Chapter 3

Optimum PI/PID Controllers Tuning via an Evolutionary Algorithm

Jorge-Humberto Urrea-Quintero,
Jesús-Antonio Hernández-Riveros and
Nicolás Muñoz-Galeano

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.74297

Abstract

In this chapter, it is demonstrated that when using advanced evolutionary algorithms, whatever the adopted system model (SOSPD, nonminimum phase, oscillatory or nonlinear), it is possible to find optimal parameters for PID controllers satisfying simultaneously the behavior of the system and a performance index such as absolute integral error (IAE). The Multidynamics Algorithm for Global Optimization (MAGO) is used to solve the control problem with PID controllers. MAGO is an evolutionary algorithm without parameters, with statistical operators, and for the optimization, it does not need the derivatives, what makes it very effective for complex engineering problems. A selection of some representative benchmark systems is carried out, and the respectively two-degree-of-freedom (2DoF) PID controllers are tuned. A power electronic converter is adopted as a case study and based on its nonlinear dynamical model, a PI controller is tuned. In all cases, the control problem is formulated as a constrained optimization problem and solved using MAGO. The results found are outstanding.

Keywords: evolutionary algorithms, PID controller, nonlinear model, MAGO, 2DoF PID-based controllers, SOSPD model, control benchmark, power electronic converters

1. Introduction

It is well known that most of 90% of the closed-loop implemented strategies are PI or PID controllers [1]. Since its introduction in 1940, researchers' interest has been focused on the development on simplistic but effective tuning rules for PID controllers [2]. For any industrial plant without resonant characteristics, a SOSPD model can represent the dominant dynamics

and that a suitable PID tuning can be achieved [3]. Moreover, Åström and Hägglund proposed a collection of systems models with various difficulties of control that are suitable for testing PID controllers. Those models illustrate systems with various difficulties of control. Nevertheless, SOSPD representation is limited, and a wide range of process dynamics can be found, for example, process with a non-minimum phase behavior or oscillations. Therefore, PID control traditional tuning is not well suited for most of the complex problems. The trend remains in the so-called two degree-of-freedom (2DoF) PID controllers [4]. The 2DoF PID control structure has two components: one to tune the controller considering the regulatory closed-loop mode performance and robustness and the second one to improve the servo-control behavior. 2DoF PID tuning could be also based on a system transfer function. A 2DoF PID control structure is the option to achieve simultaneously a good system performance as both regulator and servomechanism modes what is a challenge of control requirements for a traditional PID controller.

Evolutionary algorithms (EA) have been proved to be an effective tool to optimal PID controllers tuning [5]. In general, EA is considered as an optimal algorithm that is able to deal with ill-defined problem domain such as multimodality, discontinuity, time variance, randomness and noise [6]. MAGO as EA does not work with genetic operators. MAGO operators are inspired in numerical derivation applying the Nelder-Mead method, the estimation distribution of the actual population and a statistical quality control technique. Additionally, MAGO has only two tunable parameters: the population size and the number of generations. These two parameters could be removed, but they remain because in real situations they help to understand the context of the problem. Particularly, MAGO has been successfully tested for the tuning of PID controllers based on SOSPD models [7]. MAGO has been used in various fields of engineering [8], LQR tuning [9–11], drivers in tuning PID controllers [12, 13], showing successful solutions in each case applied.

Despite a lot of works in PID controllers tuning, a general concern remains because real processes have multiple operational constraints, and some exhibit high nonlinear dynamics that cannot successfully be captured by transfer functions. This chapter shows that advanced evolutionary algorithms are suitable to solve the control problem with PID controllers when the control system is formulated as an optimization problem. The evolutionary algorithm MAGO is used to solve the control problem with traditional PI and with 2DoF PID controllers. MAGO is an evolutionary algorithm without parameters; it is based on statistical operators and does not need the derivatives of the nonlinear optimization problems. Furthermore, this chapter demonstrates that whatever system model is adopted (SOSPD, non-minimum phase, oscillatory or nonlinear), it is possible to find optimal parameters for PID controllers satisfying the system behavior and a performance index such as absolute integral error (IAE). This chapter is divided into four sections as follows: in Section 2, the general problem statement of PID controllers tuning is introduced and formulated as an optimization problem. In Section 3, the (EA) MAGO is presented and the evolutionary design procedure of a PID controller is established. In Section 4, a selection of some representative benchmark systems from [5] is carried out, and the respectively 2DoF PID controllers are tuned. In Section 5, a power electronic converter (DC-DC buck converter) is adopted as a case study, and based on its nonlinear

dynamical model, a PI controller is tuned by MAGO. PI-MAGO controller performance is tested, and a comparison is carried out against a PID controller tuned by the pole placement method. In all cases, the control problem is formulated as a constrained optimization problem and solved using MAGO.

2. Problem statement

2.1. Control system representation

Consider a single-input single-output (SISO) control system depicted in **Figure 1**. In this system, $r(t)$ is the set-point, $u(t)$ is the controller output signal, $d(t)$ is the load-disturbance, and $y(t)$ is the controller process variable.

The system output $y(t)$ simultaneously depends on $r(t)$ and $d(t)$. Two operation modes should be taken into account for the controller design, one as a servomechanism and the other as a regulator. In the first case, the control objective is to track the set-point $r(t)$. In the second case, the control objective is to reject a change in $d(t)$ while $y(t)$ is keeping as close as possible to $r(t)$. However, it is not always possible to specify distinctly performance criteria for both operation modes. Furthermore, a trade-off between servo-regulator modes must be specified as in the traditional PID controller tuning case.

The general form to represent a dynamical system is given by Eqs. (1) and (2), where x is the system state and $\dot{x}$ is the time derivative of the system state; $y(t)$ is the system output and functions g and h are nonlinear and represent the dynamical system evolution.

$$\dot{x}(t) = g(x, u, d) \tag{1}$$

$$y(t) = h(x, u, d) \tag{2}$$

Through Taylor linearization, it is possible to obtain a linear representation of Eqs. (1) and (2) given by Eqs. (3) and (4). Where $A \in \mathcal{R}^{nxn}$ is the system Jacobian, $B \in \mathcal{R}^{nxm}$ is an input matrix, $C \in \mathcal{R}^{kxn}$ is an output matrix and $D \in \mathcal{R}^{kxm}$ is a direct transmission matrix.

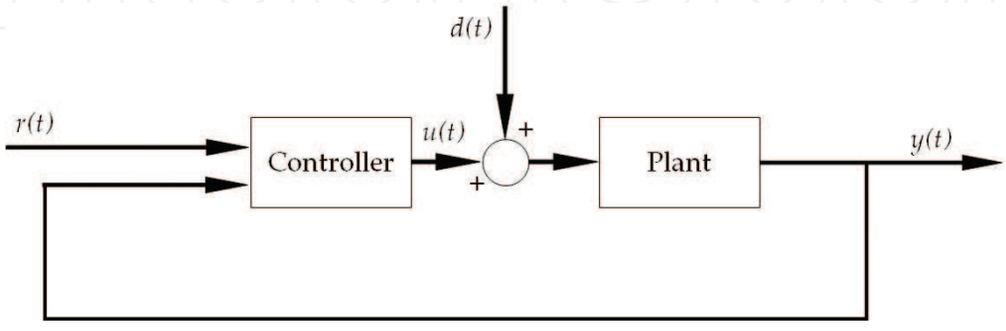

Figure 1. Single-input single-output (SISO) feedback control system.

$$\dot{x} = Ax + Bu \tag{3}$$

$$y = Cx + Du \tag{4}$$

Eqs. (3) and (4) are usually employed in control theory for designing multi-input multi-output (MIMO) control structures. However, transfer functions are a better approximation for designing SISO control structures. The realization given by Eq. (5) is obtained by applying the Laplace transform to Eqs. (3) and (4).

$$G(s) = C\left[\frac{adj(sI - A)^T}{\det(sI - A)}\right]B + D \tag{5}$$

On the other hand, the control policy of an ideal PID controller is expressed by Eq. (6). Where, $e(t) = r(t) - y(t)$, K_p is the proportional gain, T_i is the integral time constant and T_d is the derivative time constant.

$$u(t) = K_p\left\{e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right\} \tag{6}$$

PID frequency domain representation is given by Eq. (7).

$$C(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{7}$$

The closed-loop transfer function form for the system represented in **Figure 1** considering a PID controller is given by Eq. (8).

$$Y(s) = \frac{C(s)G(s)}{1 + C(s)G(s)}R(s) + \frac{G(s)}{1 + C(s)G(s)}D(s) \tag{8}$$

If the system operates in the servomechanism mode, that is when disturbances are not considered, the output signal can be represented as in Eq. (9).

$$Y_{sp}(s) = \frac{C(s)G(s)}{1 + C(s)G(s)}R(s) \tag{9}$$

If the system operates in the regulation mode, that is when the signal reference is not considered, the output signal can be represented as in Eq. (10).

$$Y_{ld}(s) = \frac{G(s)}{1 + C(s)G(s)}D(s) \tag{10}$$

From Eqs. (9) and (10), both control objectives cannot be optimally achieved because the controller parameters simultaneously affect the servo and regulatory operation modes. At most, a tuning PID controller process can be carried out by establishing a servo-regulatory trade-off to obtain a closed-loop performance that is not optimum for neither servo nor regulatory operation modes, but it has an acceptable performance in both cases.

Controllers of the form given by either Eqs. (6) or (7) are known as one degree of freedom (1DoF) PID controllers. Two degree of freedom (2DoF) PID controllers are an alternative to overcome the 1DoF PID controller operation limitations. The control policy for a 2DoF PID controller is given by Eqs. (11) or (12). Considering the proportional, integral and derivative error, respectively in Eqs. (13), (14) and (15), where K_p is the proportional gain, T_i is the integral time constant and T_d is the derivative time constant, β and γ are the set-point weights. In **Figure 2**, the 2DoF PID controller block diagram is depicted.

$$u(t) = K_p\left\{e_p(t) + \frac{1}{T_i}\int_0^t e_i(\varepsilon)d\varepsilon + T_d\frac{de_d}{dt}\right\} \tag{11}$$

$$C_{2DoF}(s) = K_p\left\{e_p(s) + \frac{1}{T_i s}e_i(s) + T_d e_d(s)\right\} \tag{12}$$

with

$$e_p(s) = \beta R(\mathrm{s}) - \mathrm{Y}(\mathrm{s}) \tag{13}$$

$$e_i(s) = R(s) - Y(s) \tag{14}$$

$$e_d(s) = \gamma R(s) - Y(s) \tag{15}$$

The parameter γ is more frequently applied as a derivative mode switch (0 or 1) for $R(s)$. γ is normally set to zero to avoid an extreme instantaneous change in the controller output when a set-point step change occurs. In consequence, Eq. (12) can be arranged as in Eq. (16).

$$C_{2DoF}(s) = K_p\left(\beta + \frac{1}{T_i s}\right)R(S) - K_p\left(1 + \frac{1}{T_i s} + T_d s\right)Y(s) \tag{16}$$

A compact form of Eq. (16) consistent with **Figure 2** is given by Eq. (17). Where $C_r(s)$, is the set-point controller transfer function and $C_y(s)$ is the feedback controller transfer function.

$$C_{2DoF}(s) = C_r(s)R(s) - C_y(s)Y(s) \tag{17}$$

The closed-loop transfer function form for the system represented in **Figure 2** considering a 2DoF PID controller is given by Eq. (18). Where $M_{yr}(s)$ is the transfer function from the set

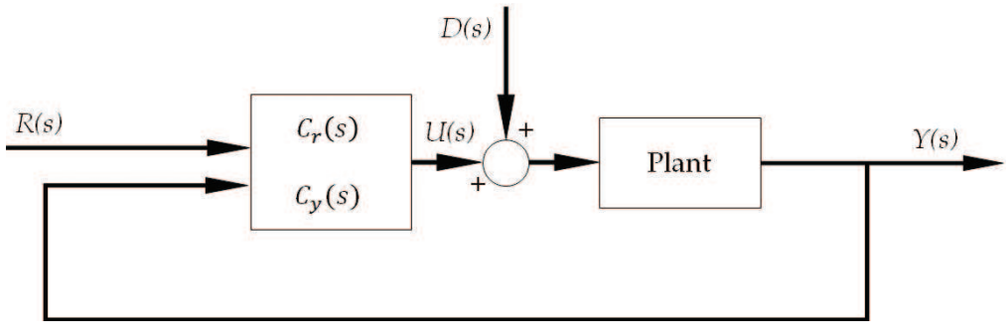

Figure 2. 2DoF closed-loop block diagram.

point to the controlled output (servo-control closed-loop), and $M_{yd}(s)$ is the transfer function from load-disturbance to the controller output (regulatory control closed-loop).

$$Y(s) = M_{yr}(s)R(s) + M_{yd}D(s) \tag{18}$$

With

$$M_{yr}(s) = \frac{C_r(s)G(s)}{1 + C_y(s)G(s)} \tag{19}$$

$$\mathrm{M_{yd}} = \frac{\mathrm{G(s)}}{1 + \mathrm{C_y(s)G(s)}} \tag{20}$$

From Eqs. (19) and (20), both control objectives can be achieved separately because of the possibility of tuning two different controllers, one for each operation mode.

2.2. Optimum control problem formulation

The usual criterion for tuning a controller is directly related to the desired closed-loop system response. Integral performance indexes allow quantifying the closed-loop system performance due to a unit step load disturbance. Most common employed indexes are integral of absolute error (IAE) (see Eq. (21)) and integral of absolute control action $u(t)$ (IAU) (see Eq. (22)).

$$IAE = \int_0^\infty |e(t)|dt \tag{21}$$

$$IAU = \int_0^\infty |e_u(t)|dt \tag{22}$$

For the PID case, $e(t)$ and $e_u(t)$, Eqs. (21) and (22), respectively, can be calculated as $e(t) = r(t) - y(t, \theta)$ and $e_u(t) = u_0 - u(t, \theta)$, where u_0 is the initial condition $\theta = \begin{bmatrix} K_p & T_i & T_d \end{bmatrix}$.

The load disturbance may enter at many different places, and extreme cases occur when it enters at the process input or output. However, when a feedback error appears, integral performance indexes evaluate the controller performance indistinctly, whereas load disturbance appears [14]. The abovementioned fact is the main motivation to adopt integral performance indexes.

The integrated error for a unit step disturbance at the process input is the inverse of the controller integral gain, $IE = 1/k_i$. For a unit step output disturbance, it is instead, $IE = 1/Kk_i$, where K is the static gain of the process. When the closed-loop system is well-damped, $IE \approx IAE$ are approximately the same [1]. The criteria IE and IAE are widely employed to measure controllers performance. IAE and IAU performance indexes are initially adopted in this work because of its interpretability from the PID controller parameters, but other criteria could also been used.

The controller tuning process that minimizes an integral performance index can be seen as an optimization problem where the ultimate goal is to find a controller parameters combination such that the value of an Integral Performance Index is minimized.

For the 1DoF PID controller, the optimal tuning problem consists of minimizing the objective function given by Eq. (23), where the minimum is the result of obtaining a suitable combination of the 1DoF PID parameters $\theta = \begin{bmatrix} K_p & T_i & T_d \end{bmatrix}$.

$$\text{Min}_\theta \left\{ J_{IAE} = \sum_{t_k}^{t_{kf}} |e(t_k)| \right\} \tag{23}$$

Subject to a process model (see Eqs. (1)–(5)), a control action (see Eq. 6) and

$$K_{p_{min}} \leq K_p \leq K_{p_{max}}$$

$$T_{i_{min}} \leq T_i \leq T_{i_{max}}$$

$$T_{d_{min}} \leq T_d \leq T_{d_{max}}$$

Similarly, for the 2DoF PID controller, the optimal problem consists of minimizing the objective function given by Eq. (24), where the minimum is the result of obtaining a suitable combination of the 2DoF PID parameters $\theta = \begin{bmatrix} K_p & T_i & T_d & \beta \end{bmatrix}$.

$$\text{Min}_\theta \left\{ J_{IAE+IAU} = \sum_{t_k}^{t_{kf}} |e(t_k)| + \sum_{t_k}^{t_{kf}} |e_u(t_k)| \right\} \tag{24}$$

Subject to a process model (see Eqs. (1)–(5)), a control action (see Eq. 11) and

$$K_{p_{min}} \leq K_p \leq K_{p_{max}}$$

$$T_{i_{min}} \leq T_i \leq T_{i_{max}}$$

$$T_{d_{min}} \leq T_d \leq T_{d_{max}}$$

$$\beta_{min} \leq \beta \leq \beta_{max}$$

3. Tuning of PID controllers using an evolutionary algorithm

From the observation of living beings, we can see that these reproduce, adapt, and evolve in relation to the environment where they develop. Some of the characteristics acquired during life may be inheritable by the next generation. The synthetic theory of evolution has been able to explain these processes and biological variations in detail [15]. This theory bases on genes as units of inheritance transfer, that is, functional units of basic information for the development of an organism. The genetic material of an individual is in its genotype. The genotype consists of an organization of hierarchical structures of genes. The complex information contained in

the genotype is expressed in the phenotype, that is, the visible characteristics and functionality of individuals. In the evolutionary process, the occurrence of small variations in the phenotypes, apparently random and without a clear purpose, is recognized. Such variations which are usually called mutations prove their efficacy in the light of the environment and prevail through the selection of the individual, or otherwise they disappear. The natural needing to produce offspring motivates the selection of individuals. Because of a severe competition for reproduction, which only the fittest individuals achieve, it is assumed that the offspring overcome their parents by inheriting their mixed characteristics. When resources in the environment become insufficient, only the fittest individuals will have a better chance of survival and reproduce. The selective pressure on individuals of a species makes them continually improve with respect to its environment. Evolutionary algorithms (EA) emulate the synthetic theory of evolution.

As natural evolution, an EA begins with an initial set of potential solutions to a specific problem. This set can be composed at random in a delimited searching space or using information of the problem. EA include operators that select and create new individuals. Crossover operator exchange of genetic material among "parents" to generate new "sons," and the mutation operator makes small variations. The new set of possible solutions is evaluated using a "fitness" function. When evaluating, the fittest are favored, leaving them as new "parents." The process is cyclically repeated to find the best solution to the problem in a delimited searching space.

EA encompass different approaches that transfer the behavior of adaptation and evolution of species, giving rise to several methods. Among the most popular approaches are genetic algorithms, genetic programming, evolutionary strategies, and evolutionary programming. Nowadays, EA are not only based in the biological evolution, rather EA are identified as algorithms that search iteratively for a solution through a population in evolution. Some of the main reasons for new optimization heuristics are the need to identify the interrelationships between the variables used to represent individuals according to the coding applied and the need to reduce the own parameters of the classical EA. New EA use operators different to the genetic ones. Some of those algorithms are differential evolution, estimation of distribution algorithms, and the multidynamics algorithm for global optimization.

Differential evolution at first glance is not based on any natural process. The proportional difference of two randomly chosen individuals from the population is added to a third individual, also randomly chosen. From this differential mutation, a fourth individual appears. This individual is compared against its parent, the third one. The best of them is selected to the next generation. The process is repeated until a stop criterion [16].

Estimation of distribution algorithms also bases their search on populations that evolve. The new population is recreated in each generation from the probability distribution obtained from the best individuals of the previous generation. The interrelationships between the variables are expressed explicitly through that distribution. There are no crossing or mutation operators. The process is repeated until a stop criterion [17].

In MAGO, a differential crossover is applied between the target individual and its mutant coming from a numerical derivation. A tournament chooses the best of them. The interrelationships

among the variables are explicit through a distribution of the population in each generation. The new population is sampled from a set composed of the best individuals until now, the historical trend and other individuals completely new. Next, this algorithm is explained in detail.

3.1. Multidynamics algorithm for global optimization

EA emulates the mechanisms of natural selection and genetic inheritance inspiring from the Neo-Darwinian theory of biological evolution. EA have evolved themselves to treat with artificial evolution processes. MAGO is a good example of this evolution. MAGO does not work with genetic operators [18]. MAGO starts with a random initial population on a search space bounded by the problem. To guarantee diversity and increase the exploitation of the search space, MAGO creates new individuals by means of three different subgroups of the population simultaneously. Each group has its own dynamics: a normal distribution over the searching space, a conservation mechanism of the best individual, and a strategy for maintaining diversity [19]. Because introducing statistics operators, MAGO provides a strong way to demonstrate the evolution. The mutation based on numerical derivation, generalizes the searching space where MAGO can acts. MAGO takes advantage of the concept of control limits [20] to produce individuals on each generation simultaneously from the three different subgroups. The size of the population is fixed, but the cardinality of each subgroup changes in each generation according to the first, second and third deviation of the actual population, respectively. The exploration is performed creating new individuals from these three subpopulations, individuals that are governed by any of their dynamics, the exploration is performed. For the exploitation, MAGO, looking for the goal, uses a greedy criterion in the first subset.

MAGO is evolutionary in the sense that works with a population of possible solutions randomly distributed throughout the searching space approaching iteratively to the final solution. MAGO is autonomous in the sense that it regulates its own behavior and does not need human intervention. Unlike other EA, MAGO has only two parameters: the population size and the number of generations. These two parameters could be removed, but they remain because in real situations they help to understand the context of the problem.

In each generation, MAGO divides the population into three subgroups. To know how many individuals will belong to each subgroup, the actual entire population is observed as having a normal distribution. The average location, the first, second and third dispersion of the whole population are calculated to form the three groups. To each subgroup is assigned many individuals as the cardinality of each different level of standard deviation. Each group has its own evolution. The cardinality of these subgroups changes autonomously in each generation.

The subgroup named Emerging Dynamics (G_1) creates a subpopulation of individuals around the individual with better characteristics; this group is the evolutionary elite of each generation, that is, the fittest individuals contributing with their genes to the next generation. The Crowd Dynamics (G_2) creates a group of individuals but around the current population mean, configuring the historical trend. This dynamic is applied to the largest portion of the population, and it is always close to the emerging dynamics, but never close enough to be confounded. These two dynamics could be merged within a same territory only until there are

sufficient and necessary conditions to ensure a full exploration of the searching space, usually at the end of the evolutionary process. The Accidental Dynamics (G_3) is a small group created by quantum speciation. It is established in isolation from individuals of the other two dynamics generation after generation. This portion of the population is always formed spontaneously and contains entirely new individuals. MAGO uses the covariance matrix of the population of each generation to establish a distribution of exploration. With the Accidental Dynamics, the main diagonal of the covariance matrix is different from zero, ensuring numerical stability of the evolutionary process. Because in each generation, the population is treated for its division by a normal distribution for its division according to the first, second, and third deviation, the subgroups G_1, G_2, and G_3 not interbreed.

Emerging Dynamics: This subset is created with the N1 fittest individuals in each generation. The N1 fittest individuals within the first standard deviation of the average location of the current population of individuals move in a line toward the best one of the entire population, in a kind of mutation that incorporate information from the best of all. The mutation and selection of individuals who have obtained the best values in their objective function is based on the simplex search method of numerical derivation [21]. MAGO uses only two individuals for this mutation, the best one and the trial one. If this movement generates a better individual, this one passes to the next generation; otherwise, its predecessor passes on with no changes. This method does not require gradient information for the derivation.

The fittest individuals are ordered from the best one. Test individuals are created bringing them closer to the best one, following the rule in Eq. (25):

$$x_T^{(j)} = x_i^{(j)} + F^{(j)} \times \left(x_B^{(j)} - x_i^{(j)}\right) \tag{25}$$

where $x_B^{(j)}$ is the best individual of generation j and $x_i^{(j)}$ is the selected fittest individual. $F^{(j)}$ is a matrix that includes information about the covariance of the problem variables, Eq. (26), including information about the interrelationships of the variables in the actual generation. The covariance matrix of the current population considers the effect of the evolution, and Eq. (25) propagates it on new individuals.

$$F^{(j)} = \frac{S^{(j)}}{\left\|S^{(j)}\right\|} \tag{26}$$

where $S^{(j)}$ is the sample covariance matrix of the individual population in generation j.

Crowd Dynamics: The number of individuals of this subgroup corresponds to the cardinality of the second deviation of the normal distribution of the actual population. This subgroup has the role of exploring the searching space in a neighborhood close to the population mean. If the population mean and dispersion matrix for generation j are $x_M^{(j)}$ and $S^{(j)}$, then the Crowd Dynamics individuals are created from a uniform distribution on the hyper-rectangle-$\left[LB^{(j)}, UB^{(j)}\right]$, see Eqs. (27) and (28). The diagonal of the population dispersion matrix of the generation j, described by Eq. (29).

$$LB^{(j)} = x_M^{(j)} - \sqrt{\text{diag}\left(S^{(j)}\right)} \tag{27}$$

$$UB^{(j)} = x_M^{(j)} + \sqrt{\text{diag}\left(S^{(j)}\right)} \tag{28}$$

$$\text{diag}\left(S^{(j)}\right) = \left[S_{11}^{(j)} \quad S_{22}^{(j)} \quad \dots \quad S_m^{(j)}\right]^T \tag{29}$$

Initially, the neighborhood around the mean may be large, but as evolution proceeds, this neighborhood is reduced, and the population mean is getting closer to the optimal but following on another path.

Accidental Dynamics: This group is a smaller one in relation to its impacts on the population. N3 new individuals are created from a uniform distribution over the whole search space, as in the initial population. The two dynamics mentioned above concentrate the population around their local optima. To maintain diversity, MAGO introduces new individuals in each generation with the accidental dynamic, sampling a uniform distribution throughout the search space. This dynamic also ensures the numerical stability of the covariance dispersion matrix. The accidental dynamics always guarantees the diversity and dispersion of the population, even if the other two groups already have converged. Following, the pseudo code of MAGO is presented.

MAGO Pseudo Code.

1: $j = 0$. Initial Generation.
2: Random initial population generation uniformly distributed over the searching space.
3: repeat
4: Evaluate each individual with the objective function.
5: Calculate the population covariance matrix and the first, second and third dispersion.
6: Calculate the cardinalities N1, N2 and N3 of the groups G_1, G_2 and G_3.
7: Select N1 fittest individuals, modify them according to Eq. (25), translate the winners toward the best one and make them compete. Pass the fittest to the next generation $j + 1$.
8: Sample N2 individuals from a uniform distribution in hyper rectangle $\left[LB^{(j)}, UB^{(j)}\right]$ and pass them to generation $j + 1$.
9: Sample N3 individuals from a uniform distribution over the whole search space and pass them to generation $j + 1$.
10: $j = j + 1$
11: until an ending criterion is satisfied

Cardinalities. For control tables, if the process is outside the control limits, then it is assumed that the process is out of order. The next step in MAGO is a type of variance decomposition, inspired by the well-known variance analysis (ANOVA). Consider the population dispersion matrix of generation j, $S^{(j)}$ and its diagonal $\text{diag}\left(S^{(j)}\right)$. If $Pob^{(j)}$ is the set of possible solutions in generation j, then three groups can be defined as in Eqs (30), (31), and (32). If N1, N2, and N3 are the cardinalities of the sets G_1, G_2, and G_3, then the cardinality of the Emerging Dynamics, Crowd Dynamics and Accidental Dynamics are set, respectively.

$$G_1 = \left\{ \boldsymbol{x} \in Pob^{(j)} / x_M^{(j)} - \sqrt{\text{diag}\left(S^{(j)}\right)} \leq \boldsymbol{x} \geq x_M^{(j)} + \sqrt{\text{diag}\left(S^{(j)}\right)} \right\} \tag{30}$$

$$G_2 = \left\{ \boldsymbol{x} \in Pob^{(j)} / x_M^{(j)} - 2\sqrt{\text{diag}\left(S^{(j)}\right)} \leq \boldsymbol{x} \geq x_M^{(j)} + \sqrt{\text{diag}\left(S^{(j)}\right)} \right\} \tag{31}$$

$$G_3 = \left\{ \boldsymbol{x} \in Pob^{(j)} / x \leq x_M^{(j)} - 2\sqrt{\text{diag}\left(S^{(j)}\right)}; \boldsymbol{x} \geq x_M^{(j)} + 2\sqrt{\text{diag}\left(S^{(j)}\right)} \right\} \tag{32}$$

where, $XM^{(j)}$ is the mean of the actual population and $Pob^{(j)} = G1 \bigcup G2 \bigcup G3$.

This way of defining the elements of each group is dynamical in nature and autonomous in MAGO. Cardinalities depend on the dispersion of the whole population in generation j.

3.2. Design of PI/PID controllers via an evolutionary algorithm

A feedback controller is a device that automatically manipulates a predetermined variable to ensure the balance of the system around an operating point. It compares the actual value of the controlled variable to its desired value (feedback) obtaining an error signal to calculate the control action so that it maintains or returns the system to the point of operation [9].

The output, $u(t)$, (control action) is a composite of three effects, K_p, the proportional action, T_i, the integral time and, T_d, the derivative time, which are calculated based on the error. An optimal PID controller consists of adjusting its parameters K_p, T_i, and T_d so that a performance criterion (error between the actual output of the plant regarding the desired value and/or effort control) is minimized. MAGO is a real-valued evolutionary algorithm, very efficient and effective instrument to solve problems in continuous domain. It has been chosen as a tool for estimating the parameters of a controller that minimizes an integral performance index. The representation of the evolutionary individual is a vector containing the controller parameters, as positive values in a continuous domain. See **Table 1**.

The fitness function for the optimization problem of a 1DoF PID controller is defined in Eq. (33). The error is calculated for each point of time, t_k, throughout the measurement horizon as the difference between the system output and the reference signal.

$$J(\theta) = J\left(K_p, T_i, T_d\right) = \text{Min}_\theta \left\{ J_{IAE} = \sum_{t_k}^{t_{kf}} |e(t_k)| \right\} \tag{33}$$

A complete analysis of the methods of tuning controllers based on SOSPD was made in [23, 24]. Each tuning rule for PID controllers has restrictions on the behavior of the plant, expressed

$K_p \in \boldsymbol{\mathcal{R}}$	$T_i \in \boldsymbol{\mathcal{R}}$	$T_d \in \boldsymbol{\mathcal{R}}$	$\beta \in \boldsymbol{\mathcal{R}}$

Table 1. Structure of the evolutionary individual.

in the range of validity, and has only been applied to a certain group of processes. In general, those rules are based on several relationships and/or conditions of the parameters defining the process model. Most methods for optimal tuning of SOSPD systems require, from experiments carried out directly on the plant, additional critical system information. Readers are referred to [25] for a good compilation of the PID tuning rules and [26] for a complete analysis of the different tuning rules characteristics and features. It is not always possible to perform experiments such as reaction curves and closed-loop tests because the extreme stress and oscillations may create instability and damage to the system. This scenario shows that a general rule for tuning PID controllers must be sought. A tuning method that best satisfies the operation requirements of each problem and ensures optimal values for the controller parameters according to the 2DoF PID chosen criterion. This situation could be reduced to an optimization problem consisting of minimizing an objective function. A suitable combination of the three parameters required by the PID controller will be the result of minimizing a performance criterion. This is the approach taken in this chapter.

MAGO has shown a great capacity to optimize nonlinear dynamical problems in the continuous domain. Because that, it has been selected for tuning several optimal 2DoF PID controllers. Next section is concerned to optimal 2DoF PID controllers satisfying an integral performance index and not requiring additional system information coming from experiments on the plant.

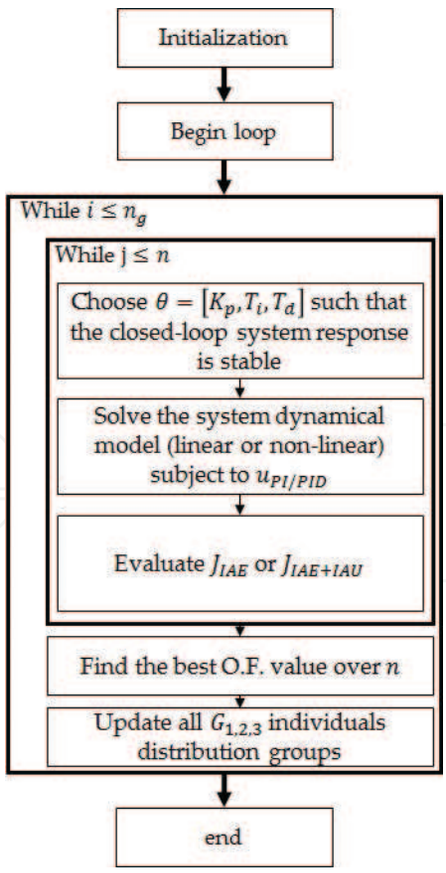

Figure 3. Flowchart of the MAGO algorithm for PI/PID controllers tuning.

With MAGO, the optimization of 2DoF PID controllers is made in only one stage and simultaneously for the regulator and servomechanism modes. The flowchart for tuning of PID controller using MAGO is shown in **Figure 3**. In this chapter, it is not included a convergence analysis of the MAGO; however, its convergence has been previously demonstrated in [18, 19].

4. 2DoF PID controllers tuning on benchmark plants

A 2DoF PID controller attempts simultaneously achieve good closed-loop servo-regulatory dynamical responses. Most of the recent literature working on 2DoF PID controllers had been based on both First Order Systems plus Time Delay (FOSPD) and Second Order plus Time Delay (SOSPD) models with satisfactory results [27–30].

In [22], a set of system models as a benchmark suitable for testing PID controllers was proposed. FOSPD and SOSPD were included in the benchmark. This set of system models presents different challenges of control because PID controllers are not well suited for all of them. From the best of our knowledge, none of the PID controller tuning rules applies to obtain suitable values for its parameters for all the systems in that benchmark.

In this chapter, nine different system models are taken from the benchmark and 2DoF PID controllers are designed, one for each system model. The 2DoF PID controller parameters obtained here are compared with parameters reported in [22] for the same system models. The 2DoF PID controller-tuning problem is formulated as a constrained optimization problem based on IAE and IAU performance indexes. Then, MAGO is employed to solve the optimization problem. Consider the system models given by Eqs. (34)–(42).

Multiple equals poles:

$$G_1(s) = \frac{1}{(s+1)^8} \tag{34}$$

Fourth-order system:

$$G_2 = \frac{1}{(s+1)(0.5s+1)(0.25s+1)(0.125s+1)} \tag{35}$$

Right half plane zero (non-minimum phase):

$$G_3 = \frac{1-5s}{(s+1)^3} \tag{36}$$

Time delay and lag: (FOSPD):

$$G_4 = \frac{1}{0.1s+1}e^{-s} \tag{37}$$

Time delay and double lag (SOSPD):

$$G_5 = \frac{1}{(0.1s+1)^2} e^{-s} \tag{38}$$

Fast and slow modes:

$$G_6 = \frac{100}{(s+10)^2}\left(\frac{1}{s+1} + \frac{0.5}{s+0.05}\right) \tag{39}$$

Conditionally stable system:

$$G_7 = \frac{(s+6)^2}{s(s+1)^2(s+36)} \tag{40}$$

Oscillatory system:

$$G_8 = \frac{25}{(s+1)(s^2+s+25)} \tag{41}$$

Unstable pole:

$$G_9 = \frac{1}{s^2-1} \tag{42}$$

In [22], the control problem was solved as an optimization process where the objective function was a combination of the IAE and IAU performance indexes. The main features of the procedure used in to solve the control problem were: (1) not only IAE but also IAU are included in the objective function establishing a kind of trade-off between the system performance and robustness due to the restriction imposed in the controller action effort through the IAU index. (2) Four 2DoF PID controller parameters were simultaneously optimized to take the full advantage of the control structure qualities. Traditionally, the tuning process for a 2DoF PID controller is carried out in two stages as follows: firstly, values of K_p, T_i, and T_d are found such that closed-loop system achieves some dynamical behavior. Secondly, the value of β is settled such that the closed-loop dynamical response of the system is improved when the system operates as servomechanism. These two stages tuning process imply that the closed-loop system responses are not optimal for any of the system operation modes. (3) The optimization process simultaneously considers both system operation modes, that is, optimum 2DoF PID controller parameters were found such that the closed-loop system response is optimal for both servo and regulatory modes. MATLAB functions `fminsearh` and `fmincon` were employed to solve the optimization problem. `Fminsearch` function was used to find a set of initial conditions for the controller parameters. Next, `fmincon` function was used trying to find the overall optimal controller parameters. The authors highlight that with this optimization strategy exists the possibility that the problem solution is not the global optimum of the objective function although the closed-loop systems performance was satisfactory in all cases.

In this chapter, a similar procedure for solving the optimization is adopted to facilitate the comparison of results. However, MAGO is adopted as optimizer instead of using native MATLAB functions `fminsearh` and `fmincon`. The objective function to be optimized is given by Eq. (43), where *r* and *s* refer to regulation-servo operation modes.

$$\text{Min}_{\theta}\left\{\left(\sum_{t_k}^{t_{kf}}|e(t_k)|+\sum_{t_k}^{t_{kf}}|e_u(t_k)|\right)_s+\left(\sum_{t_k}^{t_{kf}}|e(t_k)|+\sum_{t_k}^{t_{kf}}|e_u(t_k)|\right)_r\right\} \tag{43}$$

Subject to $G_{1,2,\ldots,9}$ (see Eqs. (34)–(42))

$$U(s)=K_p\left(\beta+\frac{1}{T_i s}\right)R(S)-K_p\left(1+\frac{1}{T_i s}+T_d s\right)Y(s)$$

$$K_{p_{min}} \le K_p \le K_{p_{max}}$$

$$T_{i_{min}} \le T_i \le T_{i_{max}}$$

$$T_{d_{min}} \le T_d \le T_{d_{max}}$$

$$\beta_{min} \le \beta \le \beta_{max}$$

Table 2 summarizes results obtained with MAGO and in [22] for 2DoF PID controller parameters. **Table 3** summarizes results obtained with MAGO and in [22] for each of the IAE and IAU performance indexes.

From **Table 2**, it is seen that parameters obtained with MAGO for 2DoF PID controllers are different in all cases, although they have similar magnitude scales. This implies that MAGO found a different minimum for the objective function (see Eq. (43)).

From **Table 3**, it is seen that MAGO found a better minimum value for the integral performance indexes except for the performance index IAU_s for G_6, G_8, and G_9 systems. In [22], an

System model	K_p		T_i		T_d		β	
	[22]	**MAGO**	**[22]**	**MAGO**	**[22]**	**MAGO**	**[22]**	**MAGO**
G_1	0.890	0.9544	5.147	5.4354	1.999	1.8095	0.661	0.4453
G_2	3.637	3.2947	1.334	1.2791	0.420	0.4270	0.222	0.3096
G_3	0.335	0.3515	2.665	2.6949	0.774	0.7941	0.844	0.5355
G_4	0.423	0.5278	0.538	0.5765	0.137	0.1557	1.000	0.2593
G_5	0.367	0.5013	0.497	0.6117	0.103	0.2380	1.000	0.7687
G_6	0.626	1.8491	0.441	0.8014	0.000	0.1580	0.000	0.9654
G_7	65	68.4154	1.736	1.2209	0.632	0.3924	0.141	0.0228
G_8	0.596	0.6238	0.424	0.3392	0.172	0.1877	1.000	0.4617
G_9	40	33.7561	1.430	0.7854	0.297	0.3159	0.231	0.0486

Table 2. 2DoF PID controller parameters.

System model	IAE_s		IAU_s		IAE_r		IAU_r		Total	
	[22]	MAGO	[22]	MAGO	[22]	MAGO	[22]	MAGO	[22]	MAGO
G_1	8.420	8.9159	2.878	1.9858	5.999	5.8509	7.643	6.8976	24.943	23.6502
G_2	1.460	1.3676	1.511	0.6211	0.375	0.4045	1.692	0.6923	5.039	3.0855
G_3	8.575	9.0000	1.969	1.1623	16.912	17.0410	9.005	7.8505	36.462	35.0539
G_4	1.396	1.5763	1.318	0.4935	1.300	1.3516	2.338	1.4282	6.353	4.8496
G_5	1.464	1.5213	1.351	0.3665	1.431	1.3194	2.415	1.3957	6.610	4.6029
G_6	1.543	0.7787	1.734	7.8872	1.205	0.4895	2.311	0.5851	6.793	9.7405
G_7	1.555	1.2055	1.212	11.187	0.027	0.0180	1.121	0.1752	3.915	12.5855
G_8	1.176	1.2154	1.593	0.7958	1.003	0.9138	2.112	1.0807	5.883	4.0057
G_9	1.064	0.7669	2.087	16.310	0.036	0.0242	1.173	0.2073	4.360	17.3089

Table 3. 2DoF PID controller IAE and IAU performance indexes.

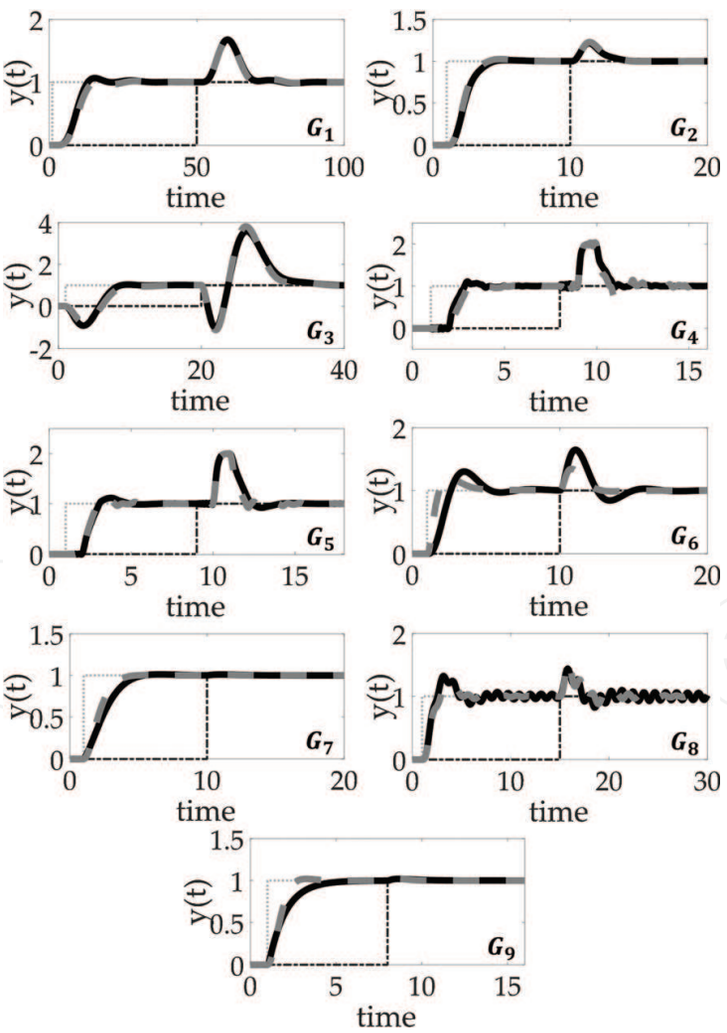

Figure 4. Simulation results for 2DoF PID controllers applied to the benchmark. Closed-loop system dynamical response comparison: 2DoF PID controller tuned with parameters reported in [22] (solid line) and results from MAGO (dashed line).

additional optimization constraint was imposed for systems G_6, G_8, and G_9 to avoid changes larger than 10 times the steady-state value of the control action, that is, $\Delta u < u_{ss}$, where u_{ss} is the steady-state of u. In this chapter, this additional constraint was not included. This constraint could be incorporated as follows: $\Delta u = K_p \beta \Delta r < 10 u_{ss}$.

Figure 4 shows the closed-loop dynamical response for systems given by Eqs. (34)–(42). To test the 2DoF PID controllers' performance, a step change in $r(t)$ at $t = 0$ (dotted line) and a step change in $d(t)$ at $t = 50$ (dash-dot line) were applied. Results by MAGO (dashed line) are compared with results in [22] (solid line). PID controllers tuned by MAGO perform similar but faster for G_6, G_8, and G_9 systems.

Figure 5 shows control actions for systems given by Eqs. (34)–(42). In **Figure 5**, results obtained by MAGO (dashed line) and in [22] (solid line) are contrasted. From **Figure 5**, it is seen that the control action effort is similar for most cases except for G_6, G_8, and G_9 systems, for which an

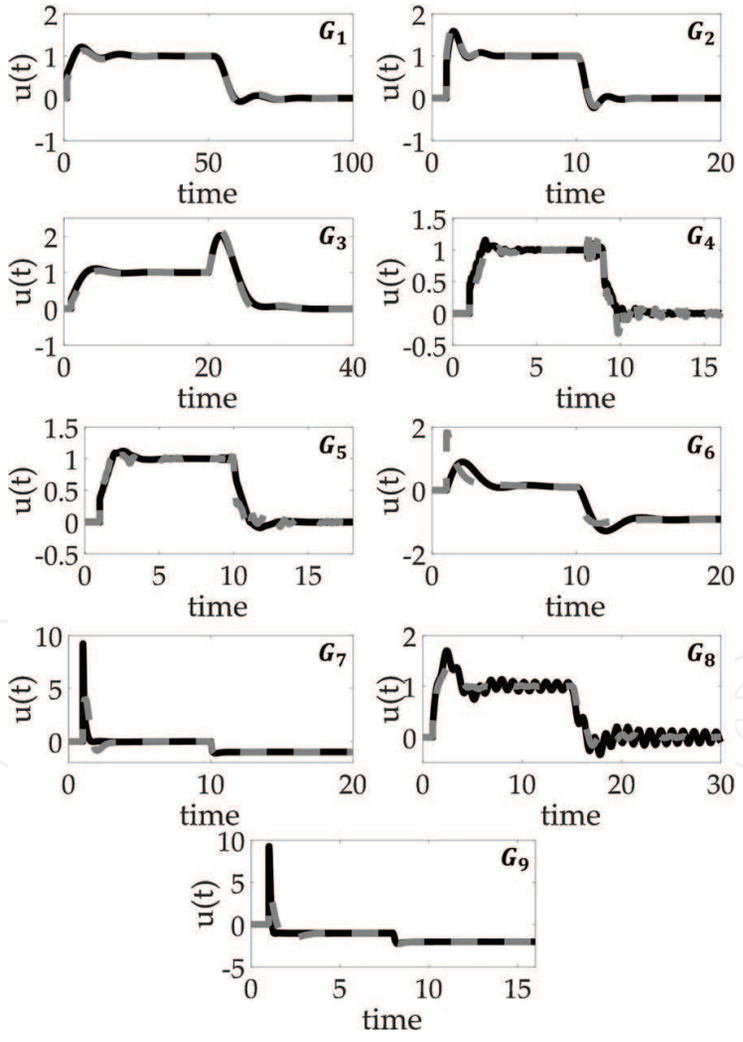

Figure 5. Simulation results for 2DoF PID controllers applied to the benchmark. Closed-loop dynamical response of the control action: 2DoF PID controllers tuned with parameters in [22] (solid line) and results from MAGO (dashed line).

additional optimization constrain was included in [22] to avoid control action changes larger than 10 times u_{ss}.

From **Figures 4** and **5**, it is seen that MAGO solves the optimization problem satisfactorily because the control objective is achieved for all systems. For instance, MAGO is not only suitable to solve the 1DoF PID control problem based on SOSPD [6], but also to solve the 2DoF PID control problem based on a wide range of systems, each one of them with different dynamical characteristics and challenges from the control point of view.

5. Case study: Evolutionary PI controller tuning for a buck DC-DC converter based on its nonlinear model

Power electronic converters (PEC) are electronic circuits that are commonly designed to regulate the voltage in their output when the input is a nonregulated current or voltage source. A PEC is highly efficient, highly reliable, negligible maintenance and very small. A PEC is usually composed of switches (Q IGBT or Mosfets and D diodes). Because switches a PEC is nonlinear dispositive with an interesting behavior, so a DC-DC buck converter is chosen in this chapter as a case of study.

DC-DC buck converters are nonlinear systems which analysis and control could be difficult. Linear techniques based on classical controller have problems related to the stability around the operation point [31]. Nonlinear controllers can be implemented to improve the stability of the converter, but such techniques could be complex [32, 33]. Control techniques-based artificial intelligence simplifies the design and implementation, not requiring the mathematical model; nonetheless, they are designed based on expert knowledge of the converter [34–36]. Sliding mode control technique has the advantage to reject easily perturbations, but variable frequency of switching may be handled [37]. There are many articles dealing with control for the DC-DC buck converter but is still no consensus on the control strategy should be implemented. This is another reason to choose a DC-DC buck converter as a study case.

DC-DC buck converter, or step-down converter, is a power converter that steps down the voltage from its supply (input) to its load (output). **Figure 5** shows the equivalent circuit of the DC-DC buck converter. It is composed of an inductor (L), a capacitor (C), a diode (D), and a switch (Q) (IGBT is considered as the switch for the analysis). The converter is supplied by a DC input voltage (v_g) and feeds in its output (v_o) a resistive load (R). **Figure 6** also depicts the reference currents and voltages of the circuit, using passive sign convention. In this chapter, it is assumed that the converter operates in continuous condition mode (CCM).

The converter has two states per switching cycle (T_s) according to the position of the switch Q when CCM operation is considered. The on state is when Q is on (closed) and D is inversely polarized (open), while the off state is when Q (open) is off and D is directly polarized (closed). Dynamical model and steady-state analysis can be done if Kirchhoff laws are applied, see Eqs. (44) and (45). Step-by-step DC-DC buck converter dynamical model deduction can be

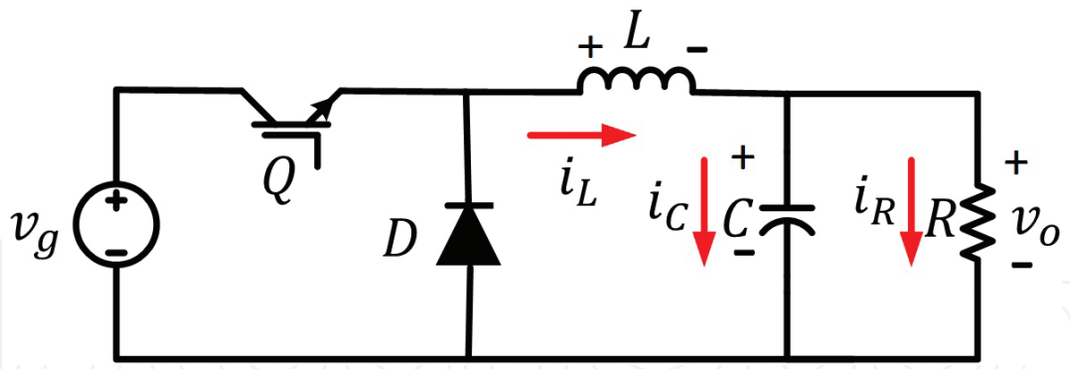

Figure 6. DC-DC buck converter equivalent circuit schematic.

found in [38]; where u is the switching function ($u = 1$ for on-state and $u = 0$ for off-state). The nonlinearity is because the product between v_g and u.

$$L\frac{di_L}{dt} = v_g u - v_c \tag{44}$$

$$C\frac{dv_c}{dt} = \left(i_L - \frac{v_c}{R}\right) \tag{45}$$

The DC-DC buck converter model given by Eqs. (44) and (45) describes the nonlinear dynamical evolution of the system. From the control point of view, a model given by Eqs. (44) and (45) is inconvenient, and a linear approximation is preferred. The linear DC-DC buck converter state-space representation is given by Eqs. (46) and (47). Where d is an input control (duty cycle) and correspond with $\langle u \rangle = d$, $d \in [0, 1]$, the average value of u. V_g is the average value of v_g (state in its rated value) and D is d in an operation point (D is also defined as the reason between the on-time and the switching-time $D = t_{on}/T_s$). Note that v_g is chosen as a system input, and it represents a system disturbance.

$$\begin{bmatrix} i_L \\ \dot{v_c} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ v_c \end{bmatrix} + \begin{bmatrix} \frac{V_g}{L} & \frac{D}{L} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ v_g \end{bmatrix} \tag{46}$$

$$\begin{bmatrix} i_L \\ v_o \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_L \\ v_c \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ v_g \end{bmatrix} \tag{47}$$

Transfer functions ($G_{i_L d}$, $G_{i_L v_g}$, $G_{v_c d}$ and $G_{v_c v_g}$) can be obtained by applying the realization of Eq. (5). Transfer functions are shown by Eq. (48).

$$\begin{bmatrix} G_{\frac{i_L}{d}} & G_{\frac{i_L}{v_g}} \\ G_{\frac{v_C}{d}} & G_{\frac{v_C}{v_g}} \end{bmatrix} = \begin{bmatrix} \frac{CV_g Rs + V_g}{RLCs^2 + Ls + R} & \frac{RCDs + D}{RLCs^2 + Ls + R} \\ \frac{V_g R}{RLCs^2 + Ls + R} & \frac{DR}{RLCs^2 + Ls + R} \end{bmatrix} \tag{48}$$

Transfer function $G_{v_c d}$ can be used for controlling the output voltage by means of the change of d; similarly, $G_{i_L d}$ can be used for controlling the inductor current. $G_{v_c v_g}$ and $G_{i_L v_g}$ can be used to determine the change on v_c and i_L when v_g is considered as a perturbation. Narrowing this chapter to controlling v_c, $G_{v_c d}$ and $G_{v_c v_g}$ are the transfer functions that we are focus on; similarly, $G_{i_L d}$ and $G_{i_L v_g}$ should also be used for the explanation. Classic control commonly uses $G_{v_c d}$ to regulate v_c; nevertheless, $G_{v_c v_g}$ also affects v_c and must be taken into account if the implementation of a robust control strategy is desired. For controlling purposes, v_g is a perturbation while d is the control input that allows the control of the system, both v_g and d affect v_c (see **Figure 7**).

From the control point of view, the DC-DC buck converter can operate in two modes, as regulator and servomechanism. Regulation mode objective is to maintain constant the converter output voltage against any system disturbance. Servomechanism mode objective is to track an output voltage reference. Both operation modes can be combined at any moment. For instance, the selected control strategy for the DC-DC buck converter should assure a good performance for both operation modes. In this chapter, a PI/PID-based control structure fulfills both DC-DC buck converter control objectives. A DC-DC buck converter is parameterized as follows: $C = 22\ \mu F$, $L = 0.5$ mH, $R = 10\ \Omega$, $V_g = 12\ V$, $V_o = 6\ V$, $D = 0.5$, $R_L = R_c = 1\ \Omega$.

A rigorous way to tune PI controllers for DC-DC buck converters is through the pole placement method [36]. As PI controller tune requirements, wide accepted specifications are a bandwidth $BW \geq \frac{1}{5} f_{sw}$, where f_{sw} is the converter switching frequency, and a damping factor $\zeta = 1/\sqrt{2}$. These closed-loop requirements are settled to establish a trade-off between DC-DC buck closed-loop performance and robustness.

Sisotool is a friendly MATLAB environment to tune PI/PID controller parameters. **Figure 8** shows the DC-DC buck converter root locus diagram for the PI and PID controllers. In **Figure 8**, points represented as cruces correspond to poles and circles correspond to zeros, the diagonal line from the origin corresponds to $\zeta = 1/\sqrt{2}$. In consequence, shaded area is for $\zeta < /\sqrt{2}$, while non-shaded area is for $\zeta > 1/\sqrt{2}$. From **Figure 8a**, it is possible to observe that $\zeta = 1/\sqrt{2}$ cannot be achieved no matter how large the value of the PI controller real zero is. On the other hand, in **Figure 8b**, it is possible to observe that $\zeta = 1/\sqrt{2}$ can be achieved with a PID controller. **Figure 8** shows the DC-DC buck converter closed-loop bode diagram. From **Figure 9**, it is possible to observe that $BW = \frac{1}{5} f_{sw} = 4$ kHz. PID parameters are $K_p = 0.1624$,

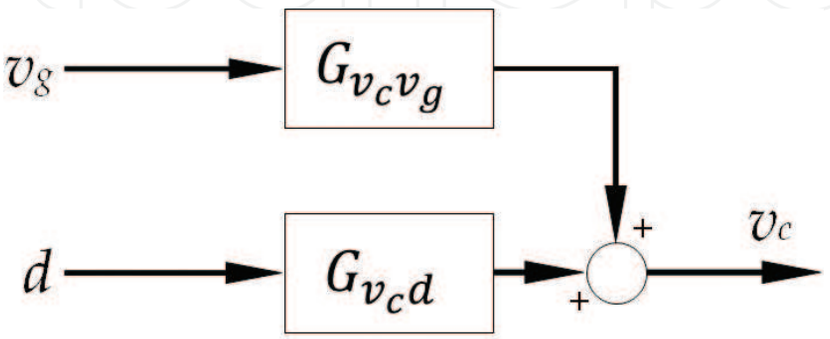

Figure 7. DC-DC buck converter open-loop block diagram.

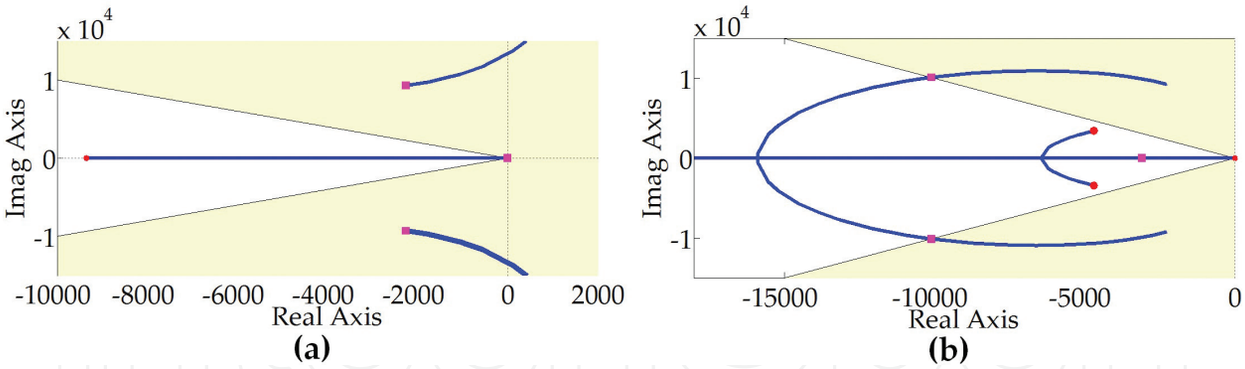

Figure 8. DC-DC buck converter root locus: (a) PI controller and (b) PID controller.

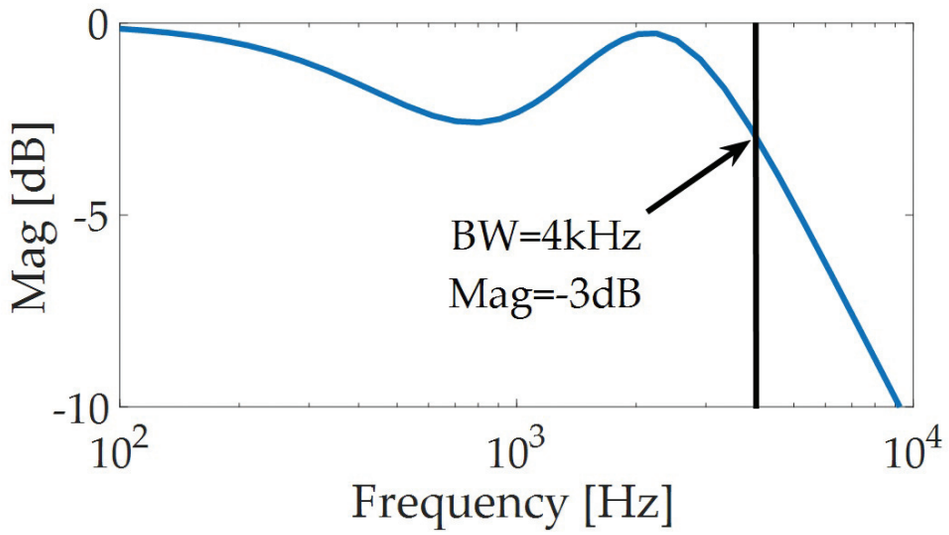

Figure 9. DC-DC buck converter closed-loop bode diagram for the PID controller.

$T_i = 1.7240\,e-3\,[s]$ and $T_d = 1.6763\,e-5\,[s]$. Furthermore, the tuned PID controller fulfills the desired closed-loop requirements for the DC-DC buck converter, while a PI controller cannot be tuned to fulfill the closed-loop requirements for the DC-DC buck converter.

PI rather than PID control structures are preferred in PECs applications because PI-based control structures significantly reduce the feedback induced noise. In this chapter is proposed an alternative way to tune PI controllers for DC-DC buck converters in order to overcome the limitations with the pole placement method. PI controller tuning problem is formulated as a constrained optimization problem and solved using MAGO as in Eq. (49). $\theta = \begin{bmatrix} K_p & T_i \end{bmatrix}$ are the PI controller parameters, α and γ are tuneable weighs establishing a trade-off between IAE and IAU performance indexes and $t_{s_{max}}$ is the maximum allowed closed-loop setting time.

$$\text{Min}_\theta \left\{ \alpha \left(\sum_{t_k}^{t_{kf}} |e(t_k)| \right) + \gamma \left(\sum_{t_k}^{t_{kf}} |e_u(t_k)| \right) \right\} \tag{49}$$

Subject to *Buck DC-DC nonlinear dynamical model* Eqs. (43) and (45)

$$u(t) = K_p\left\{e(t) + \frac{1}{T_i}\sum_{t_k}^{t_{kf}} |e(t_k)|\right\}$$

$$0 \le K_p \le 5$$

$$0 \le T_i \le 1e-3$$

$$\alpha = 1$$

$$\gamma = 0.5$$

Main features of the formulated optimization problem are: (1) DC-DC buck nonlinear dynamical model is incorporated instead of duty to output voltage transfer function. Using the nonlinear dynamical model of the DC-DC buck converter, it is possible to solve the optimization problem simultaneously considering regulation and servomechanism operation modes. (2) Objective function includes simultaneously IAE and IAU performance indexes. The inclusion of the IAU gives a kind of closed-loop robustness because it limits the maximum control action effort avoiding system oscillations and actuator saturations. (3) The optimization process is carried out based on the temporal dynamical response of the DC-DC buck converter operating in closed-loop. For each MAGO iteration, it is necessary to solve the DC-DC buck converter nonlinear dynamical model in the closed-loop mode. To solve the nonlinear dynamical model in the closed-loop mode, it is necessary to define an experiment such that multiple set-point changes and disturbances appear through the simulation time. The solution of the optimization problem is the set of PI controller parameters that minimize both IAE and IAU indexes for the experiment setup.

The total simulation time for the DC-DC buck converter is 25 min. The experiment setup for the optimization process is as follows: for the time interval $t \in [0, 5]$ ms, the system is at the nominal conditions. For $t \in (5, 10]$ ms, the set-point for the output voltage is settled in 8 V. For $t \in (10, 15]$ ms, the input voltage is settled in 14 V, while the output voltage remains in 8 V. For $t \in (15, 20]$ ms, output and input voltages are settled in 4 V and 10 V, respectively. For $t \in (20, 25]$ ms, output and input voltages are returned to their nominal values. Both adjustable MAGO parameters n and ng were settled in 100. The optimization problem solution using MAGO is: objective function value equals 0.1066, PI controller parameters values are $K_p = 0.0382$ and $T_i = 1.5364\, e - 4\ [s]$.

PSIM is a recognized platform to simulate and validate the control system performance for PEC. A PSIM simulation is carried out for the DC-DC buck converter to validate MAGO optimization results. A DC-DC buck converter simulation is implemented on PSIM including parasitic losses in their passive elements as equivalent series resistances (ESR). Simulation aims to test the real closed-loop system performance. Therefore, two DC-DC buck converter closed-loop simulations are carried out as follows: (1) PID controller obtained by the pole placement method is implemented and its performance is tested. (2) PI controller obtained by MAGO is implemented and its performance is tested. In both cases, the experiment setup used in MAGO optimization is applied to test PID and PI controllers' performance.

$G_{v_c d}$ and $G_{v_c v_g}$ were deduced in this chapter without including losses in the inductor and capacitor; switching losses which includes the losses in the IGBT and the diode neither were considered. A comparison between a DC-DC buck converter including resistive losses (PSIM simulation) and a DC-DC buck converter which does not include them (MATLAB simulation) are presented in **Figure 10**. For this purpose, pointed line diagrams correspond to PSIM bode diagrams (nonlinear model), while continuous line diagrams correspond to MATLAB bode diagrams of transfer functions $G_{v_c d}$ and $G_{v_c v_g}$ (linear model). PSIM simulation corresponds to the most realistic approximation of the system due to it is the nonlinear representation which includes the resistive losses. In **Figure 10**, there is a big difference between MATLAB and PSIM bode diagrams. This chapter uses the linear model (Eqs. 46 and 47) to implement the control strategy, while the DC-DC buck converter to be controlled includes the resistive losses.

Figure 11 shows the DC-DC buck converter closed-loop dynamical response for both PI controller tuned by MAGO (PI-MAGO) and PID controller tuned by the pole placement method. **Figure 10a** shows both v_g and $v_{o_{Ref}}$ time trajectories. From **Figure 11a**, it is seen that these time trajectories agree with the experiment setup proposed for the optimization problem. Multiple set-point changes ($v_{o_{Ref}}$) and disturbances (v_g) are applied to test the closed-loop dynamical response of the DC-DC buck converter.

Figure 11b shows the dynamical closed-loop response for the DC-DC buck converter output (v_o). From **Figure 11b**, it is seen that the converter control objective is achieved satisfactorily. DC-DC buck converter in closed-loop mode can simultaneously track the set point and reject the disturbance. Both faster than 2.5 ms and with an approximated over-shoot of 12 and 2.5% for the PI and PID controllers, respectively. A remarkable issue is that the converter control objective is achieved when the PSIM DC-DC buck converter model includes the parasitic losses in their passive elements. This implies that both controllers are robust to the model

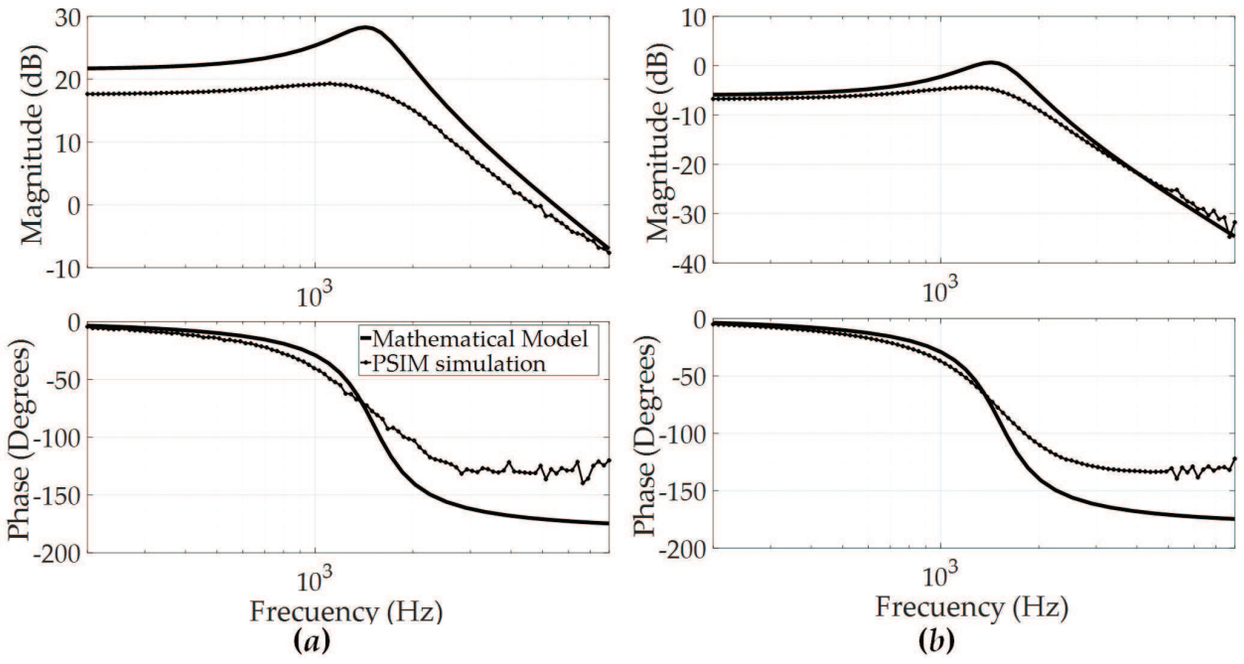

Figure 10. PSIM simulation and nonlinear mathematical model comparison: (a) $G_{v_c d}$ and (b) $G_{v_c v_g}$.

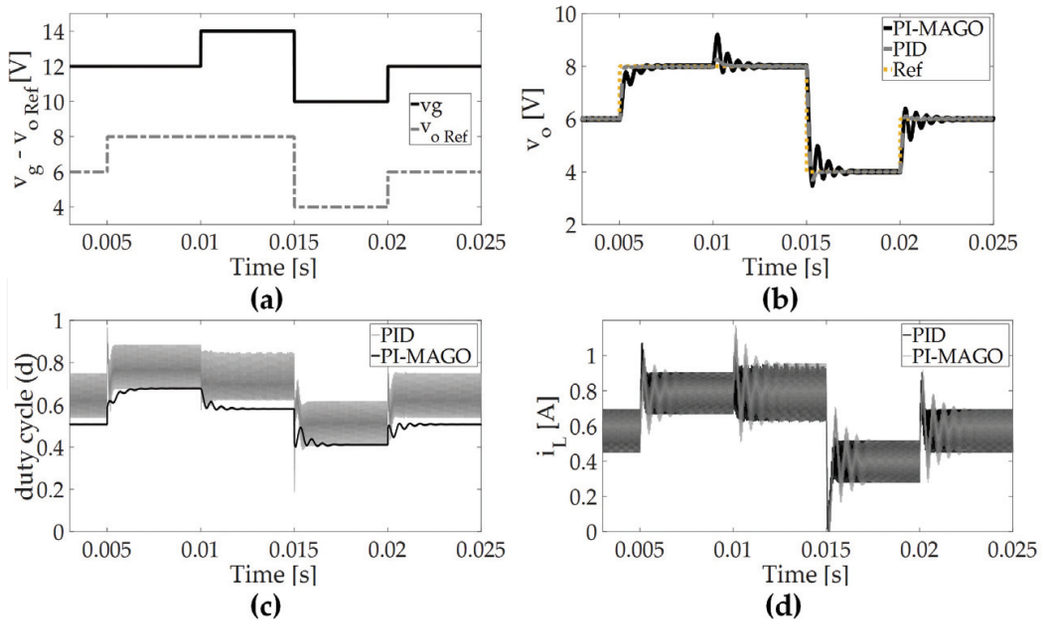

Figure 11. DC-DC buck converter closed-loop PSIM simulation.

mismatch. From **Figure 11b**, it is seen that PID controller performs better than PI controller does; however, the PI performance, it is also acceptable.

Figure 11c shows the control action (d) dynamical response for both PI and PID controllers. From **Figure 11c**, it is seen that d has significant oscillations for PID controller compared with PI controller. The most probable reason for these oscillations is the induced noise caused by the time derivative constant (T_d) in the PID controller. For instance, the PI controller is preferred in PEC applications because the closed-loop induced noise is smaller than in the PID controller case, although the PI controller has a poor dynamical performance than PID controller. An important issue related with d behavior is that never reaches its maximum or minimum value, implying that for the selected setup the actuator does not saturates.

Figure 11d shows the inductor current (i_L) dynamical response for both PI and PID controller. From **Figure 10d**, DC-DC buck converter operates in the CCM overall simulation time.

In summary, both PI and PID controllers achieve the DC-DC buck converter control objective. PID controller performs better than PI controller, but the last one has a smaller induced noise in d, which is a desired characteristic in PEC applications. In consequence, MAGO solution improves the solution obtained by the pole placement method, where a PI controller was not possible to tune because the root locus limitations (see **Figure 8**).

6. Conclusions

A method of tuning optimal controllers on nonlinear systems through the evolutionary algorithm MAGO has been successfully developed and implemented. The MAGO resolves the tuning as a constrained nonlinear optimization problem for both 1DoF and 2DoF PI/PID

controllers on several types of plants running on different modes of operation. To ensure the desired performance of the control loop, everything is reduced to the characterization and adjusting of the value of a set of tuning parameters whatever the chosen control structure is. The usual tuning controller procedures involve uncertainty and time to obtain the right parameters, generally by trial and proof by trained personal depending of the method used. This chapter showed a way to reduce these challenges. MAGO straightforwardly assesses the controller's parameters penalizing the error between the reference value and the output of the plant and minimizing the IAE and IAU performance criteria.

Traditional PID controller tuning rules are restricted to certain values on the behavior of the system and are limited to an only one type of operation. Most methods require experiments to be carried out directly on the plant to get additional necessary system information to apply them; these activities are not always possible to achieve because the triggering of extreme stresses and oscillations of the plant which may create instability and damages on the system, so that, they are not recommended. For benchmark models, the PID controller tuning was made by the MAGO without additional knowledge of the plant. The evolutionary solution obtained with the MAGO covers all those restrictions, extends their maximum and minimum limits, and it does not need additional experimental information from the plants and is suitable for both servo and regulator operating modes. The results showed that regardless of both the plant or controller models used, MAGO gets a satisfactory closed-loop system performance in agreement with the literature reported.

A linear model without losses for the DC-DC buck converter was used to implement the control strategy while a nonlinear system which includes parasitic losses was simulated as a most realistic system in PSIM. There was a big difference between the bode diagrams of the linear model and the nonlinear system; nevertheless, the controllers tuned by MAGO have a successfully behavior in terms of performance and robustness. MAGO controllers let overcome the limitations of traditional PI controllers.

The challenge of rely on a single method to assess the controller for different kinds of plants has been solved applying the evolutionary algorithm MAGO as a tool of optimization. The MAGO was applied to a set of benchmark plants, represented in both transfer functions and differential equation systems, with a control loop operating on both servo and regulator modes. The optimal tuning values of the K_p, T_i and T_d parameters for the optimal PID controller found by the evolutionary method achieved successful results for each of the cases studied. Noticeable results tuning with the algorithm MAGO were obtained when comparing the performance of the traditional PID controller performance for the DC-DC buck converter against the PI-MAGO controller. PI-MAGO controller has a comparable performance with the PID controller tuned by pole placement method. Moreover, PI-MAGO controller minimized the induced closed-loop noise.

This chapter showed that, although there would be options in the traditional rules of controllers tuning, the use of heuristic algorithms is indeed easy than using the classic methods of optimal tuning. The evolutionary algorithm MAGO was used as a tool to optimize the controller parameters for optimal 2DoF PID controllers working on benchmark plants and to optimize the PI controller parameters in a DC-DC buck converter application. Regardless of

the operating mode of the controller and the representation type of the plant used, better results are yielded when optimization is made with the MAGO algorithm than with the traditional methods for optimal tuning.

Acknowledgements

The authors gratefully acknowledge the University of Antioquia (UdeA) for the support of the project CODI 2015-7747. This work was partially supported by COLCIENCIAS (Fondo Nacional de Financiamiento para la Ciencia, la Tecnología y la Innovación Francisco José de Caldas) with the doctoral scholarship 727-2015.

Author details

Jorge-Humberto Urrea-Quintero[1]*, Jesús-Antonio Hernández-Riveros[2] and Nicolás Muñoz-Galeano[1]

*Address all correspondence to: jhurreaq@unal.edu.co

1 Faculty of Engineering, Universidad de Antioquia, Medellín, Colombia

2 Department of Electrical Energy and Automatics, Universidad Nacional de Colombia, Sede Medellín, Colombia

References

[1] Åström KJ, Hägglund T. Advanced PID Control. Systems and Automation Society: ISA-The Instrumentation; 2006 http://lup.lub.lu.se/record/535630

[2] O'Dwyer A. Handbook of PI and PID Controller Tuning Rules. 3rd edition. Imperial College Press; 2009. 608 p. ISBN: 978-1-908978-77-6

[3] Alcántara S, Vilanova R, Pedret C. PID control in terms of robustness/performance and servo/regulator trade-offs: A unifying approach to balanced autotuning. Journal of Process Control. 2013;**23**(4):527-542. DOI: 10.1016/j.jprocont.2013.01.003

[4] Alfaro VM, Vilanova R. Model-Reference Robust Tuning Design Methodology. In: Model-Reference Robust Tuning of PID Controllers. Springer International Publishing; 2016. pp. 29-34. DOI: 10.1007/978-3-319-28213-8_4

[5] Reynoso-Meza G, Sanchis J, Blasco X, Freire RZ. Evolutionary multi-objective optimisation with preferences for multivariable PI controller tuning. Expert Systems with Applications. 2016;**51**(1):120-133. DOI: 10.1016/j.eswa.2015.11.028

[6] Willjuice Iruthayarajan M, Baskar S. Evolutionary algorithms based design of multivariable PID controller. Expert Systems with Applications. 2009;**36**(5):9159-9167. DOI: 10.1016/j.eswa.2008.12.033

[7] Hernández-Riveros JA, Urrea-Quintero JH, Carmona-Cadavid CV. Evolutionary tuning of optimal PID controllers for second order systems plus time delay. In: Computational Intelligence. Cham: Springer; 2016. pp. 3-20. DOI: 10.1007/978-3-319-26393-9_1

[8] Fleming PJ, Purshouse RC. Evolutionary algorithms in control systems engineering: A survey. Control engineering practice. 2002;**10**(11):1223-1241. DOI: 10.1016/S0967-0661(02)00081-3

[9] Ghoreishi SA, Mohammad AN, Basiri SO. Optimal design of LQR weighting matrices based on intelligent optimization methods. International Journal of Intelligent Information Processing. 2011;**2**(1). DOI:10.4156/ijiip.vol2

[10] Tijani IB, Akmeliawati R, Abdullateef AI. Control of an inverted pendulum using MODE-based optimized LQR controller. In: Industrial Electronics and Applications (ICIEA), 8th IEEE Conference on; June 2013. p. 1759-1764. DOI:10.1109/ICIEA.2013.6566653

[11] Hassani K, Lee, WS. Optimal tuning of linear quadratic regulators using quantum particle swarm optimization. In: Proceedings of the International Conference of Control, Dynamics and Robotics; May 2014. p. 1-8

[12] Li Y, Ang KH, Chong GC. PID control system analysis and design. IEEE Control Systems magazine. 2006;**26**(1):32-41. DOI: 10.1109/MCS.2006.1580152

[13] Hernández-Riveros JA, Urrea-Quintero JH. SOSPD controllers tuning by means of an evolutionary algorithm. International Journal of Natural Computing Research (IJNCR). 2014;**4**(2):40-58. DOI:10.4018/978-1-4666-7456-1.ch038

[14] Garpinger O, Hägglund T, Åström KJ. Performance and robustness trade-offs in PID control. Journal of Process Control. 2014;**24**(5):568-577. DOI: 10.1016/j.jprocont.2014.02.020

[15] Freeman Scott, Herron Jon C. Evolutionary Analysis. 5th ed. Pearson; 2014. 848 p. ISBN-13: 9780321868992.

[16] Chakraborty U, editor. Advances in differential evolution. Springer; 2008. DOI:10.1007/978-3-540-68830-3

[17] Hansen, N. Towards a new evolutionary computation advances in the estimation of distribution algorithms, vol. 192 de. Studies in Fuzziness and Soft Computing, Springer; 2006. pp. 75-102. ISBN 978-3-33 540-32494-2

[18] Hernández JA, Ospina JD. A multi dynamics algorithm for global optimization. International Journal of Mathematical and Computer Modelling. Elsevier. 2010;**52**(7-8):1271-1278. DOI: 10.1016/j.mcm.2010.03.024

[19] Hernández-Riveros JA, Villada-Cano D. Sensitivity analysis of an autonomous evolutionary algorithm. Advances in Artificial Intelligence-IBERAMIA; 2012. p. 271-280. DOI: 10.1007/978-3-642-34654-5_28

[20] Montgomery DC. Introduction to Statistical Quality Control. 6th edition. John Wiley & Sons; 2007. 734 p. ISSN: 978-0-470-16992-6

[21] Nelder JA, Mead R. A simplex method for function minimization. The Computer Journal. 1965;**7**(4):308-313. DOI: 10.1093/comjnl/7.4.308

[22] Alfaro VM, Arrieta O, Vilanova R. Control de Dos-Grados-de-Libertad (2-GdL) aplicados al "Benchmark" de Sistemas para Controladores PID. Revista Iberoamericana de Automatica e Informatica Industrial RIAI. 2009;**6**(2):59-67. DOI: 10.1016/S1697-7912(09)70093-7

[23] Comparación del desempeño de los controladores PI y PID [Internet]. 2004. Available from: http://eie.ucr.ac.cr/uploads/file/proybach/pb0431t.pdf [Accessed: 2018-03-07]

[24] Sintonización de controladores PI/PID con los criterios IAE e ITAE, para plantas de polo doble [Internet]. 2005. Available from: http://eie.ucr.ac.cr/uploads/file/proybach/pb0508t.pdf [Accessed: 2018-03-07]

[25] Åström KJ, Hägglund T. Benchmark systems for PID control. IFAC Proceedings Volumes. Elsevier. 2000;**33**(4):165-166. DOI: 10.1016/S1474-6670(17)38238-1

[26] Desanti J. Robustness of Tuning Methods of Based on Models of First-Order plus Dead Time PI and PID Controllers. Escuela de Ingeniería Eléctrica: Universidad de Costa Rica; 2004

[27] Alfaro VM, Vilanova R, Arrieta O. Maximum sensitivity based robust tuning for two-degree-of-freedom proportional−integral controllers. Industrial & Engineering Chemistry Research. 2010;**49**(11):5415-5423. DOI: 10.1021/ie901617y

[28] Alfaro VM, Vilanova R. Model reference based robust tuning of five-parameter 2DoF PID controllers for first-order plus dead-time models. In: Control Conference (ECC), 3931-3936 July 2013. European: IEEE; 2013. pp. 3931-3936

[29] Kumar M, Patel V. Tuning of two degree of freedom PID controller for second order processes. International Journal of Science, Engineering and Technology Research. 2015; **4**(5):1543-1546 ISSN: 2278-7798

[30] Moliner R, Tanda R. Herramienta para la sintonía robusta de controladores PI/PID de dos grados de libertad. Revista Iberoamericana de Automática e Informática Industrial RIAI. 2016;**13**(1):22-31. DOI: 10.1016/j.riai.2015.05.003

[31] Guesmi K, Essounbouli N, Hamzaoui A. Systematic design approach of fuzzy PID stabilizer for DC-DC converters. Energy Conversion and Management. 2008;**49**(10):2880-2889. DOI: 10.1016/j.enconman.2008.03.012

[32] Wen Y, Trescases O. Non-linear control of current-mode buck converter with an optimally scaled auxiliary phase. In: Industrial Technology (ICIT), IEEE International Conference on; March 2010. p. 783-788. DOI:10.1109/ICIT.2010.5472628

[33] Barrado A, Lazaro A, Pleite J, Vazquez R, Vazquez J, Olias E. Linear-non-linear control (LnLc) for DC-DC buck converters: Stability and transient response analysis. In: Applied Power Electronics Conference and Exposition (APEC'04), Nineteenth Annual IEEE; February 2004. p. 1329-1335. DOI:10.1109/APEC.2004.1295995

[34] Eker I, Torun Y. Fuzzy logic control to be conventional method. Energy conversion and management. 2006;**47**(4):377-394. DOI: 10.1016/j.enconman.2005.05.008

[35] Gao D, Jin Z, Lu Q. Energy management strategy based on fuzzy logic for a fuel cell hybrid bus. Journal of Power Sources. 2008;**185**(1):311-317. DOI: 10.1016/j.jpowsour.2008.06.083

[36] Qi W, Li S, Tan SC, Hui SR. Parabolic-modulated sliding mode voltage control of buck converter. IEEE Transactions on Industrial Electronics. 2017;**65**(1):844-854. DOI: 10.1109/TIE.2017.2716859

[37] Trejos A, Gonzalez D, Ramos-Paja CA. Modeling of step-up grid-connected photovoltaic systems for control purposes. Energies. 2012;**5**(6):1900-1926. DOI: 10.3390/en5061900

[38] Bacha S, Munteanu I, Bratcu AI. Power electronic converters modeling and control. Adv. Textb. Control Signal Process. 2014;**454**:454. DOI: 10.1007/978-1-4471-5478-5