International Journal of Control, Automation and Systems 14(6) (2016) 1607-1617
http://dx.doi.org/10.1007/s12555-015-0173-1
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Robust RST Control Design based on Multi-Objective Particle Swarm Optimization Approach

Riadh Madiouni, Soufiene Bouallègue*, Joseph Haggège, and Patrick Siarry

Abstract: In this paper, a novel method for the digital two-Degrees-Of-Freedom (2DOF) controller design, called canonical RST structure, is proposed and successfully implemented based on a Multi-Objective Particle Swarm Optimization (MOPSO) approach. This is a polynomial control structure allowing independently the regulation and the tracking of discrete-time systems. An application to the variable speed control of an electrical DC Drive is investigated. The RST design and tuning problem is formulated as a multi-objective optimization problem. The proposed MOPSO algorithm which is based on the Pareto dominance is used to identify the non-dominated solutions. This approach used the leader selection strategy that is called a geographically-based system. In addition, the adaptive grid method is used to produce well-distributed Pareto fronts in the multi-objective formalism. The well known NSGA-II and the proposed MOPSO algorithms are evaluated and compared with each other in terms of several performance metrics in order to show the superiority and the effectiveness of the proposed method. Simulation results demonstrate the advantages of the MOPSO-tuned RST control structure in terms of performance and robustness.

Keywords: DC drive, digital RST controller, external archive, multi-objective design, NSGA-II, particle swarm optimization, Pareto front, robustness constraints.

1. INTRODUCTION

The synthesis of robust control is currently a developing research theme. Several methods of synthesis are being developed for the design and analysis of robust control. The two-Degrees-Of-Freedom (2DOF) digital design, called RST controller, is a polynomial control structure allowing independently the regulation and the tracking of discrete-time systems. This canonical structure of digital controllers is based on the calculation of three polynomials R, S and T which allow the poles placement of the closed-loop. It is a robust and effective control strategy that is widely used in industrial applications [1–5]. In this formalism, the most useful method to synthesize the digital RST controllers is based on the well-known closed-loop poles placement [2]. In this design case, Sylvester's method can be used to determine the search parameters of the RST controller by resolving a polynomial equation which will ensure the desired closed-loop poles. The major drawback of this synthesis technique is the choice of the closed-loop poles that is usually difficult and becomes more complicated with the complexity of the controlled plant.

Up to now, there has been no clear and systematic procedure in the literature to guide the closed-loop poles placement choice in the RST control design. To overcome this problem, several techniques have been proposed in the literature. The two classical methods, based on pole placement combined with the sensitivity functions shaping, have been proposed by Landau and Karimi [3]. The first approach combines the pole placement and the calibration of the sensitivity functions, using the fixed parts in the controller. It iteratively adjusts the sensitivity functions in the frequency domain where it is necessary and uses robustness templates to obtain the placement of the poles. In the second method, the synthesis problem has been transferred to $\mathcal{H}_\infty$ optimal control via a new interpretation of the weighting filters, as the reverse of the desired sensitivity function. In this approach, the weighting filter selection is carried out automatically by an optimization program. These iterative and trial error-based methods are not suitable for the complex systems. In [4], Rotella et al. proposed a new approach for digital RST controller synthesis based on the flatness property of dynamical systems. Such a flatness-based method is also applied for a DC drive real-time control in [1]. In [6], another design

Manuscript received April 23, 2015; revised August 29, 2015, October 9, 2015, and October 31, 2015; accepted December 9, 2015. Recommended by Associate Editor DaeEun Kim under the direction of Editor Euntai Kim.

Riadh Madiouni and Patrick Siarry are with the Images, Signals and Intelligent Systems Laboratory, LiSSi-EA-3956, University of Paris-Est Créteil, 122 rue Paul Armangot, 94400 Vitry sur Seine, France (e-mails: riadh.madiouni@univ-paris-est.fr, siarry@univ-paris12.fr). Soufiene Bouallègue and Joseph Haggège are with the Research Laboratory in Automatic Control (LARA), National Engineering School of Tunis (ENIT), BP 37, Le Belvédère, 1002 Tunis, University of Tunis El Manar, Tunisia (e-mails: soufiene.bouallegue@issig.rnu.tn, joseph.haggege@enit.rnu.tn).
* Corresponding author.

approach, using convex optimization was proposed by Galdos et al. The performance specifications given as the infinity norm of the weighted sensitivity functions are represented as convex constraints in the Nyquist diagram. Unfortunately, this method is time-consuming, difficult to implement and becomes ineffective for non-convex problems.

In view of these difficulties, proposing a systematic and easy procedure for the RST synthesis problem is an important and interesting task in this area. To deal with these synthesis problems of RST controller we proposed in a promising solution which gave good results. The recourse to the optimization theory, such as the Particle Swarm Optimization (PSO) technique [7, 8], was proposed and validated. The synthesis problem of RST controller is formulated as a constrained optimization problem.

The developments presented in this paper are an extension of our previous work in [5] that deals with the RST synthesis problem within a mono-objective framework. The aim contribution of this paper is to formulate and solving the 2DOF RST problem using a developed Multi-Objective Particle Swarm Optimization (MOPSO) algorithm with a constraints handling mechanism. Different optimization criteria, such as the IAE (Integral of the Absolute Error) and MO (Maximum Overshoot) index, are considered under various control constraints. Controller's robustness specifications on disturbance rejection and stability margins guarantee, classically analyzed after design stage, become inequality type constraints for the formulated RST problem optimization and can be held into account since the synthesis phase. The proposed MO-PSO algorithm is based on rules of dominance and a controller external archive. All the best obtained solutions are stored and will be represented by an adaptive grid to form the Pareto front. The robustness of the RST-tuned MOPSO algorithm is verified by numerical simulations when varying their main control parameters. This implemented algorithm was firstly verified on a set of literature test functions and then applied to resolve the formulated RST optimization-based control problem. The MOPSO algorithm is also compared with another similar one (NSGA-II) that is widely used in multi-objective optimization formalism. Several statistical analysis and popular performance assessment metrics were carried out for its validation and effectiveness.

The remainder of this paper is organized as follows. In Section 2, the formulation of the optimization-based control problem is presented. In Section 3, the developed MOPSO is described and validated on the known test functions from the literature. In Section 4, we apply the proposed control approach to an electrical DC drive benchmark control. All MOPSO-based simulation results are compared and discussed with those obtained by the NSGA-II-based approach in order to show the effectiveness and superiority of the proposed strategy.

2. RST DESIGN AND TUNING PROBLEM FORMULATION

2.1. Digital RST controller structure

In this study, the discrete-time model of the plant to be controlled is described in the time-domain by the following transfer function:

$$H\left(q^{-1}\right) = \frac{y_k}{u_k} = \frac{q^{-z}B\left(q^{-1}\right)}{A\left(q^{-1}\right)}, \tag{1}$$

where q^{-1} is the backward shift operator, z is the integer number of sampling period contained in the time-delay of the plant, k is the normalized discrete time and corresponds to the discrete time divided by the sampling period T_s, u_k and y_k are the discrete plant input and output, respectively.

The terms $A\left(q^{-1}\right)$ and $B\left(q^{-1}\right)$, assumed to be co-prime polynomials, are respectively the denominator and the numerator of the transfer function, defined as:

$$A\left(q^{-1}\right) = 1 + a_1 q^{-1} + \ldots + a_{n_A} q^{-n_A}, \tag{2}$$

$$B\left(q^{-1}\right) = b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{n_B} q^{-n_B}. \tag{3}$$

The canonical structure of the digital RST controller is shown in Fig. 1. This structure has two-degrees-of-freedom. The digital polynomials R and S are designed in order to achieve the desired regulation performance. The polynomial T is designed afterwards in order to achieve the desired tracking one [2, 3]. This structure allows the achievement of different levels of performance in tracking and regulation.

The classical RST control law, is given as follows:

$$S\left(q^{-1}\right) u_k = T\left(q^{-1}\right) y^*_{k+z+1} - R\left(q^{-1}\right) y_k. \tag{4}$$

The polynomials $R\left(q^{-1}\right)$, $S\left(q^{-1}\right)$ and $T\left(q^{-1}\right)$ have the following forms:

$$R\left(q^{-1}\right) = r_0 + r_1 q^{-1} + \ldots + r_{n_R} q^{-n_R}, \tag{5}$$

$$S\left(q^{-1}\right) = s_0 + s_1 q^{-1} + \ldots + s_{n_S} q^{-n_S}, \tag{6}$$

$$T\left(q^{-1}\right) = t_0 + t_1 q^{-1} + \ldots + t_{n_T} q^{-n_T}. \tag{7}$$

The desired tracking trajectory y^*_{k+z+1} may be generated by the following tracking reference model:

$$y^*_{k+z+1} = \frac{B_m\left(q^{-1}\right)}{A_m\left(q^{-1}\right)} r_k, \tag{8}$$

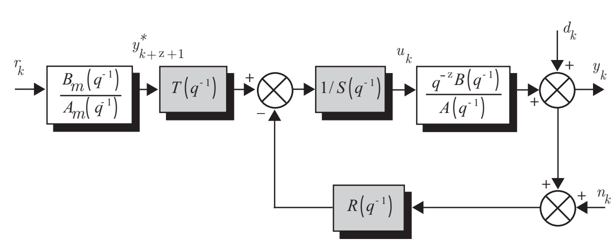

Fig. 1. Canonical RST structure of the controller.

where r_k is the reference trajectory.

Hence, the closed-loop poles have been defined by the following polynomial equation :

$$P(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-z}B(q^{-1})R(q^{-1}). \tag{9}$$

2.2. Robustness and performance constraints

For reasons of robustness and performance specifications such as disturbance rejection and noise attenuation, the polynomials $R\left(q^{-1}\right)$ and $S\left(q^{-1}\right)$ contain some fixed parts which are specified before solving (9). In [3], it is shown that the pre-specified polynomial $H_R\left(q^{-1}\right)$ is used to eliminate the high frequency noises on the input signal. The polynomial $H_S\left(q^{-1}\right)$ performs polynomial to allow the rejection of static disturbances on the output signal. These polynomials are described as follows:

$$H_S\left(q^{-1}\right) = 1 - q^{-1};\ H_R\left(q^{-1}\right) = 1 + q^{-1}. \tag{10}$$

While taking into account these pre-specified parts, the new polynomials of the MOPSO-tuned RST controller become:

$$\hat{S}\left(q^{-1}\right) = H_S\left(q^{-1}\right)S\left(q^{-1}\right), \tag{11}$$

$$\hat{R}\left(q^{-1}\right) = H_R\left(q^{-1}\right)R\left(q^{-1}\right). \tag{12}$$

In order to ensure a delay margin $\Delta\tau = T_s$, the modulus of the output sensitivity function $S_{yd}\left(q^{-1}\right)$ must lie between the upper and lower templates as follows :

$$\begin{aligned} \left|W^{-1}\right|_{\inf} &= 1 - \left|1 - q^{-1}\right|^{-1} < \left|S_{yd}(q^{-1})\right|, \\ \left|W^{-1}\right|_{\sup} &= 1 + \left|1 - q^{-1}\right|^{-1} > \left|S_{yd}(q^{-1})\right|. \end{aligned} \tag{13}$$

In (13), the variables $\left|W^{-1}\right|_{\sup}$ and $\left|W^{-1}\right|_{\inf}$ represent the upper and lower robustness bounds respectively in the frequency domain for the modulus of the output sensitivity function $\left|S_{yd}\left(q^{-1}\right)\right|$. These two analytical expressions represent later the inequality type constraints $g_m(\boldsymbol{x})$ of the formulated optimization problem (14). Since the sensitivity function $\left|S_{yd}\left(q^{-1}\right)\right|$ is a robustness index of the designed RST controller in terms of disturbance rejection and modulus and delay margins guarantee, the frequency shape of this closed-loop transfer function must remain inside of the envelope formed by these upper and lower bounds. Graphically, these discreet-time filters behave like frequency robustness templates.

2.3. Multi-objective problem formulation

A multi-objective optimization problem involves the simultaneous satisfaction of two or more objective functions [9–14]. In this section, the RST design is formulated as a constrained multi-objective optimization problem which is solved using the proposed MOPSO-based technique. Such a constrained optimization problem can be mathematically described as:

$$\begin{aligned} &\underset{\boldsymbol{x}\in\mathbb{D}\subseteq\mathbb{R}^n}{\text{minimize}}\quad f_p(\boldsymbol{x}),\quad p = 1,\ 2, \ldots, P \\ &\text{subject to :} \\ &h_l(\boldsymbol{x}) = 0,\quad l = 1,\ 2, \ldots, L \\ &g_m(\boldsymbol{x}) \leq 0,\quad m = 1, 2, \ldots, M, \end{aligned} \tag{14}$$

where $f_p(\boldsymbol{x})$, $h_l(\boldsymbol{x})$ and $g_m(\boldsymbol{x})$ are functions of the design vector $\boldsymbol{x} \in \mathbb{R}^n$.

The multi-objective optimization-based RST synthesis problem consists in finding the optimum decision variables $\boldsymbol{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$, which represent the RST controller parameters grouped in the design vector given as follows:

$$\boldsymbol{x} = \left[\ s_0,\ s_1, \ldots,\ s_{n_S},\ r_0,\ r_1, \ldots,\ r_{n_R}\ \right]^T. \tag{15}$$

These decision variables minimize the two considered cost functions, such as the MO and IAE criteria, given respectively as follows:

$$f_{MO}(\boldsymbol{x}, t) = \frac{y_{\max}(\boldsymbol{x}, t) - y(\boldsymbol{x}, t \to +\infty)}{y(\boldsymbol{x}, t \to +\infty)}, \tag{16}$$

$$f_{IAE}(\boldsymbol{x}, t) = \int_0^{+\infty} |\varepsilon(\boldsymbol{x}, t)|\, dt, \tag{17}$$

where ε denotes the continuous-time error tracking of the closed-loop and $y_{\max}$ the maximum value of the plant output.

In this design case, we denote that only the regulation performance is considered, i.e., $R\left(q^{-1}\right)$ and $S\left(q^{-1}\right)$ polynomials are optimized. Hence, the digital filter $T\left(q^{-1}\right)$ can be designed afterwards as follows [2, 3]:

$$T\left(q^{-1}\right) = t_0 = \frac{P(1)}{B(1)}. \tag{18}$$

Indeed, the difficult stage in the 2DOF RST controller design is the calculation of R and S polynomials that define the regulation dynamics of this control approach. This calculation is based on poles placement of the closed-loop system, usually difficult and delicate. However, the tracking problem resolution in RST design is systematic since we have the mathematical formulas to calculate this dynamics. The polynomial $T(q^{-1})$ of tracking behavior is expressed as a function of open loop model poles (polynomial $B(q^{-1})$) and closed-loop ones ($P(q^{-1})$). For this purpose, we chose to reduce the size of the formulated optimization problem for only regulation stage, and nothing prevents to also treat the tracking problem. In this case, the optimization problem becomes larger and other decision variables (coefficients t_i of the tracking polynomial $T(q^{-1})$ will be added in the vector $\boldsymbol{x}$ of Equation (15).

3. PROPOSED MULTI-OBJECTIVE PSO APPROACH

3.1. MOPSO algorithm implementation

The original PSO algorithm was introduced back in 1995 by Eberhart and Kennedy [7, 8]. The basic concept of PSO is to have a set of solutions called particles finding in a search space. Each particle moves in the search space with an adaptable velocity, while retaining in the memory the best position it has ever visited [15, 16]. This advanced meta-heuristic technique have been applied in variours control approaches such as the fuzzy and structured $\mathcal{H}_\infty$ designs [17, 18].

In a search space $\mathbb{D}$, the particle i of the swarm is modeled by its position $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and by its velocity $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. This particle remembers the best position through which it has already passed, denoted $p_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The best position reached by any particle swarm is denoted as $g_i = (g_{i1}, g_{i2}, \ldots, g_{iD})$. At time t and dimension j the velocity is calculated from (19):

$$\begin{aligned} v_{ij}(t) =& w v_{ij}(t-1) + c_1 \chi_{1,t} \left(p_{ij}(t-1) - x_{ij}(t-1) \right) \\ &+ c_2 \chi_{2,t} \left(g_j(t-1) - x_{ij}(t-1) \right), \end{aligned} \tag{19}$$

where w is the inertia factor, c_1 and c_2 are the cognitive and the social scaling factors respectively, $\chi_{1,t}$ and $\chi_{2,t}$ are random numbers uniformly distributed in $[\![0,1]\!]$.

The position at the time t of the particle i is defined by equation (20):

$$x_{ij}(t) = x_{ij}(t-1) + v_{ij}(t). \tag{20}$$

Solving a multi-objective problem is about selecting the non-dominated positions found by the algorithm during its execution. At the end of execution, the chosen solutions need to be non-dominated compared to all the positions reached by the particles in successive iterations. From a mathematical point of view, the sense of Pareto optimality can be expressed in terms of dominance [19–21].

The principle of the particle swarm optimization for multi-objective problems is based on dominance and neighborhood manipulation in space criteria [22–26]. The comparison strategy for the solutions through dominance and neighborhood manipulation is essential to the construction of the archive. Therefore, we used an archive controller to select the solutions to be added to the archive. The main objective of the external storage space is to keep a history of non-dominated vectors founded along the search process. The decision process is as follows: the non-dominated vectors founded in each iteration of the main population of the algorithm are compared. As regards the contents of the external archive, it will be empty at the beginning and it accepts the first solution found. If the new solution is non-dominated then the solution in the archive will be rejected and inversely. The meta-heuristics-based multi-objective optimization concepts are extensively used in the control design theory [27–30].

The basic idea is to use an external archive to store all the non-dominated solutions, hence we propose the use of an adaptive grid to plot the Pareto front which is a set of the final solutions. The grid is composed of hypercubes which contains a number of particles. If a particle exceeds

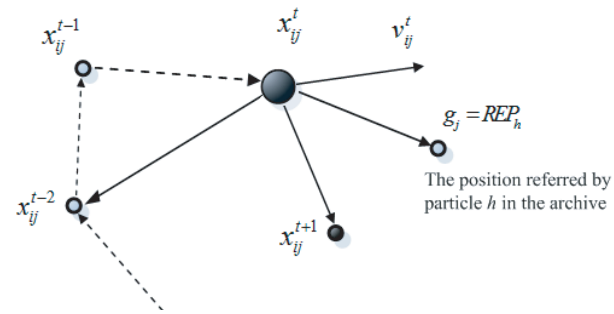

Fig. 2. Velocity and position update.

the maximum limit, the grid is recalculated again [22]. In the proposed MOPSO, a particle moves toward one of known Pareto solutions and searches around the solution exploitatively. Hence, the principle of a particle displacement in the swarm is graphically shown in Fig. 2.

The variable Rep_h is selected from the archive so the equation (21) becomes:

$$\begin{aligned} v_{ij}(t) =& w v_{ij}(t-1) + c_1 \chi_{1,t} \left(p_{ij}(t-1) - x_{ij}(t-1) \right) \\ &+ c_2 \chi_{2,t} \left(Rep_h(t-1) - x_{ij}(t-1) \right). \end{aligned} \tag{21}$$

The method of selecting a guide to the evolution of particles is strongly related to the archive [23]. The selected guides determine the velocity of convergence of the swarm to a set of satisfactory solutions. The approach adopted a strategy for the choice of leader Rep_h. The roulettewheel selection is applied to select the hypercubes. Each hypercubes contains n particles, hence we took randomly a particle. The main algorithm proposed to the MOPSO approach is described as follows:

1) Randomly initialize the position and velocity of particles in D-dimensional space.
2) Evaluate each particle.
3) Store the non-dominated solutions or the ones founded in the archive.
4) Produce the hypercubes of the search space.
5) Memorize and store the best positions of particles in the archive *Pbest* to calculate the next positions.
6) Calculate the speed and the position of each particle.
7) Evaluate each particle. Along their research more particles beyond the boundaries. They are returned into the search space, then they will be generated.
8) Update the coordinates of the particles in all the hypercubes and updated the archive that will be full at the t^{th} iteration, then the particles in the less dense areas are prioritized. The capacity of the archive (Arch.) is already defined in advance.
9) Update the *Pbest* solution based on the Pareto dominance.

3.2. MOPSO algorithm improvements

The MOPSO algorithm, proposed in this paper for synthesis and tuning of polynomial RST controllers, is improved compared to the conventional PSO one, initially developed by R. Eberhart and J. Kennedy for the single-objective optimization. Our improvements are especially related to the following items:

- Constraints handling mechanism;
- Leader roulette-wheel selection strategy;
- Adaptive grid method for the external archive;
- Inertia factor decreasing linearly with iterations.

The original PSO algorithm is formulated as an unconstrained optimizer. Several techniques have been proposed to deal with constraints. One useful approach is by augmenting the cost function of problem (14) with penalties proportional to the degree of constraint infeasibility. In this paper, the following external static penalty technique is used to improve the proposed MOPSO algorithm:

$$F_p(\boldsymbol{x}) = f_p(\boldsymbol{x}) + \sum_{m=1}^{M} \lambda_m \max^2\{0, g_m(\boldsymbol{x})\}, \tag{22}$$

where λ_m is a prescribed scaling penalty parameters and M the number of the problem inequality constraints as depicted in (14).

In a typical optimization procedure, the scaling parameters of (22) will be linearly increased at each iteration step so constraints are gradually enforced. The quality of the solutions will directly depend on the value of these scaling parameters. For simplicity purposes and to save the computational time algorithm, great and constant scaling penalty parameters λ_m, equal to 10^4, are used in this paper.

The proposed MOPSO algorithm is implemented with the concept of leader selection strategy. Since the solution of a multi-objective problem consists of a set of equally good solutions, it is evident that the concept of leader, traditionally adopted in PSO, has to be changed. The selection of a leader is a key component in multi-objective PSO paradigm. This MOPSO algorithm is based on the idea of having an external archive in which every particle will deposit its flight experiences after each flight cycle. The updates of the external archive are performed considering a geographically-based system defined in terms of the objective function values of each particle. The explored search space is divided on hyper-cubes. Each one of these hyper-cubes receives a fitness value based on the number of particles it contains. Thus, in order to select a leader for each particle of the swarm, a roulette-wheel selection using these fitness values is first applied to select the hyper-cube from which the leader will be taken. Once a hyper-cube has been selected, the leader is randomly chosen.

To produce well-distributed Pareto fronts, our approach uses a variation of the adaptive grid concept. The basic idea is to use an external archive to store all non-dominated solutions that are with respect to the contents of the archive. In the archive, objective function space is divided into several regions. Note that if the individual inserted into the external population lies outside the current bounds of the grid, then the grid has to be recalculated and each individual within it has to be relocated. The adaptive grid is a space formed by hyper-cubes. Such hyper-cubes have as many components as objective functions. Each hyper-cube can be interpreted as a geographical region that contains a number of individuals.

On the other hand, and in order to improve the proposed MOPSO algorithm in terms of increasing its exploration/exploitation capacity, we used the following adaptive inertia factor variation [14, 15]:

$$w(t) = w_{\max} - (w_{\max} - w_{\min})\frac{t}{t_{\max}}, \tag{23}$$

where $w_{\max} = 0.9$, $w_{\min} = 0.4$ and $t_{\max}$ is the maximum iteration number.

3.3. Validation of the proposed MOPSO algorithm

The algorithms have been coded in MATLAB 7.8 and executed on a PC computer with Core 2 Duo-2.20 GHz CPU and 2.00 GB RAM. In order to study the developed MOPSO approach and to compare it with an other algorithm, we use the three popular performance assessment metrics given by (24), (25) and (26).

$$GD = \frac{1}{n_D}\sqrt{\sum_{i=1}^{n_D} d_i^2}, \tag{24}$$

$$SP = \sqrt{\frac{1}{n_D - 1}\sum_{i=1}^{n_D}\left(\bar{d} - d_i\right)^2}, \tag{25}$$

$$ER = \frac{1}{n_V}\sum_{i=1}^{n} e_i, \tag{26}$$

where $d_i = \min_j \sum_{p=1}^{P} \left|f_p^i - f_p^j\right|$, $j = 1, 2, \ldots, n$ represent the Euclidian distance between each of the solutions found and the closest one in the Pareto front. In this expression, the variables f_p^i and f_p^j represent the values of the p^{th} objective functions of the considered multi-objective optimization problem (14) needed to calculate this distance. P is the total number of the problem objective functions.

The Generational Distance (GD) measures the distance between the elements of the set of non-dominated solutions found and the elements of the Pareto optimal set. The Spacing (SP) measures the distribution of the solutions in the research space. The Error Ratio (ER) calculates the percentage of the non-dominated solutions found that are not in the optimal Pareto front [10, 19, 20]. In these equations, d_i denotes the Euclidean distance between each of

Table 1. MOPSO and NSGA-II algorithms specifications.

	Pop.	Iter.	Arch.	Parameters
MOPSO	100	100	100	$c_1 = 1.5$, $c_2 = 1.7$
NSGA-II	100	100	–	nGrid=30

Table 2. The algorithm's computation times in sec.

	F1	F2	F3	F4
MOPSO	74.451	37.801	90.771	63.412
NSGA-II	161.124	147.147	162.184	104.356

the solutions found and the nearest element of the Pareto optimal set, n_D is the number of non-dominated solutions found, $\overline{d}$ is the mean value of all d_i and n_V is the number of vectors in the current set of non-dominated vectors. The parameter e_i is equal to 0 if i is a member of the Pareto optimal and $e_i = 1$ otherwise. The idea is to test the ability of the proposed approach to identify the optimal Pareto front and a quantitative evaluation of its performance. For that, three points are normally considered [14, 21, 24]:

- reducing the distance between Pareto optimal front and the one produced by the developed algorithm,
- maximizing the spread of solutions,
- maximizing the number of elements of the Pareto front found by the MOPSO algorithm.

In the concept of multi-objective optimization, several test functions are used to compare the effectiveness of the proposed MOPSO and NSGA-II metaheuristic algorithms. In this paper, four benchmark problems are selected from the literature [11, 13, 14] and described in the APPENDIX. The parameters of Table 1 are used to verify, analysis and compare the MOPSO and NSGA-II algorithms. The used Non Sorting Genetic Algorithm II (NSGA-II) algorithm, proposed by, Deb [11, 12], inspires the same concepts of genetic algorithm. This technique uses a sorting method based on non-dominance to construct the Pareto front. It uses a comparison operator based on the calculation of the crowding-distance.

Table 2 gives the computation times of the proposed algorithms for the considered test functions. As shown in Figs. 3, 4, 5 and 6, as well as the results of Tables 3, 4 and 5, neither of the two algorithms converge on the optimal Pareto front for the considered test functions.

The value of ER of MOPSO algorithm is less than the one of NSGA-II algorithm. Regarding the GD metric, we can conclude that the obtained value is lower in MOPSO technique. So, we observe that MOPSO gets better performance for the used test functions in this metric but the value of SP of MOPSO is greater than the one of NSGA-II. We can conclude that the algorithm MOPSO obtained the best solutions with respect to ER, GD and SP metrics for the considered test functions. The proposed algorithm returns a greater number of solutions belonging to

Table 3. Obtained GD metric of the test functions.

Functions F1, F2, F3 and F4	MOPSO	NSGA-II
Best	$8.25\ 10^{-5}$	0.000129
Worst	0.000124	0.135470
Average	0.000101	0.013578
Std. Dev.	$2.34\ 10^{-5}$	0.031450
Best	0.006450	0.005610
Worst	0.008970	0.100100
Average	0.007450	0.028600
Std. Dev.	0.000470	0.021200
Best	0.000330	0.000600
Worst	0.152100	0.200100
Average	0.029900	0.411000
Std. Dev.	0.051200	0.052100
Best	0.001200	0.002120
Worst	0.334000	0.538000
Average	0.024100	0.072200
Std. Dev.	0.101000	0.121400

Table 4. Obtained SP metric of the test functions.

Functions F1, F2, F3 and F4	MOPSO	NSGA-II
Best	0.006670	0.000120
Worst	0.016120	0.013540
Average	0.010110	0.002870
Std. Dev.	0.001824	0.002684
Best	0.057100	0.016700
Worst	0.112000	0.554000
Average	0.091200	0.025100
Std. Dev.	0.015400	0.015300
Best	0.034500	0.021000
Worst	0.421000	0.032800
Average	0.077100	0.031200
Std. Dev.	0.102000	0.008510
Best	0.036100	0.000930
Worst	0.502000	1.238000
Average	0.092100	0.063100
Std. Dev.	0.101000	0.215000

the true Pareto front and the minimum GD. In addition, the MOPSO has a better spread of solutions compared to the given NSGA-II ones as well as a lower computation time as shown in Table 2.

4. APPLICATION TO DC DRIVE BENCHMARK SPEED CONTROL

4.1. Plant description

The considered benchmark is a 250 watts electrical DC drive [5]. The speed of the machine rotation is 3000 rpm at 180 volts DC armature voltage. The motor is supplied by an AC-DC power converter. The model parameters are given with their associated uncertainty bounds of $\pm 50\%$

Table 5. Obtained ER metric of the test functions.

Functions F1, F2, F3 and F4	MOPSO	NSGA-II
Best	0.150000	0.010000
Worst	0.390000	0.97100
Average	0.286000	0.323000
Std. Dev.	0.088100	0.364700
Best	0.130000	0.055000
Worst	0.323000	1.010000
Average	0.231700	0.461000
Std. Dev.	0.038800	0.377400
Best	0.001000	0.010000
Worst	1.010000	1.010000
Average	0.247000	0.384000
Std. Dev.	0.383000	0.411000
Best	0.071000	0.680000
Worst	0.190000	0.940000
Average	0.121000	0.781000
Std. Dev.	0.388000	0.051100

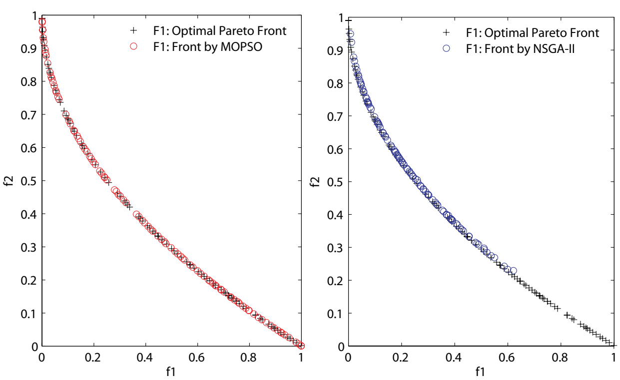

Fig. 3. Pareto fronts produced by the algorithms for F1.

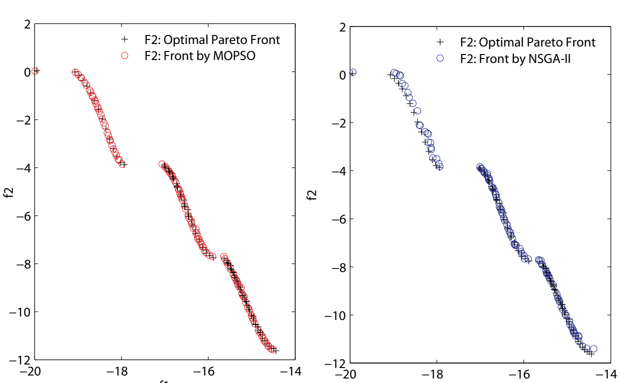

Fig. 4. Pareto fronts produced by the algorithms for F2.

as used in [1, 18]. We denote by $G_0 = 0.05$ the static gain of the plant, $\tau_m = 300\ ms$ and $\tau_e = 14\ ms$ are the mechanical and electrical constant times, respectively. The discrete-time model is obtained by sampling of this continuous second order transfer function where $T_s = 0.01$ sec.

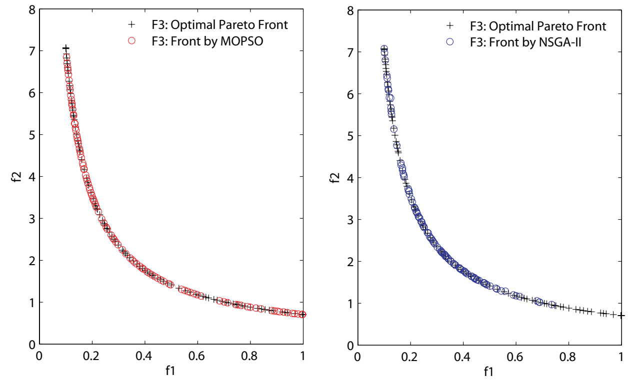

Fig. 5. Pareto fronts produced by the algorithms for F3.

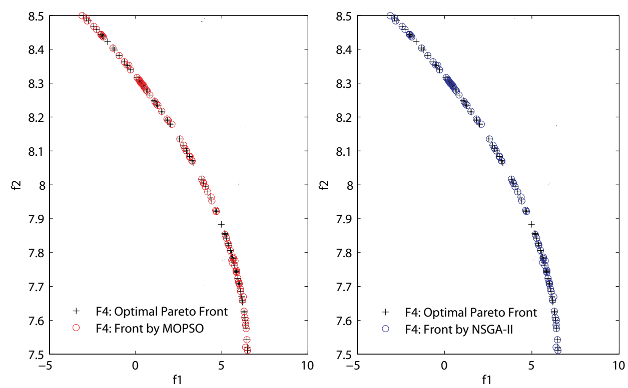

Fig. 6. Pareto fronts produced by the algorithms for F4.

4.2. Simulation results

The RST synthesis problem can be solved by the constrained MOPSO-based approach given as follows:

$$\begin{cases} \underset{\boldsymbol{x}=[s_1,r_0,r_1]^T\in\mathbb{R}^3}{minimize} \quad (f_1(\boldsymbol{x}), f_2(\boldsymbol{x})) \\ \text{subject to :} \\ \qquad g_1(\boldsymbol{x}) = \left|S_{yd}\left(q^{-1},\boldsymbol{x}\right)\right| - \left|W^{-1}\right|_{\sup} \le 0 \\ \qquad g_2(\boldsymbol{x}) = \left|W^{-1}\right|_{\inf} - \left|S_{yd}\left(q^{-1},\boldsymbol{x}\right)\right| \le 0, \end{cases} \tag{27}$$

where g_1 and g_2 are the inequality constraints of the formulated RST design and tuning problem.

Remember that the output sensitivity function $\left|S_{yd}\left(q^{-1},\boldsymbol{x}\right)\right|$ of (27) denotes the closed-loop transfer function, between the output system y_k an the disturbance input d_k, explicitly defined in (28). The variables $\left|W^{-1}\right|_{\sup}$ and $\left|W^{-1}\right|_{\inf}$ are the modulus bounds of this transfer function in the frequency responses domain. These filters, as given in (13), represent the upper and lower robustness templates for the sensitivity modulus shaping.

$$S_{yd} = \frac{A\left(q^{-1}\right)S\left(q^{-1},\boldsymbol{x}\right)}{A\left(q^{-1}\right)S\left(q^{-1},\boldsymbol{x}\right) + q^{-z}B\left(q^{-1}\right)R\left(q^{-1},\boldsymbol{x}\right)}. \tag{28}$$

In this multi-objective optimization RST control problem, the IAE and MO criteria are considered as the objective functions f_1 and f_2. Fig. 7 is the result of the two

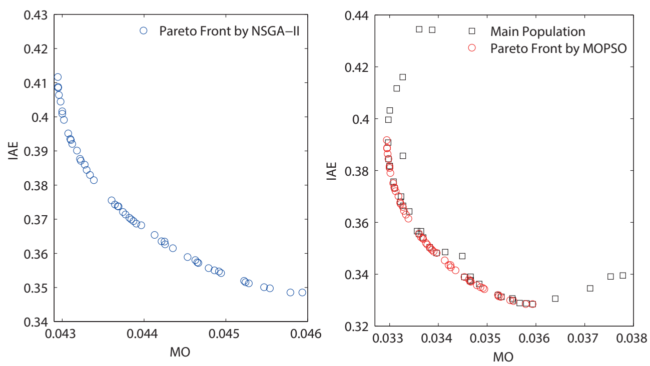

Fig. 7. Pareto fronts for the RST tuning problem.

Table 6. Optimization results for the RST tuning problem.

	s_1	r_0	r_1	IAE	MO
MOPSO	0.504	0.044	0.455	0.363	0.035
NSGA-II	0.371	0.038	0.461	0.395	0.045

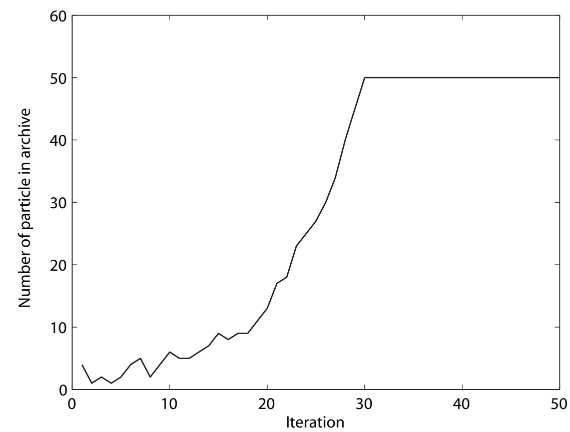

Fig. 8. Evolution of particles number in the archive.

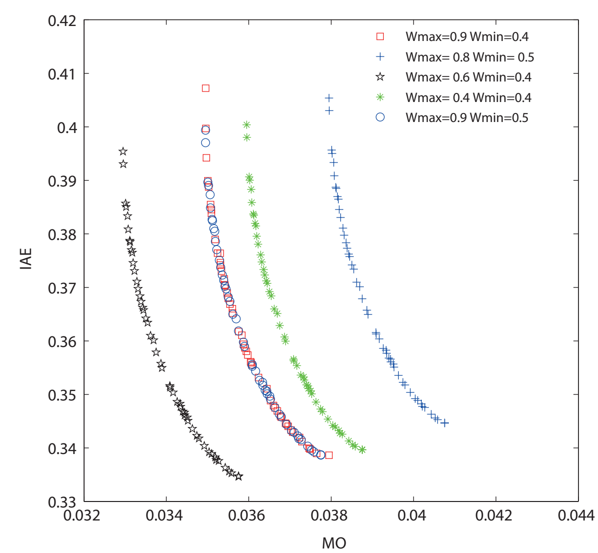

Fig. 9. Robustness of the MOPSO algorithm.

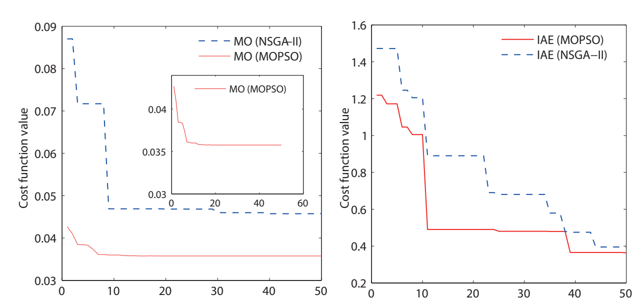

Fig. 10. Convergence properties of the algorithms.

algorithms for the RST synthesis problem (27). The solution placed in the hypercubes have a coordinates (x_{grid}, y_{grid}) which are the values of the two objective functions MO and IAE respectively.

For the implemented MOPSO algorithm, we chose a population size equal to 50. The maximum iteration number is fixed as 50. The archive size is equal to 50. The PSO cognitive and social parameters c_1 and c_2 are kept as 1.5 and 1.7, respectively. The optimized RST coefficients obtained in the Mean case are given in Table 6. Remember that all the solutions in the obtained Pareto front related to objectives f_1 and f_2 are non-dominated and can be considered as potential solutions for the RST optimization problem. So, we can choose arbitrarily any solution among them as an optimum of the formulated multi-objective optimization problem.

The objective of the external archive is to store all non-dominated solutions when the number of non-dominated solutions is equal to the maximum archive size. The new solution founded will be compared 50 times with the existing solutions in the archive. Fig. 8 shows the change in the non dominated particle number in the archive, which affected its maximum capacity in an iteration interval between 30 and 50. In our case and after testing, we achieve the optimal result when we used the mentioned MOPSO and NSGA-II parameters. The robustness of the proposed MOPSO algorithm convergence, under variation of the cognitive and inertia factor parameters, is analysed on the basis of numerical simulations as shown in Fig. 9.

The comparison of MOPSO and NSGA-II algorithms shows the performance of the proposed one, as presented in Fig. 10. The MOPSO algorithm converges to the better solution after a number of iterations less than the number of generations for the NSGA-II one. Hence, the convergence speed of the proposed approach and the quality of the solutions is proved relative to the NSGA-II algorithm. In order to get some statistical data on the quality of optimization results, it is necessary to run the algorithm several times. The algorithm is executed 20 times and feasible solutions were found in 95% of trials and within an acceptable CPU computation time.

As explained in [1, 3], the robustness of the designed RST controller is guaranteed as can be seen in Fig. 11

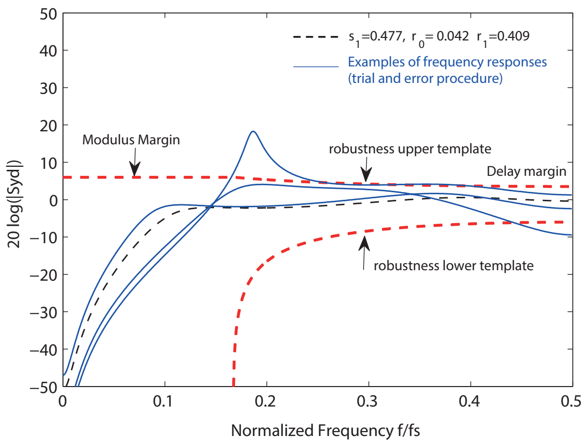

Fig. 11. Modulus margin and static disturbance rejection.

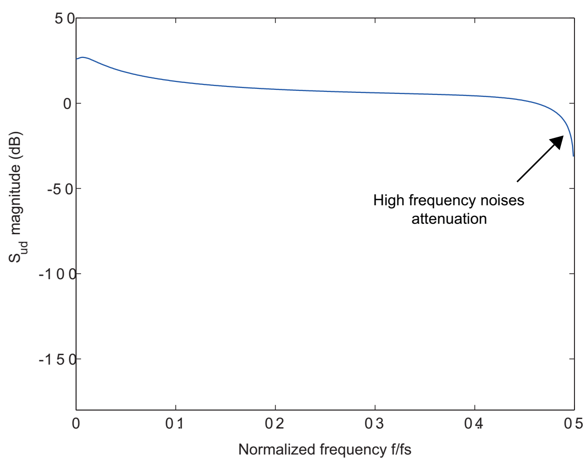

Fig. 12. High frequency noises elimination.

and Fig. 12. Indeed, the module of the output sensitivity functions S_{yd} remains inside the predefined template and that of the input sensitivity function S_{ud} presents attenuation in high frequencies. This result leads also to obtaining high time-domain performances of the proposed MOPSO-tuned RST controller structure. These simulation results show that the actual speed of DC motor tracks the desired trajectory with high performance. The tracking error is very small in the transient regime and equal to zero in steady-state. Fig. 13 shows the system step responses for the MOPSO, NSGA-II and Sylvester-based algorithms, which are plotted with the mean values of the optimized parameters.

The proposed MOPSO-based approach produces better responses than that obtained using the other methods [1,5]. The multi-objective optimization well improves performance and design stage of the polynomial RST controllers. The practical implementation of such a digital MOPSO-tuned controller becomes easy while programming the obtained control law as a recurrent equation. In our previously work [1], we used the low cost Microchip PIC16F876 microcontroller to implement, for such a DC

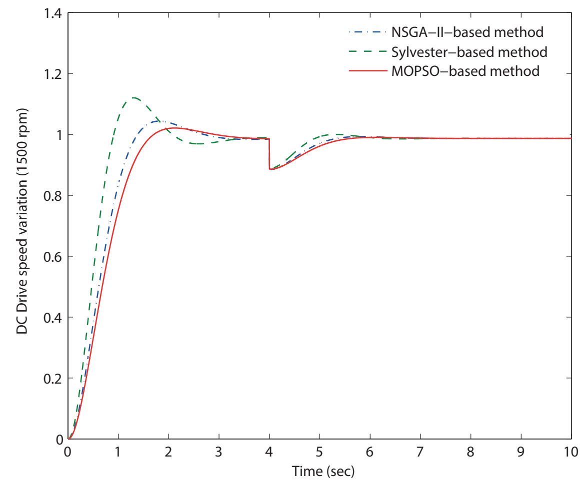

Fig. 13. Responses comparison for the RST controller.

drive plant, the RST recurrent equation obtained by the classical Sylvester-based approch.

5. CONCLUSIONS

The synthesis and tuning problem of digital RST controllers, using a MOPSO-based technique, is proposed and successfully applied to the variable speed control of an electrical DC drive. Different tests on the benchmark functions are done in order to validate the designed and implemented MOPSO algorithm. The MOPSO technique, based on Pareto dominance, proves its superiority compared to the NSGA-II technique which is based on crowding distance strategy. The obtained simulation results show the efficiency in terms of performances, robustness and less complexity of the proposed RST control approach. In addition, our approach shows a better computation time than with the NSGA-II one. This work can be enriched by using the MOPSO with the two strategies; ε-dominance and the mutation operator.

APPENDIX A

- Test Function F1:

$$\begin{cases} f_1(x_1,x_2) = x_1 \\ f_2(x_1,x_2) = g(x_1,x_2)\,h(x_1,x_2) \end{cases} \tag{A.1}$$

$$0 \le x_1 \le 1,\ -30 \le x_2 \le 30$$

$$g(x_1,x_2) = 11 + x_2^2 - 10\cos(2\pi x_2)$$

$$h(x_1,x_2) = \begin{cases} 1 - \sqrt{\dfrac{f(x_1,x_2)}{g(x_1,x_2)}}, & \text{if } f(x_1,x_2) \le g(x_1,x_2) \\ 0, & \text{otherwise} \end{cases}$$

- Test Function F2:

$$\begin{cases} f_1(x) = \sum_{i=1}^{n-1} \left(-10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right) \\ f_2(x) = \sum_{i=1}^{n} \left(|x_i|^{0.8} + 5\sin\left(x_i^3\right)\right) \end{cases} \tag{A.2}$$

$$-5 \le x \in \mathbb{R}^3 \le 5$$

- Test Function F3:

$$\begin{cases} f_1(x_1, x_2) = x_1 \\ f_2(x_1, x_2) = \frac{g(x_1, x_2)}{x_1} \end{cases} \tag{A.3}$$

$$0.1 \le x_1, x_2 \le 1$$

$$g(x_1, x_2) = 2 - \exp\left(\frac{x_2 - 0.2}{0.004}\right)^2 - 0.8\exp\left(\frac{x_2 - 0.6}{0.4}\right)^2$$

- Test Function F4:

$$\begin{cases} f_1(x_1, x_2) = -x_1^2 + x_2 \\ f_2(x_1, x_2) = 0.5x_1 + x_2 + 1 \end{cases} \tag{A.4}$$

$$0 < x_1, x_2 \le 7$$

$$0.5x_1 + x_2 - 6.5 \le 0$$

$$0.5x_1 + x_2 - 7.5 \le 0$$

$$\frac{5}{x_1} + x_2 - 30 \le 0$$

REFERENCES

[1] M. Ayadi, J. Haggège, S. Bouallègue, and M. Benrejeb, "A Digial Flatness-based Control System of a DC Drive," *Studies in Informatics and Control*, vol. 17, no. 2, pp. 201–214, 2008.

[2] I.D. Landau, "The RST Digital controller design and applications," *Control Engineering Practice*, vol. 6, pp. 155-165, 1998. [click]

[3] I.D. Landau and A. Karimi, "Robust Digital Control using Pole Placement with Sensitivity function shaping method," *International Journal of Robust and Nonlinear Control*, vol. 8, pp. 191-210, 1998. [click]

[4] F. Rotella, F. Carillo, and M. Ayadi, "Polynomial controller design based on flatness," *Proc. of The First IFAC-IEEE Symposium on System Structure and Control*, pp. 936-941, Prague, 2001.

[5] R. Madiouni, S. Bouallègue, J. Haggège, and P. Siarry, "Particle Swarm Optimization-based Design of Polynomial RST Controllers," *Proc. of The 10th International Multi-Conference on Systems, Signals and Devices*, Hammamet, 2013.

[6] G. Galdos, A. Karimi, and R. Longchamp, "RST Controller Design by Convex Optimization Using Frequency-Domain Data," *The 18th IFAC World Congress*, pp. 11429-11434, Milano, 2011.

[7] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," *Proc. of The 6th International Symposium on Micro Machine and Human Science*, pp. 39-43, Nagoya, 1995.

[8] J. Kennedy and R. Eberhart, "A new optimizer using particle swarm theory," *Proc. of The Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43, Indianapolis, 1995.

[9] Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*, Decision Engineering, Springer-Verlag, Berlin, 2004.

[10] C. A. Coello Coello, G. B. Lamont, and D. *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and Evolutionary Computation Series, Springer, 2007.

[11] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205-230, 1999. [click]

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[13] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multi-Objective Optimization*, Technical Report-112, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich, 2001.

[14] S. Z. Martinez and C. A. Coello Coello, "A multi-objective particle swarm optimizer based on decomposition," *The 13th annual Genetic and Evolutionary Computation Conference*, pp. 69-76, Dublin, 2011.

[15] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," *Proc. of The Congress on Evolutionary Computation*, pp.1945-1950, Washington, 1999.

[16] R. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," *The IEEE Congress on Evolutionary Computation*, pp. 81-86, Seoul, 2001.

[17] S. Bouallègue, J. Haggège, and M. Benrejeb, "Particle swarm optimization-based fixed-dtructure $\mathcal{H}_\infty$ control design," *International Journal of Control, Automation, and Systems*, vol. 9, no. 2, pp. 258-266, 2011. [click]

[18] S. Bouallègue, J. Haggège, M. Ayadi, and M. Benrejeb, "PID-type fuzzy logic controller tuning based on particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 25, pp. 484-493, 2012. [click]

[19] A. B. Carvalho and A. Pozo, "Measuring the convergence and diversity of CDAS multi-objective particle swarm optimization algorithms: a study of many-objective problems," *Neurocomputing*, vol. 75, no. 1, pp. 43-51, 2012. [click]

[20] D. A. Van Veldhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyzes, and New Innovations*, Ph.D. dissertation, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson, 1999.

[21] X. Blasco, J. M. Herrero, J. Sanchis, and M. Martinez, "A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization," *Information Sciences*, vol. 178, no. 20, pp. 3908-3924, 2008.

[22] S. Y. Chiu, T. Y. Sun, S. T. Hsieh, and C. W. Lin, "Cross-searching strategy for multi-objective particle swarm optimization," *The IEEE Conference on Evolutionary Computation*, pp. 3135-3141, New York, 2007.

[23] M. Reyes-Sierra and C. A. Coello Coello, "Multi-objective particle swarm optimizers: a survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287-308, 2006.

[24] M. Shokrian and K. A. Highb, "Application of a multi objective multi-leader particle swarm optimization algorithm on NLP and MINLP problems," *Computers and Chemical Engineering*, vol. 60, pp. 57-75, 2014. [click]

[25] H. Moslemi and M. Zandieh, "Comparisons of some improving strategies on MOPSO for multi-objective (r, Q) inventory system," *Expert Systems with Applications*, vol. 38, pp. 12051-12057, 2011. [click]

[26] S. Gangulya, N.C. Sahoob and D. Dasc, "Multi-objective particle swarm optimization based on fuzzy-Pareto-dominance for possibilistic planning of electrical distribution systems incorporating distributed generation," *Fuzzy Sets and Systems*, vol. 213, pp. 47-73, 2013. [click]

[27] S. Panda and N. K. Yegireddy, "Automatic generation control of multi-area power system using multi-objective non-dominated sorting genetic algorithm-II," *Electrical Power and Energy Systems*, vol. 53, pp. 54-63, 2013. [click]

[28] G. Reynoso-Meza, X. Blasco, J. Sanchis and M. Martínez, "Controller tuning using evolutionary multi-objective optimization: Current trends and applications," *Control Engineering Practice*, vol. 28, pp. 58-73, 2014. [click]

[29] G. Reynoso-Meza, X. Blasco, J. Sanchis and S. Garcya-Nieto, "A multi-objective optimization design methodology for SISO PID controllers," *The IFAC Conference on Advances in PID Control*, Brescia-Italy, 2012.

[30] I. K. Kookos, K. G. Arvanitis and G. Kalogeropoulos, "PI controller tuning via multiobjective optimization," *The 7th Mediterranean Conference on Control and Automation*, Haifa, pp. 408-419, 1999.

Riadh Madiouni was born in Tunisia in 1987. He graduated from the Faculty of Sciences of Bizerte in Computer Sciences, in 2012. He received his master's degree in Automatic and Signal Processing from the National School of Engineers of Tunis (ENIT), in 2012 and is currently preparing a PhD Thesis in the LARA and LiSSi (University of Paris-Est) laboratories, since 2012. His research interests are in multi-objective particle swarm optimization and robust control design.

Soufiene Bouallègue was born in 1982 in Nafta, Tunisia. He graduated from the National School of Engineers of Tunis (ENIT) in 2006 and received the PhD degree in Electrical Engineering in 2010. He is currently an Associate Professor of Electrical Engineering at the High Institute of Industrial Systems of Gabés (ISSIG). His research interests are in the area of meta-heuristics optimization, intelligent control, robotics, renewable energies, and digital control applications.

Joseph Haggège was born in 1975 in Tunis, Tunisia. He graduated from National School of Engineers of Tunis in 1998. He received his PhD degree in Electrical Engineering 2003 and the Habilitation in 2010. He is currently a Senior Lecturer at the National School of Engineers of Tunis (ENIT). His research interests are in the area of meta-heuristics optimization, embedded systems and robust digital control.

Patrick Siarry was born in France in 1952. He received his PhD degree from the University Paris 6, in 1986, and the Doctorate of Sciences (Habilitation) from the University Paris 11, in 1994. He was first involved in the development of analog and digital models of nuclear power plants at Electricité de France (E.D.F.). Since 1995 he is a Professor in automatics and informatics. His main research interests are computer-aided design of electronic circuits, and the applications of new stochastic global optimization heuristics to various engineering fields. He is also interested in the fitting of process models to experimental data, the learning of fuzzy rule bases, and of neural networks.