2013 10th International Multi-Conference
on Systems, Signals & Devices (SSD)
Hammamet, Tunisia, March 18-21, 2013

Particle Swarm Optimization-based Design of Polynomial RST Controllers

Riadh Madiouni [1, 2, *], Soufiene Bouallègue [1], Joseph Haggège [1] and Patrick Siarry [2]

[1] Research Laboratory in Automatic Control (LA.R.A), National Engineering School of Tunis (ENIT), BP 37, Le Belvédère, 1002 Tunis, University Tunis El Manar, Tunisia.
[2] Signals Images & Intelligent Systems Laboratory, LiSSi-EA-3956, University Paris-Est Créteil, France.

E-mail addresses: riadh.madiouni@univ-paris-est.fr, soufiene.bouallegue@issig.rnu.tn, joseph.haggege@enit.rnu.tn, siarry@univ-paris12.fr

Abstract—In this paper, we propose a new method for digital RST controller design based on the Particle Swarm Optimization meta-heuristic. It is a systematic RST synthesis and tuning procedure to deal with the complexity of the known classical poles placement methods. The case of an electric DC drive benchmark has been successfully obtained to illustrate the efficiency of the proposed PSO-based RST control approach. Simulation results show the advantages of the designed PSO-tuned RST structure in terms of performance and robustness. A comparison with the well known Genetic Algorithm Optimization technique is investigated in order to show the superiority and effectiveness of the proposed PSO-based approach.

Index Terms—Robust control, RST controller, Particle Swarm Optimization (PSO), Genetic Algorithm Optimization (GAO), DC drive benchmark .

1 Introduction

In the last decades, the robust control theory has reached a remarkable level of maturity. Many synthesis methods have been developed, allowing to set up a unified framework for the design and analysis of robust control laws [4]. The RST polynomial control design, as given and developed in [12, 13], is a robust and effective strategy for a digital controller synthesis that is widely used in practical and industrial applications. In the literature, there are essentially two conventional techniques of RST controller synthesis ie a method with closed loop pole placement as well as a method with poles placement and loop-shaping of sensitivity functions.The first design technique is based on the placement of the closed-loop poles for the feedback controlled system. A Sylvester's method, as given in [12], can be used to determine the search parameters of the RST controller. However, the choice of the closed-loop poles is usually difficult and becomes more complicated with the complexity of the controller plants. The second approach, developed by I.D. Landau in [12], is based on the calibration of the well known sensitivity functions to guide the choice of the desired closed-loop poles. This iterative procedure uses robustness templates as shown in [12] to obtain an adequate placement of the poles with the desired robustness and performance criteria. So, this trial-and error-based method is not a practical and systematic approach to synthesis of such a RST controller. Up to now, there has not been a clear and systematic method to guide the RST synthesis problem. So, the synthesis problem becomes more delicate and hard as the complexity of the controller plant is increasing. A systematic approach to the RST synthesis procedure and reducing the complexity of the obtained controller, for easy practical implementations, is an important and interesting task in this area. The optimization theory can provide a suitable solution to deal with this complexity, especially with the remarkable evolution of the powerful of numerical calculation tools. The RST synthesis problem can be reformulated as an optimization problem which can be solved by various optimization techniques, given in the literature [6]. Among these techniques, the most suitable for this kind of difficult and NP-hard optimization problem, are the meta- heuristic methods such as the recent Particle Swarm Optimization method [10]. In this paper, a new approach based on the Particle Swarm Optimization (PSO) meta-heuristic technique is proposed for RST controller synthesis problem. The RST control design problem is formulated as a constrained optimization problem, which is efficiently solved based on a developed PSO algorithm [9, 11]. In order to specify more robustness and performance control objectives of the proposed PSO-tuned RST, different optimization criteria, such as the IAE and ISE index, are considered, compared and subjected to several various control constraints. Convergence conditions of the proposed PSO algorithm are analytically guaranteed and verified in order to given a set of PSO algorithm parameters [17]. The main contribution of this paper consists of proposing a systematic method of synthesis and tuning of digital RST controllers. The rest of this paper is organized as follows. In section 2, the considered RST structure is presented and its optimization-based synthesis problem is formulated. In Section 3, a constrained PSO algorithm, used to solve the formulated RST synthesis problem, is proposed and implemented. Section 4 is dedicated to applying the proposed PSO-based control approach for an electrical DC drive benchmark.

All PSO-based simulation results are compared, and discussed, with those obtained by the classical Genetic Algorithms Opti-

- * *Corresponding author, Tel: +216 71 87 47 00; Fax: +216 71 87 27 29. Website: http://www.lissi.fr/lab/members/doctoral/madiouni_riadh*

978-1-4673-6457-7/13/$31.00 ©2013 IEEE

mization (GAO)-based approach in order to show the effectiveness and superiority of the proposed strategy.

2 Optimization-based Control prolblem Formulation

2.1 The RST Digital Controller

In the theory of robust control, the controller's synthesis problem and its practical implementation are increasing in complexity. This type of control is based on the synthesis of a digital controller with two degrees-of-freedom says RST, as shown in Fig. 1.

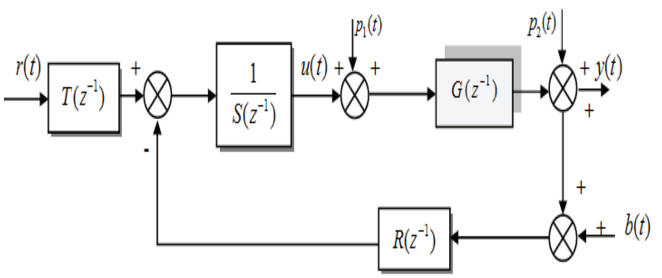

Fig. 1. Closed-loop system with an RST controller in the presence of output disturbances and measurement noise.

The synthesis of polynomial RST controller is based on two conventional techniques for RST controller synthesis: a method with closed loop poles placement as well as a method with pole placement and loop-shaping of sensitivity functions to meet the objectives of regulation and tracking desired. The RST symbol is the name of three polynomials that must be determined in order to obtain the desired control[12].

The computation of the controller in the pole placement technique requires specification of the desired closed-loop poles and of some fixed parts of the controller for the rejection of disturbances at various frequencies. This structure has two degrees of freedom: the digital filters R and S are designed in order to achieve the desired regulation performance, and T the digital filter is designed afterwards in order to achieve the desired tracking performance [13]. This structure allows to obtain different levels of performance in tracking and regulation. Figure 1 shows a schematic diagram of an RST controller. The discrete time plant is described by the following transfer operator given by:

$$G(z^{-1}) = \frac{z^{-d}B(z^{-1})}{A(z^{-1})} \tag{1}$$

Where d is the integer number of sampling periods contained in the time-delay of the plant and the two polynomials A, B are defined by:

$$\begin{cases} B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_B} z^{-n_B} \\ A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{n_A} z^{-n_A} \end{cases} \tag{2}$$

The polynomials $T(z^{-1})$,$R(z^{-1})$ and $S(z^{-1})$ have the form:

$$\begin{cases} T(z^{-1}) = t_0 + t_1 z^{-1} + t_2 z^{-2} + ... + t_{n_T} z^{-n_T} \\ R(z^{-1}) = r_0 + r_1 z^{-1} + r_2 z^{-2} + ... + r_{n_R} z^{-n_R} \\ S(z^{-1}) = s_0 + s_1 z^{-1} + s_2 z^{-2} + ... + s_{n_S} z^{-n_S} \end{cases} \tag{3}$$

2.1.1 Pole placement

In general, the desired closed-loop poles are specified in the form:

$$P(z^{-1}) = P_D(z^{-1})P_F(z^{-1}) \tag{4}$$

Where $P_D(z^{-1})$ specifies the desired dominant poles of the closed loop, and $P_F(z^{-1})$ specifies the auxiliary poles of the closed loop. Once the closed-loop poles have been defined, solving the equation:

$$P(z^{-1}) = A(z^{-1})S(z^{-1}) + z^{-d}B(z^{-1})R(z^{-1}) \tag{5}$$

Allows the determination of $S(z^{-1})$ and $R(z^{-1})$, which will ensure the desired closed-loop poles. Let the degrees of polynomials $A(z^{-1})$ and $B(z^{-1})$ be defined by:
$n_A = \deg(A(z^{-1}))$ and $n_B = \deg(B(z^{-1}))$. Then the above equation has a unique solution (assuming that $A(z^{-1})$ and $B(z^{-1})$ do not have common factors) for:

$$\begin{cases} n_P = \deg P(z^{-1}) \leq n_A + n_B + d - 1 \\ n_S = \deg S(z^{-1}) = n_B + d - 1 \\ n_R = \deg R(z^{-1}) = n_A - 1 \end{cases} \tag{6}$$

The polynomial equation (5) can be written in matrix form:

$$M.X = P \tag{7}$$

The direct implementation of RST regulator often leads to a control signal, whose amplitude can be very large, to maintain this amplitude in acceptable interval, we add to the control structure a dispositif "Anti-windup" [1].

2.1.2 Poles placement and loop-shaping of sensitivity functions

The sensitivity functions play a crucial role in the robustness analysis of the closed-loop system with respect to modelling errors. These functions will be shaped in order to assure nominal performance for the rejection of the disturbances and the stability of the closed-loop system in the presence of model mismatch. Two types of disturbances are considered: output disturbance and measurement noise [12].

- The input sensitivity function which reflects the effects of an output disturbance upon the plant input is given by: $S_{up}(z^{-1}) = \frac{-A(z^{-1})R(z^{-1})}{P(z^{-1})}$
- The transfer function between the measurement noise b(t) and the plant output y(t) (noise sensitivity function) is given by: $S_{yb}(z^{-1}) = \frac{-z^{-d}B(z^{-1})R(z^{-1})}{P(z^{-1})}$
- The transfer function between the disturbance p(t) and the plant output y(t) (output sensitivity function) is given by: $S_{yp}(z^{-1}) = \frac{A(z^{-1})S(z^{-1})}{P(z^{-1})}$

2.2 RST synthesis problem formulation

In this section, the problem of synthesis RST controller is formulated as a constrained optimization problem which is solved using the proposed PSO-based approach. The choice of the adequate pole values for the RST controller is often done by a trial-error procedure [12]. This tuning problem becomes difficult and delicate without a systematic design method. To deal with these difficulties, the optimization of these scaling factors is proposed as a promising solution. This problem can be formulated as the following constrained optimization problem:

$$\begin{cases} \underset{x \in D}{minimize} \, f(x) \\ subject \ to \\ g_l(x) \leq 0; \quad \forall l = 1, \dots, n_{con} \end{cases} \tag{8}$$

Where $f : \Re^m \rightarrow \Re$ the cost function, $D = \{x \in \Re^m; x_{\min} \leq x \leq x_{\max}\}$ the initial search space, which is supposed to contain the desired design parameters, and $g_l : \Re^m \rightarrow \Re$ the problem constraints. The optimization-based tuning problem consists in finding the optimal decision variables $x^* = (x_1^*, x_2^*, \dots, x_m^*)^T$, representing the RST controller structure, which minimize the defined cost function, chosen as the Integral of squared error ($J_{ISE} = \int_0^{+\infty} e(t)^2 dt$) and the Integral of Absolute Error ($J_{IAE} = \int_0^{+\infty} |e(t)| \, dt$) performance criteria [5, 6]. These cost functions are minimized, using the proposed constrained PSO algorithm, as shown in the equations(8). Hence, in the case of RST controller structure, the poles to be optimized are $(s_0, s_1, ..., s_n)$ and $(r_0, r_1, ...r_m)$. The formulated optimization problem is defined as follows:

$$\underset{x=(s_0,s_1,s_2\dots,s_n,r_0,r_1,r_2,\dots,r_m)^T \in \Re^{+k}}{minimize} f(x) \tag{9}$$

3 Proposed Particle Swarm Optimization approach

In this study, the proposed PSO approach is presented and a constrained PSO algorithm is also developed. The convergence conditions of such an algorithm are analyzed and established.

3.1 Overview

The PSO technique is an evolutionary computation method developed by Kennedy and Eberhart (1995). This recent meta-heuristic technique was inspired by the swarming or collaborative behaviour of biological populations [7, 8, 16, 17]. The cooperation and the exchange of information between population individuals allows to solve various complex optimization problems [10, 14, 15, 16, 17].

The recourse to this stochastic and global optimization technique is justified by the empirical evidence of its superiority in solving a variety of non-linear, non-convex and non-smooth problems Without any regularity on the cost function to be optimized. In comparison to other meta-heuristics, this optimization technique is a simple concept, easy to implement, and a computationally efficient algorithm [14, 10, 16, 5]. The convergence and parameter selection of the PSO algorithm have been proved by using several advanced theoretical analyses [16, 17]. Its stochastic behaviour allows to overcome the local minima problem.

Particle Swarm Optimisation has been enormously successful in several and various industrial domains. It has been used across a wide range of engineering applications. These applications can be summarized in the domains of robotics, image and signal processing, electronic circuits design, communication networks, and more especially the domain of plant control design, as shown in [2, 3, 4].

3.2 Basic PSO algorithm

The basic PSO algorithm uses a swarm consisting of n_p particles (i.e.$x^1, x^2, \dots, x^{n_p}$), randomly distributed in the considered initial search space, to find an optimal solution $x^* = \arg\min f(x) \in \Re^m$ of a generic optimization problem (8). Each particle, that represents a potential solution, is characterised by a position and a velocity given by $x_k^i := \left(x_k^{i,1}, x_k^{i,2}, \dots, x_k^{i,m}\right)^T$ and $v_k^i := \left(v_k^{i,1}, v_k^{i,2}, \dots, v_k^{i,m}\right)^T$ where $(i,k) \in [\![1, n_p]\!] \times [\![1, k_{\max}]\!]$. At each algorithm iteration, the i^{th} particle position, $x^i \in \Re^m$, evolves in the basis of the following update rules:

$$x_{k+1}^i = x_k^i + v_{k+1}^i \tag{10}$$

$$v_{k+1}^i = w_{k+1} v_k^i + c_1 r_{1,k}^i \left(p_k^i - x_k^i\right) + c_2 r_{2,k}^i \left(p_k^g - x_k^i\right) \tag{11}$$

where;
w_{k+1}: the inertia factor,
c_1, c_2: the cognitive and the social scaling factors respectively,
$r_{1,k}^i$, $r_{2,k}^i$: random numbers uniformly distributed in the interval $[\![0, 1]\!]$,
p_k^i: the best previously obtained position of the i^{th} particle,
p_k^g: the best obtained position in the entire swarm at the current iteration k.

Hence, the principle of a particle displacement in the swarm is graphically shown in the Fig. 2, for a two dimensional design space. In order to improve the exploration and exploitation

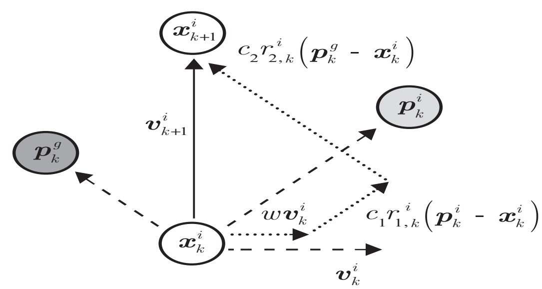

Fig. 2. Particle position and velocity update.

capacities of the proposed PSO algorithm, we chose as the inertia factor a linear evolution with respect to the algorithm iteration as given in [15]:

$$w_{k+1} = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{k_{\max}}\right) k \tag{12}$$

where $w_{\max} = 0.9$ and $w_{\min} = 0.4$ represent the maximum and minimum inertia factor values, respectively, $k_{\max}$ is the maximum iteration number. Similar to other meta-heuristic methods, the PSO algorithm is originally formulated as an unconstrained optimizer. Several techniques have been proposed to deal with constraints [2]. One useful approach is by augmenting the cost function of problem (8) with penalties proportional to the degree of constraint infeasibility. In this paper, the following external static penalty technique is used:

$$\varphi(x) = f(x) + \sum_{l=1}^{n_{con}} \lambda_l \max\left[0, g_l(x)^2\right] \tag{13}$$

where λ_l is a prescribed scaling penalty parameter and n_{con} is the number of problem constraints $g_l(x)$. Finally, the basic proposed PSO algorithm can be summarized in following steps [6]:
1. Defining all PSO algorithm parameters such as swarm size n_p, maximum and minimum inertia factor values, cognitive c_1 and social c_2 scaling factors, etc.
2. Initializing n_p the particles with randomly chosen positions x_0^i and velocities v_0^i in the search space D. Evaluating the initial population and determining p_0^i and p_0^g.
3. Incrementing the iteration number k. For each particle

apply the update equations (10) and (11), and evaluate the corresponding fitness values $\varphi_k^i = \varphi\left(x_k^i\right)$:
(i) if $\varphi_k^i \le pbest_k^i$ then $pbest_k^i = \varphi_k^i$ and $p_k^i = x_k^i$,
(ii) if $\varphi_k^i \le gbest_k$ then $gbest_k = \varphi_k^i$ and $p_k^g = x_k^i$.
where $pbest_k^i$ and $gbest_k$ represent the best previously fitness of the i^{th} particle and the entire swarm, respectively.
4. If the termination criterion is satisfied, the algorithm terminates with the solution $x^* = \arg\min_{x_k^i}\left\{f\left(x_k^i\right), \forall i, k\right\}$. Otherwise, go to step 3.

3.3 The convergence of PSO algorithm analysis

In this part, the proposed PSO algorithm is analysed on the basis of [16, 17] results. Theoretical conditions for the convergence algorithm and parameter choice are established. Let us replace the velocity update equation (11) into the position update equation (10) to get the following expression:

$$x_{k+1}^i = \left(1 - c_1 r_{1,k}^i - c_2 r_{2,k}^i\right) x_k^i + w v_k^i + c_1 r_{1,k}^i p_k^i + c_2 r_{2,k}^i p_k^g \quad (14)$$

A similar re-arrangement of the velocity term (13) leads to:

$$v_{k+1}^i = -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) x_k^i + w v_k^i + c_1 r_{1,k}^i p_k^i + c_2 r_{2,k}^i p_k^g \quad (15)$$

The obtained equations (14) and (15) can be combined and written in matrix form as follows:

$$\begin{aligned}\begin{bmatrix} x_{k+1}^i \\ v_{k+1}^i \end{bmatrix} = \begin{bmatrix} 1 - \left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \\ -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \end{bmatrix} \begin{bmatrix} x_k^i \\ v_k^i \end{bmatrix} + \\ \begin{bmatrix} c_1 r_{1,k}^i & c_2 r_{2,k}^i \\ c_1 r_{1,k}^i & c_2 r_{2,k}^i \end{bmatrix} \begin{bmatrix} p_k^i \\ p_k^g \end{bmatrix}\end{aligned} \quad (16)$$

The above expression can be considered as a state-space representation of a discrete-time dynamic linear system, given by:

$$\hat{y}_{k+1} = \mathcal{M}\hat{y}_k + \mathcal{N}\hat{u}_k \quad (17)$$

where $\hat{y}_k$ is the state vector, $\hat{u}_k$ the external input system, $\mathcal{M}$ and $\mathcal{N}$ the dynamic and input matrices respectively, defined as:

$$\hat{y}_k = \begin{bmatrix} x_k^i \\ v_k^i \end{bmatrix}, \hat{u}_k = \begin{bmatrix} p_k^i \\ p_k^g \end{bmatrix} \quad (18)$$

$$\mathcal{M} = \begin{bmatrix} 1 - \left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \\ -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \end{bmatrix}, \mathcal{N} = \begin{bmatrix} c_1 r_{1,k}^i & c_2 r_{2,k}^i \\ c_1 r_{1,k}^i & c_2 r_{2,k}^i \end{bmatrix}$$

For a given particle, the convergent behaviour can be maintained while assuming that the external input is constant, as there is no external excitation in the dynamic system. In such a case, as the iterations go to infinity the updated positions and velocities become constant from k^{th} to $(k+1)^{th}$ iteration, given the following equilibrium state:

$$\hat{y}_{k+1} - \hat{y}_k = \begin{bmatrix} -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \\ -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w-1 \end{bmatrix} \hat{y}_k + \mathcal{N} \times \hat{u}_k = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (19)$$

Therefore, we obtain an equilibrium point, for which all particles tend to converge as algorithm iteration progresses, given by:

$$\hat{y}_{eq} = [p_k^g, 0]^T \quad (20)$$

So, the dynamic behaviour of the i^{th} particle can be analysed using the eigenvalues derived from the dynamic matrix formulation (17) and (18), solutions of the following characteristic polynomial:

$$\lambda^2 - \left(1 + w - c_1 r_{1,k}^i - c_2 r_{2,k}^i\right)\lambda + w = 0 \quad (21)$$

The following necessary and sufficient conditions for stability of the considered discrete-time dynamic system (20) are obtained while applying the classical Jury criterion:

$$\begin{array}{l} |w| < 1 \\ c_1 r_{1,k}^i + c_2 r_{2,k}^i > 0 \\ w + 1 - \frac{c_1 r_{1,k}^i + c_2 r_{2,k}^i}{2} > 0 \end{array} \quad (22)$$

Knowing that $r_{1,k}^i, r_{2,k}^i \in [\![0,1]\!]$, the above stability conditions are equivalents to the following set of parameter selection heuristics which guarantee convergence for the PSO algorithm:

$$\begin{array}{l} 0 < c_1 + c_2 < 4 \\ \frac{c_1+c_2}{2} - 1 < w < 1 \end{array} \quad (23)$$

While these heuristics provide useful selection parameter bounds, an analysis of the effect of the different parameter settings is obtained and verified by some numerical simulations to determine the effect of such parameters in the PSO algorithm convergence performance.

4 DC DRIVE BENCHMARK CONTROL

4.1 Plant model description

The considered benchmark is a 250 watts electrical DC drive. 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 developed real-time application acquires input data (speed of the DC drive) and generates control signal for thyristors of AC-DC power converter (PWM signal). The considered electrical DC drive can be described by the following model that is used in the design setup:

The model parameters are obtained by an experimental identification procedure and they are summarized in Table 1 with their associated uncertainty bounds. Moreover, this model is sampled with 10 ms sampling time for simulation and experimental setups [1].

$$G(s) = \frac{k_m}{(1 + \tau_m s)(1 + \tau_e s)} \quad (24)$$

TABLE 1

Identified DC drive model parameters.

Parameters	Nominal values	Uncertainty bounds
k_m	0.05	80 %
τ_m	300 ms	80 %
τ_e	14 ms	80 %

The discretization of the continuous transfer function of the model used with a period of simpling T leads to the transfer function which is defined as follows:

$$H(z) = \frac{0.0005z + 0.0004}{z^2 - 1.431z + 0.4482} \quad (25)$$

The polynomials $R(z^{-1})$and $S(z^{-1})$ have the form:

$$\left\{ \begin{array}{l} R(z^{-1}) = r_0 + r_1 z^{-1} \\ S(z^{-1}) = s_0 + s_1 z^{-1} \end{array} \right. \quad (26)$$

Hence the optimization problem is defined with a dimension k=4;

$$\begin{cases} \underset{x=(s_0,s_1,r_0,r_1)^T\in\Re^{+4}}{minimize} f(x) \\ f(x)=\int_0^{+\infty}|e(x)|\,dx \end{cases} \quad (27)$$

In order to confirm the convergence conditions and the choice of parameters of the PSO algorithm implemented, we implemented the algorithm for the coffecient values of social c_1 and cognitive c_2 on the one hand, and its inertia factor w of on the other hand which are shown in the table below. These results show the robustness of the PSO technique, This justifies the use of this global optimization method.

TABLE 2

The PSO Algorithm parameters.

PSO Algorithm Parameters	values
c_1	1,2
c_2	1,2
$[w_{\min}, w_{\max}]$	$[0.4, 0.9]$

4.2 Simulation results

The swarm size algorithm choice is generally problem-dependent in the PSO framework. However,[10] as well as [14] show that this parameter is often set empirically in relation to dimensionality and perceived difficulty of a considered optimization problem. They suggest that swarm size values in the range of 20-50 are quite common. For this purpose, we tested the proposed PSO algorithm with different values in this range in the case of RST controller structure. On the whole, all the results obtained are close to each other.However, best values of the fitness are obtained while using a swarm size equal to 30. Henceforth, this value will be adopted for our future works.

In the PSO framework, it is necessary to run the algorithm several times in order to get some statistical data on the quality of results and thus to validate the proposed approach. We run the proposed algorithm 20 times for the IAE criterion case. The obtained optimization results are summarized in Tables 3. Besides, the fact that the algorithm convergence always takes place in the same region of the design space, whatever is the initial population, indicates that the algorithm succeeds in finding a region of the interesting research space to explore. The performance comparison of PSO- and GAO-based approaches is obtained in the same conditions.

Indeed, the population size, used in the GAO algorithm, is set at 30 individuals and the maximum generation number is 50. The algorithm stops when the number of generations reaches the specified value for the maximum generation. According to the statistical analysis of Table 3, we can conclude that the proposed PSO-based approach produces better results in comparison with the standard GAO-based one. On the other

TABLE 3

Optimisation results from 20 trials of the problem.

Cost function	Algorithme	Best	Mean	Worst
IAE	PSO	**0.5662**	0.6044	0.6862
IAE	GAO	0.5857	0.6771	0.7341

hand, performance on convergence properties of the proposed PSO and the used GAO algorithm, in terms of iteration number required to find the best solution, are compared in the IAE criterion case, as shown in Fig. 4 and Fig. 3 While using the proposed PSO-based method, we succeed in obtaining the optimal solution within only about 25 iterations.

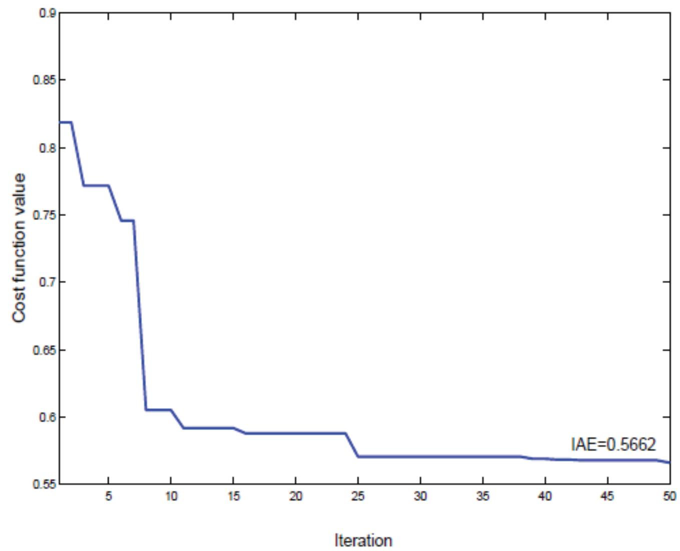

Fig. 3. Convergence of the PSO algorithms: IAE criterion case.

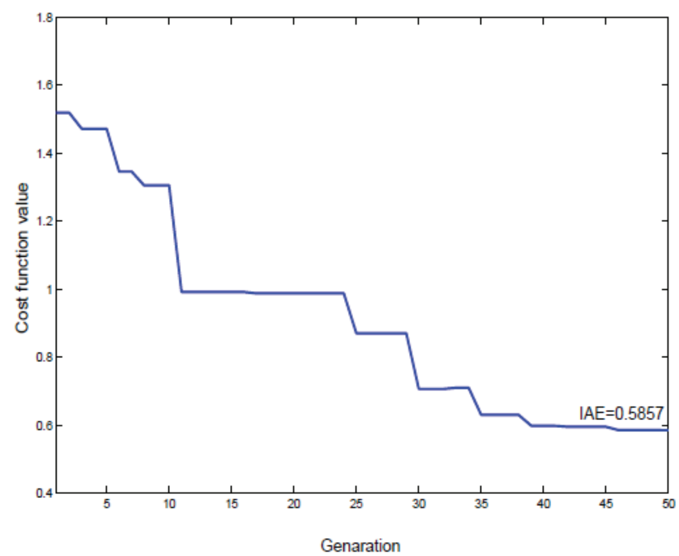

Fig. 4. Convergence of the GAO algorithms: IAE criterion case.

However, the GAO-based method finds the same result after 45 iterations. All these observations can show the superiority of the proposed PSO-based method in comparison with the GAO-based one. Indeed, the quality of the obtained optimal solution, the fastness convergence as well as the simple software implementation is better than those in the GAO-based approach. Besides the PSO algorithm convergence is faster than the GAO algorithm. The robustness of the proposed PSO algorithm convergence, under variation of the cognitive, social and inertia factor parameters, is analysed on the basis of numerical simulations as shown in Fig. 5 and Fig. 6.

The PSO algorithm's convergence is guaranteed within the established domain given by the equation (23). the Implemen-

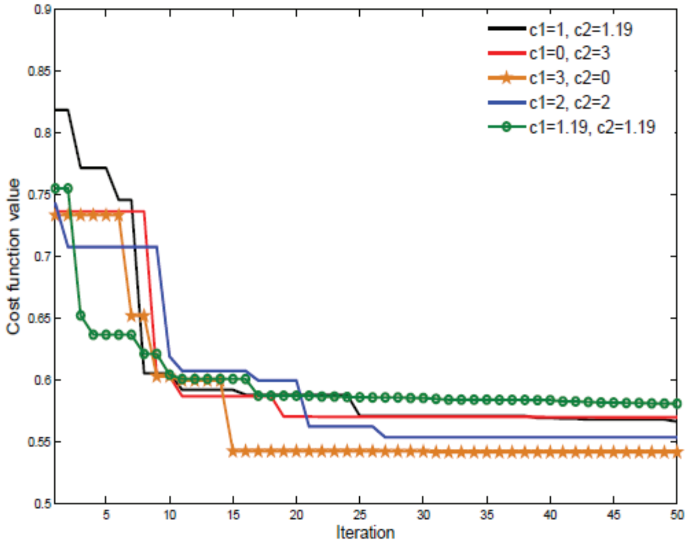

Fig. 5. robustness of the proposed PSO algorithm under variations of cognitive and social coefficients: IAE criterion case.

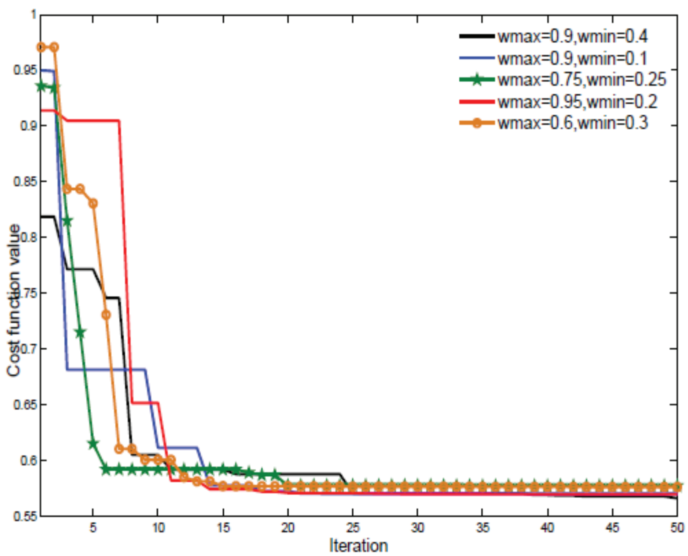

Fig. 6. robustness of the proposed PSO algorithm under variations of inertia factor: IAE criterion case.

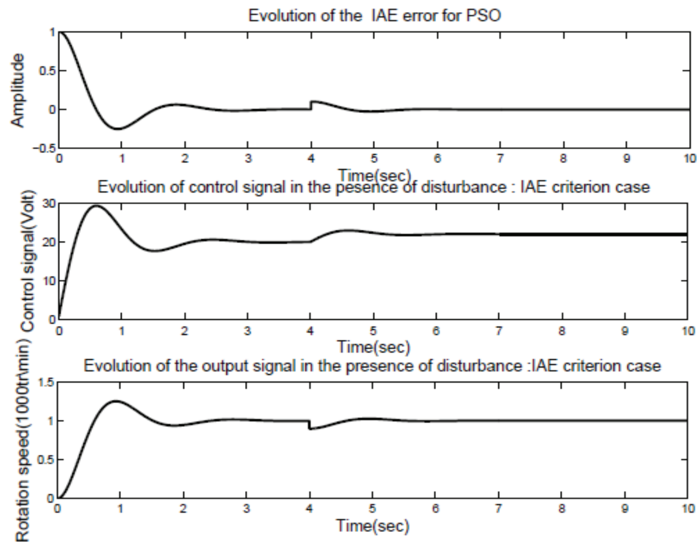

Fig. 7. Step response for the IAE criterion case-PSO.

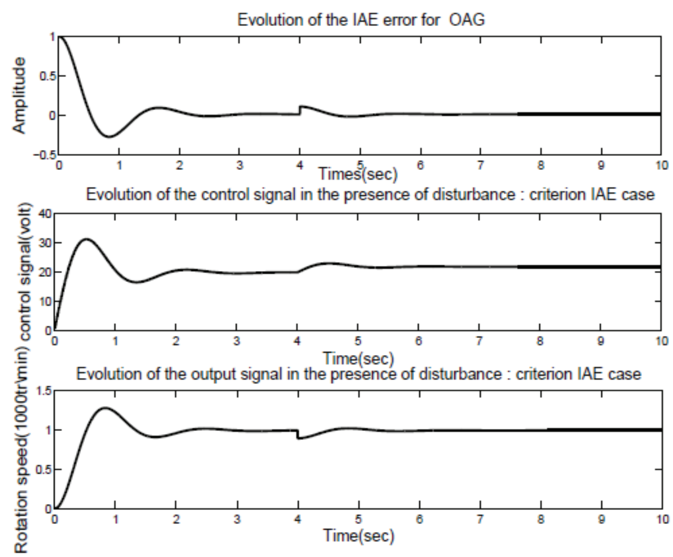

Fig. 8. Step response for the IAE criterion case-GAO.

tation in MATLAB-SIMULINK of PSO and GAO algorithms for the ISE criterion and the IAE criterion leads to the simulation results of Fig. 7and Fig. 8.

In the tables below and the above results, we obtained the parameters of an RST controller. The convergence speed of the proposed method PSO and the quality of the solutions it provides, compared to the GAO case, were particularly noticable. This can be observed, for example, from the results obtained in the case of the RST command when optimizing performance; the IAE criterion and the ISE criterion.

Based on the results we noted that the solution given by the PSO is better than the solution given by the GAO. And from a technical point of view and during the various tests that we conducted, it is clear that optimization with genetic algorithms is expensive in terms computation time in contrast with particle swarm optimization. Implementation for the speed control of a DC current machine demonstrates the robustness, performance

TABLE 4

Optimization of the RST controller parameters : ISE criterion case.

RST Parameters / Optimization techniques	s_0	s_1	r_0	r_1	ISE
PSO	1	0.0527	0.0403	0.4588	0.269
GAO	1	0.026	0.035	0.4622	0.276

TABLE 5

Optimization of the RST controller parameters : IAE criterion case.

RST Parameters / Optimization Techniques	s_0	s_1	r_0	r_1	IAE
PSO	1	0.2061	0.1823	0.3169	0.566
GAO	1	2.41e-4	0.05009	0.003	0.585

and effectiveness of the proposed control structures based PSO technique. Its superiority in terms of convergence speed and solution quality has been shown in relation to the GAO technique.

5 Conclusion

The synthesis problem of digital RST controllers, using a new PSO-based technique, is proposed and successfully applied to an electrical DC drive speed control. The performance comparison, with the standard GAO-based method, shows the efficiency and superiority of the proposed PSO-based approach in terms of the obtained solution qualities, the convergence speed and the simple software implementation of its algorithm. The obtained simulation results show the efficiency in terms of performance, robustness and less complexity of the proposed RST control approach which can be easily applied to the industrial motor control field. The implementation of this control law, within a practical and real-time control framework,will be the subject of our future work.

References

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

[2] S. Bouallègue, J. Haggège and M. Benrejeb,*"Structured Loop-Shaping H∞ Controller Design using Particle Swarm Optimization"*,Proceedings of the 2010 IEEE International Conference on Systems, Man, and Cybernetics SMC'10, 2010, Istanbul.

[3] S. Bouallègue, J. Haggège and M. Benrejeb,"Structured Mixed-Sensitivity H∞ Design using Particle Swarm Optimization". Proceedings of the 7th IEEE International Multi-Conference on Systems, Signals and Devices SSD'10, 2010, Amman.

[4] S. Bouallègue, J. Haggège and M. Benrejeb, "Particle Swarm Optimization-Based Fixed-Structure H∞ Control Design". International Journal of Control, Automation, and Systems, vol. 9, no. 2, pp. 258-266, 2011.

[5] 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.

[6] S. Bouallègue, J. Haggège and M. Benrejeb, "A New Method for Tuning PID-Type Fuzzy Controllers Using Particle Swarm Optimization", Chapter 6, pp. 139-162, in Fuzzy Controllers-Recent Advances in Theory and Applications book (edited by Sohail Iqbal), InTech Education and Publishing, Rijeka, Croatia, 2012.

[7] Y. Collette and P. Siarry, "Optimisation multi-objectif", Editions Eyrolles, Paris, 2002.

[8] M. Clerc and P. Siarry. "Une mthode inspire de comportements coopratifs observs dans la nature : l'optimisation par essaim particulaire. Revue d'Electricit et d'Electronique REE", (4) :25-32, 2009

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

[10] R. Eberhart and Y. Shi, "Particle Swarm Optimization: Developments, Applications and Resources", in Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, Korea, pp. 81-86, 2001.

[11] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1942-1948, Perth-Australia, 1995.

[12] 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.

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

[14] R. Poli, J. Kennedy and T. Blackwell, "Particle Swarm Optimization: An Overview", Swarm Intelligence, Springer, vol. 1, pp. 33-57, 2007.

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

[16] F. Van den Bergh, "An Analysis of Particle Swarm Optimizers". PhD Thesis, University of Pretoria, Pretoria, South Africa, 2006.

[17] E.P. Ruben and B. Kamran, "Particle Swarm Optimization in Structural Design", in: Chan, F.T.S., Tiwari, M.K. (Eds.), Swarm Intelligence: Focus on Ant and Particle Swarm Optimization. In-Tech Education and Publishing, Vienna, pp. 373-394, 2007.