

A stabilizing PID controller sampling procedure for stochastic optimizers ^{*}

Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco,
Juan M. Herrero

*Instituto Universitario de Automática e Informática Industrial
Universidad Politécnica de Valencia, Valencia, España
(e-mail: gilreyme@upv.es, {jsanchis,xblasco,juaherdu}@isa.upv.es).*

Abstract: PID controller remain as a practical and reliable solution for a wide range of industrial applications. Efforts to develop new tuning techniques fulfilling several requirements and specifications are worthwhile. A commonly used approach is by means of an optimization procedure to adjust its parameters. Stochastic optimizers has been well used for PID controller tuning, due to their flexibility as global optimizers. Nevertheless, such algorithms calculate their tuning proposals in a stochastic manner, bringing two additional problems for the optimization statement; on the one hand, they should satisfy that any set of PID parameters will stabilize the closed loop; on the other hand, they should assure that all stabilizing PID controllers are consider by the algorithm. In this work, we shown a simple sampling mechanism which assure stabilizing PID controllers for FOPDT processes which tackle the aforementioned problems.

Keywords: PID auto-tuning, PID controllers, evolutionary algorithms, stabilizing PID stochastic sampling.

1. INTRODUCTION

PI-PID controller remains as a reliable and practical control solution for several industrial processes (Åström and Hägglund (2001); Åström (2002)). Owing to this, research for new tuning techniques is an ongoing research topic (Åström and Hägglund (2005)). Current research points to guarantee reasonable stability margins as well as a good overall performance for a wide variety of process. One of the main advantages of PI-PID controllers is their ease of implementation as well as their tuning, giving a good trade-off between simplicity and cost to implement (Tan et al. (2004); Stewart and Samad (2011)).

New tuning techniques are being focused on the fulfilment of several objectives and requirements, sometimes in conflict among them (Ang et al. (2005); Li et al. (2006)). Some tuning procedures are based on optimization statements (Ge et al. (2002); Toscano (2005); Goncalves et al. (2008); Åström et al. (1998); Panagopoulos et al. (2002)) and some cases they are solved by means of stochastic optimizers.

In general stochastic optimizers are characterized by the randomness used in the search process, helpful to avoid local minima. Evolutionary or nature inspired algorithms (for example) are very popular stochastic optimizers and they have been used extensively for PID-like controller tuning (Reynoso-Meza et al. (2013, Accepted)). Such algorithms use matching and mutation operators to induce such randomness into the search process. Nevertheless, in the case of PID controller tuning, this process has to be carefully performed; besides the (and often uncommented)

preliminary step of defining the boundaries in the search process, not all parameter combinations in a PID guarantees closed loop stability given a nominal process.

In this work, we show a simple sampling procedure for stochastic optimizers focused in first order plus dead time (FOPDT) processes, to generate randomly PID parameters which guarantee closed loop stability. This procedure not only assure sampling stabilizing PID parameters but also guarantees that all stabilizing controllers are minded. Such sampling could be used by any stochastic optimizer saving computational resources to achieve an optimal solution (given a cost function) improving its convergence. The remain of this work is as follows: in section 2 we present the sample procedure, whilst in section 3 we validate its usefulness with two different test cases. Finally, some concluding remarks are given.

2. DEVELOPMENT

Firstly, we will briefly explains how is characterized the subset of stabilizing PID controllers \mathbb{C} and afterwards, we will explain how to sample controller from such subset.

2.1 Computation of \mathbb{C}

We will consider in this work FOPDT transfer function $P(s)$ and the PID controller $C(s)$. Both are described by the equations:

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

$$C(s) = k_p + \frac{k_i}{s} + k_d s \quad (2)$$

where $k > 0$ is the process proportional gain, T the time constant and L the lag of the system; k_p, k_i, k_d the proportional, integral and derivative gains of the controller. According to Silva et al. (2002), the set $\mathbb{C} \subset \mathbb{R}^3$

^{*} This work was partially supported by the projects ENE2011-25900 and TIN2011-28082 from the Ministerio de Economía y Competitividad, Gobierno de España, the Universitat Politècnica de València by the project PAID-06-11 and the FPI-UPV 2010/19 grant and the Generalitat Valenciana (GV/2012/073).

of stabilizing controllers given the $P(s)$ process is given by subsets \mathbf{T} , $\mathbf{\Delta}$ and \mathbf{Q} (Figure 1).

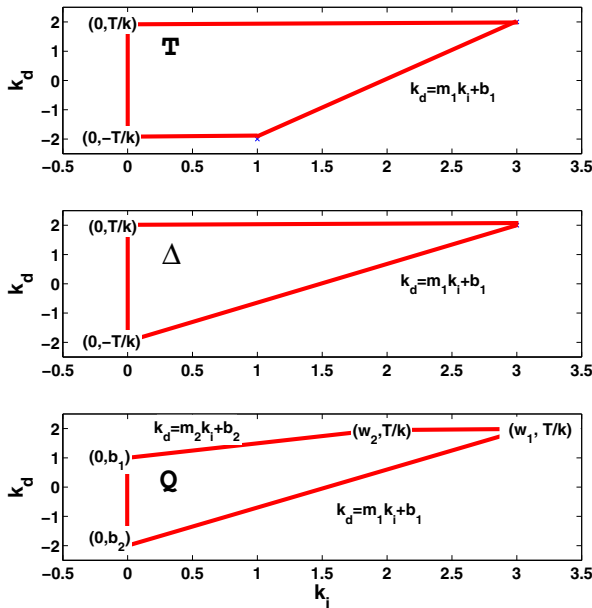


Fig. 1. PID stable regions.

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

$$-\frac{1}{k} < k_p < k_u \quad (3)$$

where k_u is usually known as the ultimate gain

$$k_u = \frac{1}{k} \left[\frac{T}{L} \alpha_1 \sin(\alpha_1) - \cos(\alpha_1) \right] \quad (4)$$

and α_1 is the solution of the equation

$$\tan(\alpha) = -\frac{T}{T+L} \alpha, \alpha \in [0, \pi] \quad (5)$$

The complete stabilization region is given by:

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

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

$$m_j = \frac{L^2}{z_j^2} \quad (6)$$

$$b_j = -\frac{L}{k \cdot z_j} \left[\sin(z_j) + \frac{T}{L} z_j \cos(z_j) \right] \quad (7)$$

$$w_j = \frac{z_j}{kL} \frac{(1 + k k_p)[1 + \cos(z_j)]}{\sin(z_j)} \quad (8)$$

with z_1, z_2 being the roots of

$$k \cdot k_p + \cos(z) - \frac{T}{L} z \sin(z) = 0 \quad (9)$$

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

$$\mathbb{C} = \mathbf{T} \bigcup \mathbf{\Delta} \bigcup \mathbf{Q} \quad (10)$$

2.2 Sampling on \mathbb{C}

We state that, given a subset $\mathbb{S} \in \mathfrak{R}^3$ and a process $P(s)$, two important features for PID $C(s)$ parameter sampling from \mathbb{S} should be fulfilled:

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

A common approach for feature 1 is to define bounds on the parameter which avoid all non-stable but also some stable PID parameters; therefore, feature 2 is not fulfilled. A second alternative, is to bound the search space with all stable PID parameters, but including non-stable parameters, which are verified whilst the algorithm is running. This obviously doesn't fulfill feature 1, and could misspent computational (CPU time) resources.

It is necessary sampling controllers from \mathbb{C} in order to fulfill both features. We assume that controllers matching $k_p > 0, k_i > 0, k_d > 0$ are the only accepted by the designer to stabilize $P(s)$. According to this, we define an injective function $\mathfrak{R}^3 \rightarrow \mathfrak{R}^3$ to map from the unitary cube to \mathbb{C} :

$$F(a, b, c) = [k_p, k_i, k_d] \quad (11)$$

$$\{a, b, c\} \in [0, 1] \quad (12)$$

Where:

$$k_p = a \cdot k_u \quad (13)$$

$$k_d = c \cdot \frac{T}{K} \quad (14)$$

$$k_i = \begin{cases} b \cdot \frac{k_d - b_1}{m_1} & \text{if } k_p \leq 1/k \\ \frac{k_d - b_2}{m_2} + b \cdot \left[\frac{k_d - b_1}{m_1} - \frac{k_d - b_2}{m_2} \right] & \text{if } 1/k < k_p \end{cases} \quad (15)$$

Therefore, the algorithm to generate stabilizing controllers is:

Algorithm I: PID.Sampling

- 1 : Read values k, T, L
- 2 : Calculate k_u (equation 4)
- 3 : Read values a, b, c
- 4 : Calculate k_p (equation 13)
- 5 : Calculate k_d (equation 14)
- 6 : Determine roots $z_j, j = [1, 2]$ (equation 9)
- 7 : Determine values $m_j, b_j, j = [1, 2]$ (equations 6,7)
- 8 : Determine k_i (equation 15)
- 9 : Return k_p, k_i, k_d
- 10 : Return $T_i = k_p/k_i, T_d = k_d/k_p$

Simple analytic solutions using Newton-Raphson optimization method could be employed. Line 1 and 2 could be calculated *off-line* if a fixed FOPDT is under consideration. If not required, line 10 could be omitted. In figure 2 a simple random sampling using Algorithm I is shown. Next, we will use this approach for several stochastic optimizers to evaluate its performance.

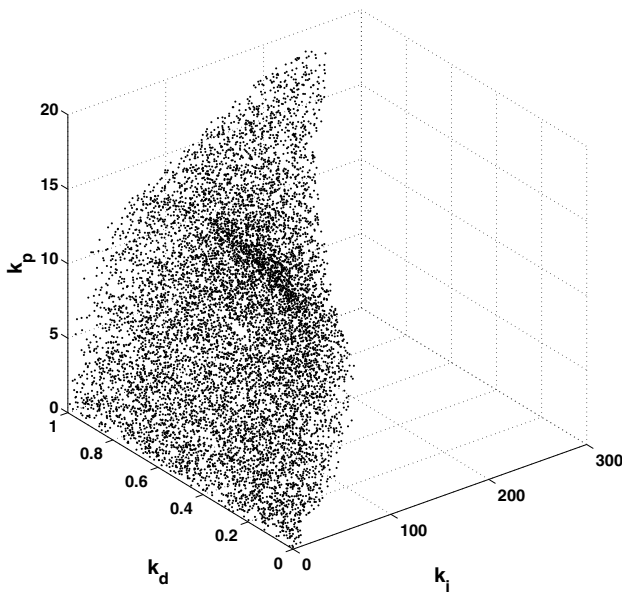


Fig. 2. Stabilizing controllers for system $\frac{1}{s+1}e^{-0.1s}$ (1e6 samples are plotted) from stochastic sampling in the unitary cube.

3. EVALUATION

We will use two different stochastic optimizers to evaluate the performance of the sampling. In this work, we will focus on single objective optimization, due to limitation of space. The optimizers under consideration are:

DE: Differential Evolution algorithm (Das and Suganthan (2010); Storn (2008); Storn and Price (1997)). It belongs to the family of evolutionary strategies, and it is a very popular and compact algorithm. It is used a version available from Matlab central.¹

SA: Simulated annealing algorithm (Suman and Kumar (2005)). A heuristic optimizer emulating annealing process in materials for search in the decision space. It is used the version available in the Matlab optimization toolbox.

CMAES: Covariance Matrix Adaptation Evolution Strategy (Hansen (2006)). An algorithm from the family of estimation of distribution algorithms. It is used the version available from the authors for Matlab.²

In all cases, each algorithm has been downloaded from its respective source and used with its standard parameters. That is, no further effort on adjusting algorithm's parameters has been done.

The process under consideration is the benchmark for PID control 2012 described by Morilla (2012). It is a benchmark which proposes a boiler control problem (Morilla (2010); Fernández et al. (2011)) based on the work of Pellegrinetti and Bentsman (1996). The original problem stated a 3x3 MIMO system with a measured load disturbance:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{bmatrix} = \begin{bmatrix} g_{11}(s) & 0 & g_{13}(s) \\ g_{21}(s) & g_{22}(s) & 0 \\ g_{31}(s) & 0 & g_{33}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \\ U_3(s) \end{bmatrix} + \begin{bmatrix} g_{1d}(s) \\ 0 \\ g_{3d}(s) \end{bmatrix} D(s) \quad (16)$$

Where the inputs are fuel flow, water flow and air flow whilst the outputs are steam pressure, oxygen level and water level. This is a verified model, useful to propose, evaluate and compare different kinds of tuning/control techniques (Garrido et al. (2012); Saeki et al. (2012); Ochi (2012); Silveira et al. (2012); Rojas et al. (2012)).

For this benchmark a reduced MIMO and SISO version were available to evaluate different controller tuning procedures. In this work, we focus in the SISO PID controller tuning (Figure 3), where the fuel flow is manipulated to control the steam pressure.

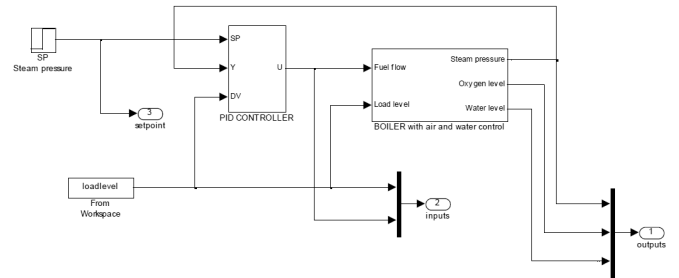


Fig. 3. SISO loop (taken from Morilla (2012)).

The benchmark also defined an index $I_{benchmark}$ to evaluate the performance for a given controller. It is an aggregate objective function, which combines ratios of IAE, ITAE and IAVU indexes using a base case PI controller ($[k_p, T_i] = [2.5, 50]$). More details can be consulted in Morilla (2012). We use the benchmark index to tune a PID controller using an identified model $G(s)$ ³ to be evaluated later in the original model.

$$G(s) = \frac{0.3934}{45.6794s + 1} e^{-3.42s} \quad (17)$$

3.1 Performance test 1

The aim of the first test design is to validate the hypothesis that the stochastic sampling will improve each algorithm performance to calculate optimal PID parameters⁴. For this purpose, a total of 2000 function evaluations have been used for each algorithm and 51 independent runs are carried out. Each one will be executed twice, in order to test the sampling proposal:

- (1) An execution using standard bounds on k_p , T_i , T_d .
- (2) An execution using the stochastic sampling previously explained.

Therefore, the following optimization problem is defined for the former instance:

¹ <http://www.mathworks.com/matlabcentral/fileexchange/38962>

² <https://www.lri.fr/~hansen/cmaesintro.html>

³ This model was obtained with a step response experiment using the identification toolbox of Matlab©.

⁴ Optimal according to the selected cost function $I_{benchmark}$.

$$\min J(k_p, T_i, T_d) = I_{benchmark}|_{G(p(s))} \quad (18)$$

s.t.

$$k_p \in (0, k_u|_{G(s)}) \quad (19)$$

$$T_i \in [0, 100] \quad (20)$$

$$T_d \in [0, 100] \quad (21)$$

$$Re[\lambda] < 0 \quad (22)$$

whilst for the latter:

$$\min J(a, b, c) = I_{benchmark}|_{G(p(s))} \quad (23)$$

s.t.

$$a \in (0, 1] \quad (24)$$

$$b \in [0, 1] \quad (25)$$

$$c \in [0, 1] \quad (26)$$

In the former case, a basic penalty function is used to identify feasible from non-feasible solutions. The results are shown in Tables 1, 2 and 3. In the case of the DE algorithm (Table 1), there are not statistical differences according to the Wilcoxon test at 95%, although it can be notice that DE algorithm without the coding is not capable of finding a suitable controller in all executions. Regarding SA and CMA-ES algorithms (Tables 2 and 3) the Wilcoxon test reveals that the instance using the sampling proposed leads to better results on both cases. In each instance, algorithms seems to be converging to a PID controller with parameters $[k_p, T_i, T_d] = [3.99, 29.41, 0]$. That is, a controller without derivative gain. The performance of such controller and its comparison with the reference case (using files and guidelines from the benchmark) are depicted in Figure 4.

Table 1. Performance of DE algorithm in test 1.

| | Stochastic Sampling | Standard bounds |
|----------|---------------------|-----------------|
| minimum | 0.6247 | 0.6247 |
| median | 0.6448 | 0.6480 |
| maximum | 22.0645 | 56.4709 |
| mean | 3.4526 | 6.7177 |
| variance | 37.4191 | 167.2447 |
| success | 51 | 47 |

Table 2. Performance of SA algorithm in test 1.

| | Stochastic Sampling | Standard bounds |
|----------|---------------------|-----------------|
| minimum | 0.6247 | 0.6255 |
| median | 0.6248 | 0.6943 |
| maximum | 0.6847 | 0.9726 |
| mean | 0.6265 | 0.7415 |
| variance | 0.0002 | 0.0350 |
| success | 51 | 51 |

3.2 Performance test 2

The aim of the second test design is to validate

- The efficacy of the proposal, due to a more efficient number of function evaluations required.
- The usefulness on this sampling for multidisciplinary optimization design.

It will be tuned a PI for the aforementioned process. Nevertheless, it will be assumed a design phase for the

Table 3. Performance of CMA-ES algorithm in test 1.

| | Stochastic Sampling | Standard bounds |
|----------|---------------------|-----------------|
| minimum | 0.6247 | 0.6247 |
| median | 0.6247 | 0.6255 |
| maximum | 14.9956 | 18.3715 |
| mean | 0.9166 | 1.3141 |
| variance | 4.0489 | 11.8571 |
| success | 51 | 51 |

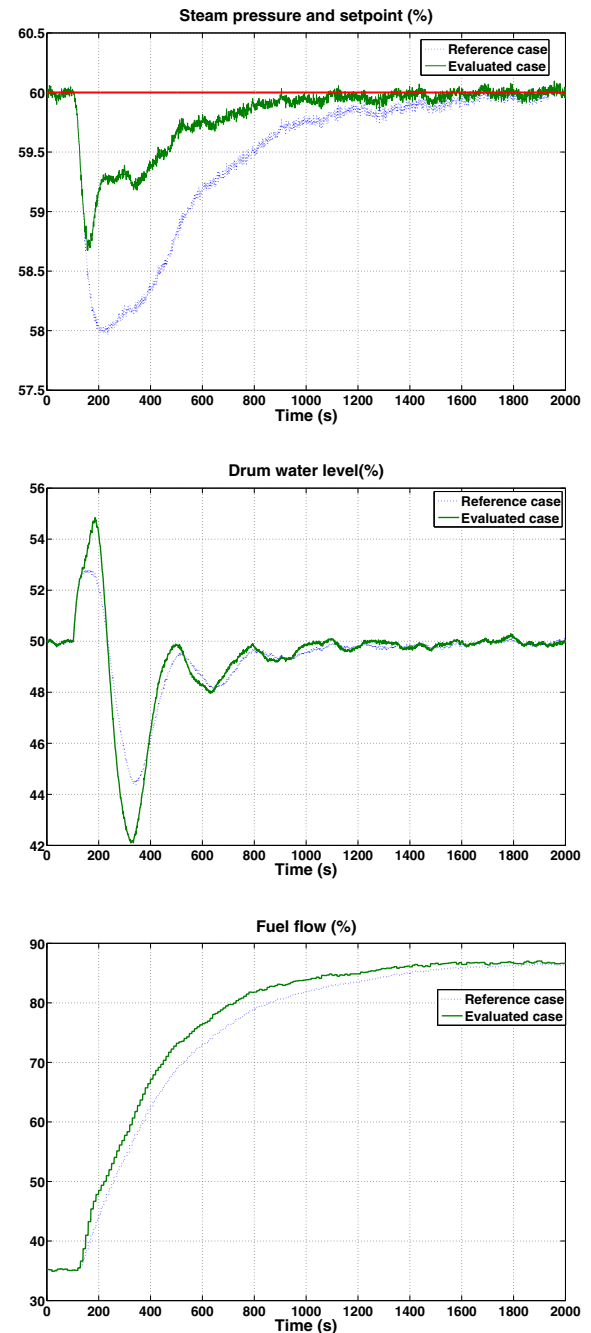


Fig. 4. Performance of the PI controller $[k_p, T_i] = [3.99, 29.41]$ and its comparison with the reference case $[k_p, T_i] = [2.5, 50]$.

boiler, where it is decided to carry a multidisciplinary design approach of plant design and control. As detailed in Roy et al. (2008) and Chai et al. (2013), multidisciplinary optimization approaches could bring new and interesting solutions for designer, since it shows a better performance than sequential design procedures. This approach has been used before with success involving PI and PID controllers (Avigad et al. (2003); Behbahani and de Silva (2008); Jiachuan et al. (2005); Lamanna et al. (2009)).

Lets assume a hypothetic case where from the nominal model identified, the plant could be (re)designed as:

$$G'(s) = f(l) = \frac{0.3934 \cdot (4 - 3l)}{45.6794 \cdot (4 - 3l)s + 1} e^{-3.42s} \quad (27)$$

with free parameter $l \in [0.7, 1.3]$. In a physical sense, with a fixed step reference change it means that:

- it is possible to (re)design a plant with quicker response (i.e. with a lower time constant) but at expense of more energy required to get reference (i.e. a lower gain).
- it is possible to (re)design a plant with a lower response (i.e. with a high time constant) that requires less energy to get reference (i.e. a bigger gain).

Only CMA-ES algorithm will be used. The termination condition will be defined by the algorithm itself. Therefore, the following optimization problems are defined:

$$\min J(k_p, T_i, l) = I_{benchmark}|_{G'(p(s)=f(l))} \quad (28)$$

s.t.

$$k_p \in \left(0, \max(k_u)|_{G'(s)=f(l)}\right) \quad (29)$$

$$T_i \in [0, 100] \quad (30)$$

$$l \in [0.7, 1.3] \quad (31)$$

$$Re[\lambda] < 0 \quad (32)$$

and

$$\min J(a, b, l) = I_{benchmark}|_{G'(p(s)=f(l))} \quad (33)$$

s.t.

$$a \in (0, 1] \quad (34)$$

$$b \in [0, 1] \quad (35)$$

$$l \in [0.7, 1.3] \quad (36)$$

Again, in the former case, a basic penalty function is used to identify stabilizing from non-stabilizing solutions. In Tables 4 and 5 the results are shown. Whilst there are not statistical differences (according Wilcoxon test) in the value achieved in the benchmark index $I_{benchmark}$, there are differences in the function evaluations used. Therefore, the sampling proposal has been useful to reduce the quantity of function evaluations required to get an optimal value.

Table 4. Performance of the CMA-ES algorithm in test 2.

| | Stochastic Sampling | Standard bounds |
|----------|---------------------|-----------------|
| minimum | 0.5855 | 0.5855 |
| median | 0.5855 | 0.5855 |
| maximum | 0.5855 | 0.5855 |
| mean | 0.5855 | 0.5855 |
| variance | 1.8e-27 | 2.8e-14 |
| success | 51 | 51 |

Table 5. Function evaluation required by CMA-ES algorithm in test 2.

| | Instance 1 | Instance 2 |
|----------|--------------|------------|
| minimum | 1815 | 2137 |
| median | 2417 | 3229 |
| maximum | 3537 | 4447 |
| mean | 2415 | 3146 |
| variance | 1.0e5 | 2.3e5 |

4. CONCLUSIONS AND FUTURE WORK

In this work, we have presented a simple coding to sample internal stable PID controllers given a FOPDT process. This coding could be potentially used by stochastic optimizers, to improve their efficiency in the global search procedure. In summary, the main advantages of this proposal are:

- Less function evaluations are used, since the algorithm is always sampling candidate solutions in a space where all controllers stabilize the closed loop.
- All the stabilizing controllers are contained in the set; therefore, *a priori*, the algorithm is minding all the possible tuning configurations.
- No guesses or hints are made about the decision search space for the evolutionary algorithm; the search space is straightforward bounded.
- It is an approach suitable for multidisciplinary optimization, where system itself could be subject to (re)design.

Whilst simple examples have been shown in this paper (due to limitation of space), the approach is suitable to face more demanding optimizations statements by including constraints. Limitations and future work are:

- The code is presented just for FOPDT processes; future work should focus on extend the sampling to process of higher complexity.
- Only PI and PID controllers can be tuning with this procedure; it should be extended to PID controllers with derivative filter and setpoint weighting.
- It is limited to SISO process; it will be interesting to use the same approach in MIMO statements.

REFERENCES

- Ang, K.H., Chong, G., and Li, Y. (2005). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*.
- Åström, K.J. (2002). *Control system design*.
- Åström, K.J. and Hägglund, T. (2005). *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709.
- Åström, K. and Hägglund, T. (2001). The future of PID control. *Control Engineering Practice*, 9(11), 1163 – 1175. doi:10.1016/S0967-0661(01)00062-4.
- Åström, K., Panagopoulos, H., and Hägglund, T. (1998). Design of PI controllers based on non-convex optimization. *Automatica*, 34(5), 585 – 601. doi:DOI:10.1016/S0005-1098(98)00011-9.
- Avigad, G., Moshaiov, A., and Brauner, N. (2003). Towards a general tool for mechatronic design. In *Control Applications, 2003. CCA 2003. Proceedings of 2003 IEEE Conference on*.
- Behbahani, S. and de Silva, C. (2008). System-based and concurrent design of a smart mechatronic system using

- the concept of mechatronic design quotient (MDQ). *IEEE/ASME Transactions on Mechatronics*.
- Chai, T., Jin, Y., and Sendhoff, B. (2013). Evolutionary complex engineering optimization: opportunities and challenges. *Computational Intelligence Magazine*, 8(3), 12 – 15.
- Das, S. and Suganthan, P.N. (2010). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, PP(99), 1 –28. doi:10.1109/TEVC.2010.2059031.
- 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*, Apr. 8(2), 112 – 121.
- Garrido, J., Márquez, F., and Morilla, F. (2012). Multi-variable PID control by inverted decoupling: application to the benchmark PID 2012. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)*.
- Ge, M., Chiu, M.S., and Wang, Q.G. (2002). Robust PID controller design via LMI approach. *Journal of process control*, (12), 3 – 13.
- Goncalves, E.N., Palhares, R.M., and Takahashi, R.H. (2008). A novel approach for H_2/H_∞ robust PID synthesis for uncertain systems. *Journal of process control*, (18), 19 – 26.
- Hansen, N. (2006). The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea (eds.), *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, 75 – 102. Springer.
- Jiachuan, W., Zhun, F., Terpenney, J., and Goodman, E. (2005). Knowledge interaction with genetic programming in mechatronic systems design using bond graphs. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*.
- Lamanna, R., Vega, P., Revollar, S., and Alvarez, H. (2009). Diseño simultáneo de proceso y control de una torre sulfatadora de jugo de caña de azúcar. *Revista Iberoamericana de Automática e Informática Industrial*, 6(3), 32 – 43.
- Li, Y., Ang, K.H., and Chong, G. (2006). PID control system analysis and design. *IEEE Control Systems*.
- Morilla, F. (2010). Benchmark 2009-10 grupo temático de ingeniería de control de cea-ifac: Control de una caldera. Available at www.cea-ifac.es/w3grupos/ingcontrol.
- 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.
- Ochi, Y. (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)*.
- Panagopoulos, H., Åström, K., and Hägglund, T. (2002). Design of PID controllers based on constrained optimisation. *Control Theory and Applications, IEE Proceedings -*
- Pellegrinetti, G. and Bentsman, J. (1996). Nonlinear control oriented boiler modeling-a benchmark problem for controller design. *IEEE Transactions on Control Systems Technology*.
- Reynoso-Meza, G., Blasco, X., Sanchis, J., and 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.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., and Martínez, M. (Accepted). Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice*, (DOI: 10.1016/j.conengprac.2014.03.003).
- Rojas, J.D., Morilla, F., and Vilanova, R. (2012). Multi-variable 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)*.
- Roy, R., Hinduja, S., and Teti, R. (2008). Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals - Manufacturing Technology*, 57(2), 697 – 715. doi:10.1016/j.cirp.2008.09.007.
- Saeki, M., Ogawa, K., and Wada, N. (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)*.
- Silva, G., Datta, A., and Bhattacharyya, S. (2002). New results on the synthesis of pid controllers. *Automatic Control, IEEE Transactions on*, 47(2), 241–252. doi: 10.1109/9.983352.
- Silveira, A., Coelho, A., and Gomes, F. (2012). Model-free adaptive PID controllers applied to the benchmark PID12. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)*.
- Stewart, G. and Samad, T. (2011). Cross-application perspectives: Application and market requirements. In T. Samad and A. Annaswamy (eds.), *The Impact of Control Technology*, 95 – 100. IEEE Control Systems Society.
- Storn, R. (2008). Sci: Differential evolution research: Trends and open questions. volume LNCS 143, 1 – 31. Springer, Heidelberg.
- Storn, R. and Price, K. (1997). Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341 – 359.
- Suman, B. and Kumar, P. (2005). A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, 57(10), 1143–1160.
- Tan, W., Liu, J., Fang, F., and Chen, Y. (2004). Tuning of PID controllers for boiler-turbine units. *ISA Transactions*, 43(4), 571 – 583. doi:10.1016/S0019-0578(07)60169-4.
- Toscano, R. (2005). A simple robust PI/PID controller design via numerical optimization approach. *Journal of process control*, (15), 81 – 88.