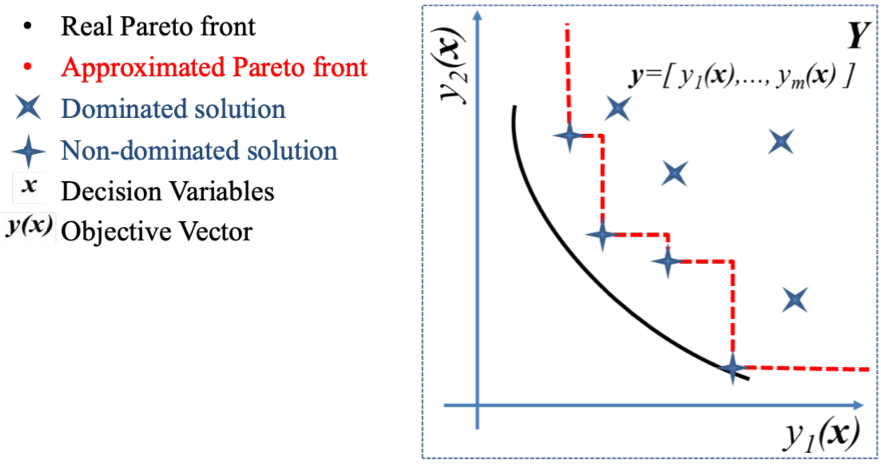

Fig. 1. Pareto front approximation concept.

and the multi-criteria decision-making (MCDM) stage. Next, a brief description of such steps is given. Readers are referred to Meza et al. (2016) for a complete description of such stages.

Multi-objective problem definition In the MOP statement, the designer must define design objectives and constraints. Additionally, it requires a parametric model and a cost function to calculate such information.

Without loss of generality, a multi-objective optimization statement is defined as follows:

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

subject to:

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

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

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

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ is defined as the decision vector with $\dim(\boldsymbol{x}) = n$; $\boldsymbol{J}(\boldsymbol{x})$ 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 the upper bounds in the decision space.

Multi-objective optimization process The MOO process aims to approximate the Pareto front and set using a given algorithm. All the solutions in the Pareto front are a set of Pareto optimal and non-dominated solutions.

- Pareto optimality (Miettinen, 1999): 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, $j \in [1, 2, \ldots, m]$.
- Dominance (Coello and Lamont, 2004): Given two objective vectors $\boldsymbol{J}(\boldsymbol{x}^1)$, $\boldsymbol{J}(\boldsymbol{x}^2)$, objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is dominated by objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ if $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$. This is denoted as $\boldsymbol{J}(\boldsymbol{x_2}) \preceq \boldsymbol{J}(\boldsymbol{x_1})$.

Most of the algorithms use such definitions in their optimization processes to approximate the Pareto front. Usually, we must rely entirely on approximations $\boldsymbol{J_P^*}, \boldsymbol{X_P^*}$ since the actual Pareto front is usually unknown. Different alternatives for algorithms exist, ranging from classical optimization techniques (Messac et al., 2003) to evolutionary multi-objective optimization (Coello, 2003).

2.2 *Controller tuning as a multi-objective problem*

A basic control loop is depicted in Figure 2. 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 $R(s)$. The controller $C(s)$ will achieve this task, using error $E(s)$ as information to compute a control action $U(s)$. Block $H(s)$ is a measuring instrument, usually considered unitary.

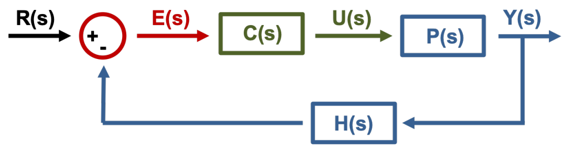

Fig. 2. Basic control loop.

The control problem consists of selecting adequate parameters of the proposed controller $C(s)$ in order to achieve a desirable performance using a model $M(s)$ of the process in the control loop, as well as robust stability margins to guarantee a reasonable performance when controlling the process $P(s)$. As commented by Garpinger et al. (2014), conflicting objectives may appear when seeking a good performance and a desirable robustness level. Since those design objectives are usually conflicting, MOO techniques could appeal to controller tuning.

3. BENCHMARK PROPOSAL

Next, we will present the experimental setup under consideration. Later, definitions and conditions to evaluate algorithms will be presented.

3.1 SISO processes

The set consists of a series of single-input, single-output processes. The test set proposed by Åström and Hägglund (2004) is used for such a purpose:

$$P_1(s) = \frac{e^{-s}}{1+sT}, \quad (5)$$
$$T \in \{0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 1.3, 1.5, 2, 4, 6, 8, 10, 20, 50, 100, 200, 500, 1000\}$$
$$P_2(s) = \frac{e^{-s}}{(1+sT)^2}, \quad (6)$$
$$T \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 1.3, 1.5, 2, 4, 6, 8, 10, 20, 50, 100, 200, 500\}$$
$$P_3(s) = \frac{1}{(s+1)(1+sT)^2}, \quad (7)$$
$$T \in \{0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 2, 5, 10\}$$
$$P_4(s) = \frac{1}{(s+1)^n}, \quad (8)$$
$$n \in \{3, 4, 5, 6, 7, 8\}$$
$$P_5(s) = \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}, \quad (9)$$
$$\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$$
$$P_6(s) = \frac{e^{-sL_1}}{s(1+sT_1)}, T_1 + L_1 = 1, \quad (10)$$
$$L_1 \in \{0.01, 0.02, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$$
$$P_7(s) = \frac{T}{(1+sT)(1+sT_1)} e^{-sL_1}, T_1 + L_1 = 1, \quad (11)$$
$$T \in \{1, 2, 5, 10\}$$
$$L_1 \quad \{0.01, 0.02, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$$
$$P_8(s) = \frac{1-\alpha s}{(s+1)^3}, \quad (12)$$
$$\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1\}$$
$$P_9(s) = \frac{1}{(s+1)((sT)^2 + 1.4sT + 1)}, \quad (13)$$
$$T \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$$

3.2 Multi-objective problem statement

Only time performance measures are considered, allowing design concept comparisons (*i.e.* different controller structures).

- Design objectives:

$$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x}) = [\overline{J_{IAE}(\boldsymbol{x})}, \overline{J_{TV}(\boldsymbol{x})}] \quad (14)$$

Where:

$$J_1(\boldsymbol{x}) = \overline{J_{IAE}(\boldsymbol{x})} = \frac{J_{IAE}(\boldsymbol{x})}{J_{IAE}(\boldsymbol{x_{BL}})} \quad (15)$$
$$J_2(\boldsymbol{x}) = \overline{J_{TV}(\boldsymbol{x})} = \frac{J_{TV}(\boldsymbol{x})}{J_{TV}(\boldsymbol{x_{BL}})} \quad (16)$$

where $\boldsymbol{x_{BL}}$ is the base-line controller for process $P_i(s), i \in \{1, 2, \cdots, 9\}$. IAE stands for the integral of the absolute value of the error, and TV for the total variation of the control action.

- Constraints:

$$\overline{J_{IAE}(\boldsymbol{x})} \leq 1 \quad (17)$$
$$\overline{J_{TV}(\boldsymbol{x})} \leq 1 \quad (18)$$

Such constraints will bring pertinency and interpretability for visualization purposes. On the one hand, meeting the constraints means that only the portion of the Pareto front dominating the baseline controller is accepted. On the other hand, this will give some degree of interpretability regarding how much improvement is obtained. Stability must be guaranteed in the approximated Pareto front.

It is relevant to notice that bounds for decision variables are not declared. This is due to the diversity of the processes. Additionally, the right choice of boundaries is not a straightforward issue, and checking and evaluating proposals in this matter is interesting.

- Base line controllers

The baseline controller suggested is the Proportional-Integral (PI) controller tuned via the Ziegler-Nichols procedure, using ultimate gain and period of the process $P_{1,\cdots,9}(s)$. This way it is possible to approximate a Pareto front dominating such a baseline solution.

- Profile test

Following a recommendation from Åström et al. (1998) for an optimization statement, most of the industrial control loops are intended to reject disturbances. Therefore, a unitary disturbance for each output of the process is defined.

- Optimization conditions

A budget of 1000 function evaluations (simulations) is defined. The rationale behind this number is that complex processes and simulations are being used nowadays as parametric models of control systems. Therefore, it must be considered that such simulations could be computationally expensive. That could also promote research on surrogate models (Peitz and Dellnitz, 2018) for such instances.

- Convergence performance measures for population based optimization algorithms

Hypervolume (Auger et al., 2012) is used to measure the performance of population based optimization algorithms used to approximate the Pareto front. Intuitively, it is the volume enclosed by the Pareto front approximation, using a *nadir* point as a reference. Hypervolume is selected due to its capabilities to measure convergence and diversity.

If population-based algorithms are used, 11 runs should be reported, indicating the best, worst, median, mean, and standard deviation values. If local algorithms are used, initial solution selection should be indicated.

Using the normalized objectives defined above, the *nadir* point correspond to the a vector $nadir = \overbrace{[1, \cdots, 1]}^{m}$. Such selection will allow us to evaluate pertinency and ensure understanding due to different scales among design objectives. As a baseline algorithm, it is also suggested to include a random strategy, using the same budget of 1000 function evaluations and 11 runs. This will give information regarding the advantage (or lack of) of using a given optimization strategy. Lastly, to evaluate the computational complexity of the algorithms proposed, a CPU-time normalization is proposed, using the time required to sample 1000 solutions (random approach).

- Platform

A platform is provided using Matlab© and Simulink© to perform the tests [1]. It is essential to notice that the integration step must be the same within tests for comparison purposes.

4. EXAMPLE

To illustrate how to report the experimental setup, a performance evaluation of a simple random approach is provided.

4.1 Algorithm description

A random approach is used for comparative purposes. A uniform sampling will be carried out within boundaries for each one of the processes under study. Boundaries are defined as: $kp = [0, K_u]$ and $ki = [0, I_u]$, where Ku is the ultimate gain of the process $P_i(s)$ and I_u is the ultimate gain for the transfer function $\frac{1}{s} \cdot P_i(s)$. Matlab and Simulink 2021B are used in a DELL Precision 3561, 11th. Generation, Intel Core i7-11800H, 2.3 GHz and 16GB RAM.

4.2 Results and discussion

Figures 3, 4, 5, 6, 7, 8, 9, 10 and 11 depicts the Hypervolume results of the random approach for each family. Due to space limitations, only graphical visualization is commented on here.

It is possible to notice diverse behaviors:

- For the process family $P_1(s)$, the greater the time constant value, the lower the variability in the hypervolume, with a medium value around 0.55 and 0.65;
- For the process family $P_2(s)$, the greater the time constant value, the lower the variability and the higher the hypervolume;
- For families $P_3(s)$, $P_4(s)$ and $P_5(s)$ the hypervolume has very low variability intra-members in each family;
- For family $P_6(s)$, each member has a very low variability, with some members easily improving the Ziegler Nichols baseline controller;
- For family $P_7(s)$, the performance is quite diverse due to the reason that two parameters are used to build this transfer function;
- For family $P_8(s)$ the variability increases with the α value of the non-minimum phase;
- For family $P_9(s)$, the higher the T value, the higher the hypervolume's variability and the medium value.

[1] Available at: https://www.mathworks.com/matlabcentral/profile/authors/2438888

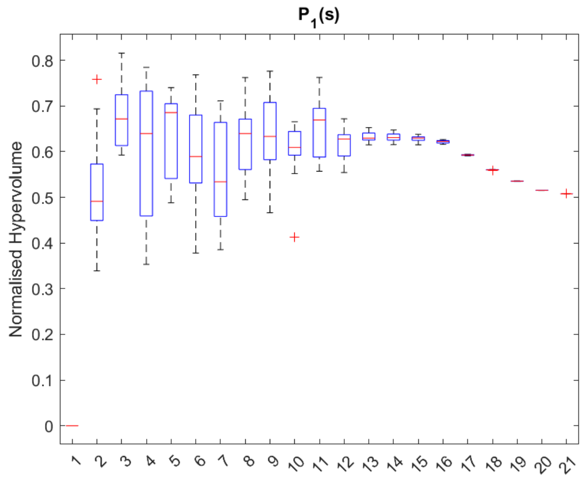

Fig. 3. Hypervolume $P_1(s)$.

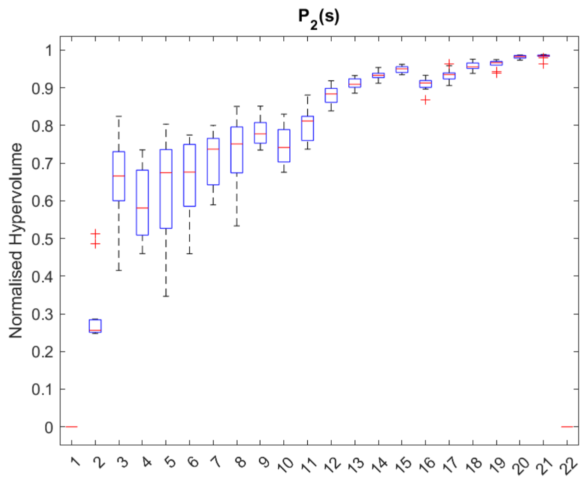

Fig. 4. Hypervolume $P_2(s)$.

The above-commented conclusions drawn from the dispersion plots give an idea regarding the value of using more elaborated methods to approximate the Pareto front. Therefore, they might give additional insights regarding which kind of process the practitioners of the area should focus on.

5. CONCLUSIONS

This paper proposes a testbed benchmark for multi-objective controller tuning for linear SISO processes. This benchmark states different guidelines and rules, allowing a comparison of different algorithms with a limited budget of function evaluations. Particularly, a PI controller case, was presented, but different structures could also be evaluated. A more comprehensive testbed, including MIMO and SISO non-linear processes, is under development to cover a wider variety of cases in controller tuning.

REFERENCES

Åström, K.J. and Hägglund, T. (2004). Revisiting the ziegler–nichols step response method for pid control. *Journal of process control*, 14(6), 635–650.

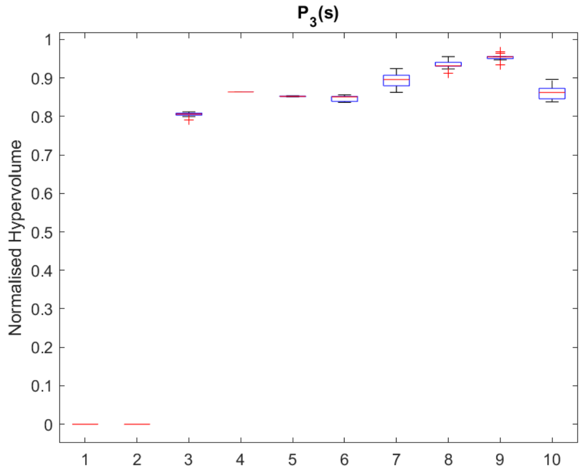

Fig. 5. Hypervolume $P_3(s)$.

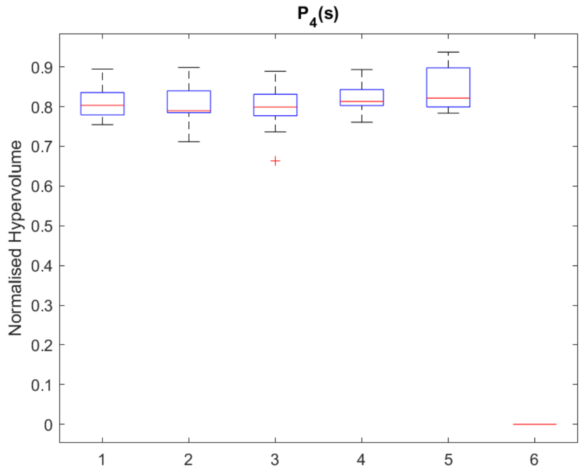

Fig. 6. Hypervolume $P_4(s)$.

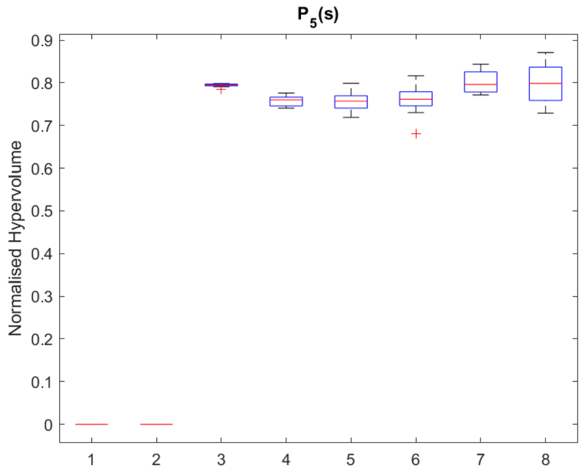

Fig. 7. Hypervolume $P_5(s)$.

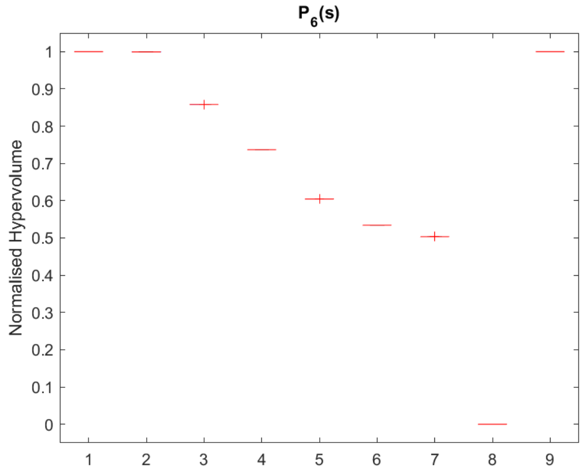

Fig. 8. Hypervolume $P_6(s)$.

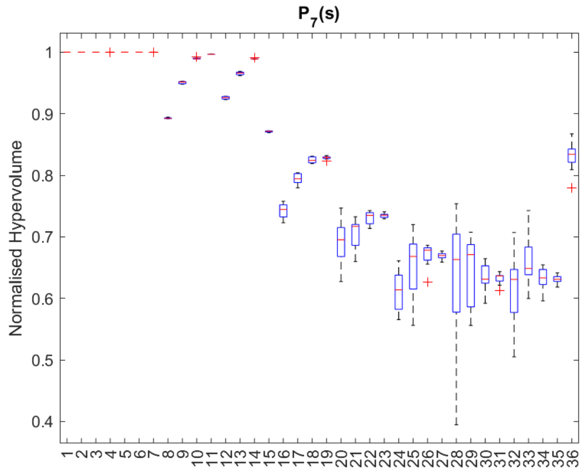

Fig. 9. Hypervolume $P_7(s)$.

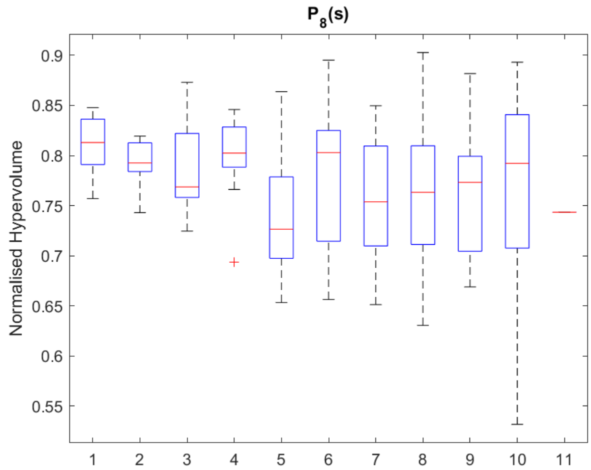

Fig. 10. Hypervolume $P_8(s)$.

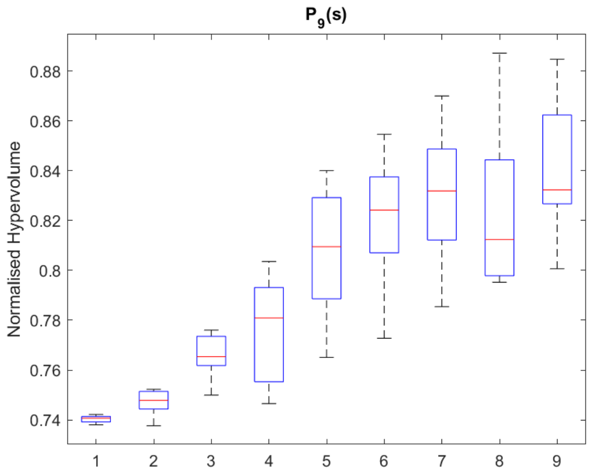

Fig. 11. Hypervolume $P_9(s)$.

Åström, K.J., Panagopoulos, H., and Hägglund, T. (1998). Design of pi controllers based on non-convex optimization. *Automatica*, 34(5), 585–601.

Atanasijevic-Kunc, M., Logar, V., Karba, R., Papic, M., and Kos, A. (2010). Remote multivariable control design using a competition game. *IEEE Transactions on Education*, 54(1), 97–103.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2012). Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425, 75–103.

Bejarano, G., Alfaya, J., Rodríguez, D., Ortega, M., and Morilla, F. (2017). Benchmark for PID control of Refrigeration Systems based on Vapour Compression. [Available at `http://servidor.dia.uned.es/~fmorilla/benchmarkPID2018/`].

Cao, Y., Acevedo, D., Nagy, Z.K., and Laird, C.D. (2017). Real-time feasible multi-objective optimization based nonlinear model predictive control of particle size and shape in a batch crystallization process. *Control Engineering Practice*, 69, 1–8.

Coello, C.A.C. (2003). Evolutionary multi-objective optimization: A critical review. *Evolutionary optimization*, 117–146.

Coello, C.A.C. and Lamont, G.B. (2004). *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific.

Dixon, R. and Pike, A.W. (2006). Alstom benchmark challenge II on gasifier control. *IEE Proceedings - Control Theory and Applications*, 153(3), 254–261.

Eriksson, L., Thomasson, A., Ekberg, K., Reig, A., Eifert, M., Donatantonio, F., Amato, A., Arsie, I., Pianese, C., Otta, P., et al. (2019). Look-ahead controls of heavy duty trucks on open roads-six benchmark solutions. *Control Engineering Practice*, 83, 45–66.

Fernández, I., Rodríguez, C., Guzman, J., and Berenguel, M. (2011). Control predictivo por desacoplo con compensación de perturbaciones para el benchmark de control 2009–2010. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(2), 112–121.

Garpinger, O., Hägglund, T., and Åström, K.J. (2014). Performance and robustness trade-offs in PID control. *Journal of Process Control*, 24(5), 568–577.

Kagami, R.M., Reynoso-Meza, G., Santos, E.A., and Freire, R.Z. (2019). Control of a refrigeration system benchmark problem: An approach based on cor metaheuristic algorithm and topsis method. *IFAC-PapersOnLine*, 52(11), 85–90.

Kroll, A. and Schulte, H. (2014). Benchmark problems for nonlinear system identification and control using soft computing methods: Need and overview. *Applied Soft Computing*, 25, 496–513.

Lim, I., Hoelzle, D.J., and Barton, K.L. (2017). A multi-objective iterative learning control approach for additive manufacturing applications. *Control Engineering Practice*, 64, 74–87.

Mercader, P., Cánovas, C., and Baños, A. (2019). Control PID multivariable de una caldera de vapor. *Revista Iberoamericana de Automática e Informática industrial*, 16(1), 15–25.

Messac, A., Ismail-Yahaya, A., and Mattson, C.A. (2003). The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25(2), 86–98.

Meza, G.R., Ferragud, X.B., Saez, J.S., and Durá, J.M.H. (2016). *Controller Tuning with Evolutionary Multiobjective Optimization: A Holistic Multiobjective Optimization Design Procedure*, volume 85. Springer.

Miettinen, K. (1999). Nonlinear multiobjective optimization, volume 12 of international series in operations research and management science.

Moarref, A.E., Sedighizadeh, M., and Esmaili, M. (2016). Multi-objective voltage and frequency regulation in autonomous microgrids using pareto-based big bang-big crunch algorithm. *Control Engineering Practice*, 55, 56–68.

Peitz, S. and Dellnitz, M. (2018). A survey of recent trends in multiobjective optimal control—surrogate models, feedback control and objective reduction. *Mathematical and Computational Applications*, 23(2), 30.

Reynoso-Meza, G., Blasco, X., Sanchis, J., and Martínez, M. (2014). Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice*, 28, 58–73.

Romero, J.A. and Sanchis, R. (2011). Benchmark para la evaluación de algoritmos de auto-ajuste de controladores PID. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(1), 112–117.

Xue, Y., Li, D., and Gao, F. (2010). Multi-objective optimization and selection for the pi control of alstom gasifier problem. *Control Engineering Practice*, 18(1), 67–76.