Chapter 6

A New Method for Tuning PID-Type Fuzzy Controllers Using Particle Swarm Optimization

S. Bouallègue, J. Haggège and M. Benrejeb

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/47139

1. Introduction

The complexity of dynamic system, especially when only qualitative knowledge about the process is available, makes it generally difficult to elaborate an analytic model which is sufficiently precise enough for the control. Thus, it is interesting to use, for this kind of systems, non conventional control techniques, such as fuzzy logic, in order to achieve high performances and robustness [8, 15, 20–22, 24, 33, 34]. Fuzzy logic control approach has been widely used in many successful industrial applications which have demonstrated high robustness and effectiveness properties.

In the literature, various Fuzzy Controller (FC) structures are proposed and extensively studied. The particular structure given by Qiao and Mizumoto in [26], namely PID-type FC, is especially established and improved within the practical framework in [11, 16, 31]. Such a FC structure, which retains the characteristics similar to the conventional PID controller, can be decomposed into the equivalent proportional, integral and derivative control components as shown in [26]. In order to improve further the performance of the transient and steady state responses of this kind of fuzzy controller, various strategies and methods are proposed to tune the PID-type fuzzy controller parameters.

Indeed, Qiao and Mizumoto [26] designed a parameter adaptive PID-type FC based on a peak observer mechanism. This self-tuning mechanism decreases the equivalent integral control component of the fuzzy controller gradually with the system response process time. On the other hand, Woo et al. [31] developed a method to tune the scaling factors related to integral and derivative components of the PID-type FC structure via two empirical functions and based on the system's error information. In [12, 16], the authors proposed a technique that adjusts the scaling factors, corresponding to the derivative and integral components of the PID-type FC, using a fuzzy inference mechanism. However, the major drawback of all these PID-type FC structures is the difficult choice of their relative scaling factors. Indeed, the fuzzy controller dynamic behaviour depends on this adequate choice. The tuning procedure depends on the control experience and knowledge of the human operator, and it is generally

achieved based on a classical trials-errors procedure. Up to now, there is neither clear mor systematic method to guide such a choice. So, this tuning problem becomes more delicate and harder as the complexity of the controlled plant increases. Hence, the proposition of a systematic approach to tune the scaling factors of these particular PID-type FC structures is interesting.

In this study, a new approach based on the Particle Swarm Optimization (PSO) meta-heuristic technique is proposed for systematically tuning the scaling factors of the PID-type FC, both with and without self-tuning mechanisms. This work can be considered as an extension of the results given in [11, 12, 16, 26, 31]. The fuzzy control design is formulated as a constrained optimization problem which is efficiently solved based on a developed PSO algorithm. In order to specify more robustness and performance control objectives of the proposed PSO-tuned PID-type FC, different optimization criteria are considered and compared subject to several various control constraints defined in the time-domain framework.

The remainder of this chapter is organized as follows. In Section 2, the proposed fuzzy PID-type FC structures, both with and without self-tuning scaling factors mechanisms, are presented and discussed within the discrete-time framework. Two adaptive mechanisms for scaling factors tuning are especially adopted. The optimization-based problems of the PID-type FC scaling factors tuning are formulated in Section 3. The developed constrained PSO algorithm, used in solving the formulated problems, is also described. An external static penalty technique is used to deal with optimization constraints. Theoretical conditions for convergence algorithm and parameters choice are established, based on the stability theory of dynamic systems. Section 4 is dedicated to apply the proposed fuzzy control approaches on an electrical DC drive benchmark and a thermal process within an experimental real-time framework based on an Advantech PCI-1710 multi-functions board associated with a PC computer and MATLAB/Simulink environment. Performances on convergence properties of the proposed PSO and the used GAO algorithm, are compared for the known Integral Absolute Error (IAE) and the Integral Square Error (ISE) criterion cases. The real-time fuzzy controllers are developed through the compilation and linking stage, in a form of a Dynamic Link Library (DLL) which is, then, loaded in memory and started-up.

2. PID-type fuzzy control design

In this section, the considered PID-type FC structures are briefly described within the discrete-time framework based on [11, 12, 16, 26, 31].

2.1. Discrete-time PID-type FLC

Proposed by Qiao and Mizumoto in [26] within continuous-time formalism, this particular fuzzy controller structure, called PID-type FC, retains the characteristics similar to the conventional PID controller. This result remains valid while using a type of FC with triangular and uniformly distributed membership functions for the fuzzy inputs and a crisp output, a product-sum inference and a center of gravity defuzzification methods.

Under these conditions, the equivalent proportional, integral and derivative control components of such a PID-type FC are given by $\alpha K_e \mathcal{P} + \beta K_d \mathcal{D}$, $\beta K_e \mathcal{P}$, and $\alpha K_d \mathcal{D}$, respectively, as shown in [16, 26, 31]. In these expressions, $\mathcal{P}$ and $\mathcal{D}$ represent relative coefficients, K_e, K_d,

α and β denote the scaling factors associated to the inputs and output of the FC, as shown in Figure 1. The proof of this computation is shown with more details in [26].

When approximating the integral and derivative terms within the discrete-time framework, we can consider the closed-loop control structure for a discrete-time PID-type FC, as shown in Figure 1.

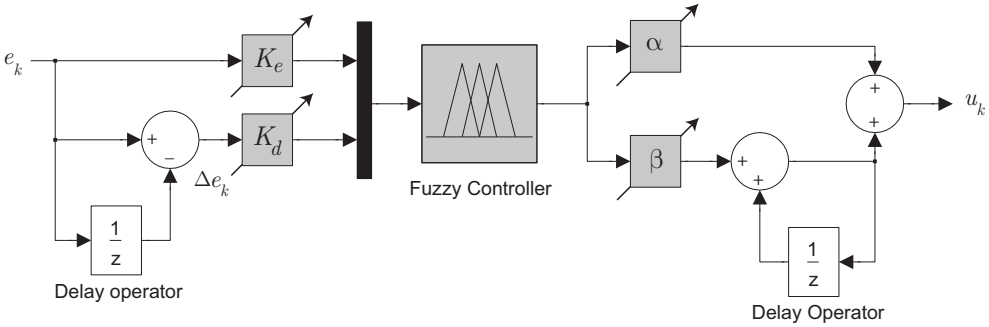

Figure 1. The proposed discrete-time PID-type FC structure.

As shown in [11, 16, 26, 31], the dynamic behaviour of this PID-type FC structure is strongly dependent on the scaling factors K_e, K_d, α and β, difficult and delicate to tune.

2.2. PID-type FC with self-tuning mechanisms

In order to improve the performances of the considered PID-type FC structure, various self-tuning mechanisms for scaling factors have been proposed in the literature. Two methods are especially adopted in this chapter.

2.2.1. Self-tuning via Empirical Functions Tuner Method EFTM

In this self-tuning method [31], the PID-type FC integral and derivative components updating are achieved based on scaling factors β and K_d, using the information on system's error as follows:

$$\begin{aligned} \beta_k &= \beta_0 \Phi(e_k) \\ K_{dk} &= K_{d0} \Psi(e_k) \end{aligned} \tag{1}$$

where β_0 and K_{d0} are the initial values of β and K_d, respectively, $\Phi(.)$ and $\Psi(.)$ are the empirical tuner functions defined, respectively, by:

$$\begin{aligned} \Phi(e_k) &= \phi_1 |e_k| + \phi_2 \\ \Psi(e_k) &= \psi_1 (1 - |e_k|) + \psi_2 \end{aligned} \tag{2}$$

In these equations, the parameters to be tuned ϕ_1, ϕ_2, ψ_1 and ψ_2 are all positive. The empirical function related to integral component decreases as the error decreases while the function related to derivative factor increases. Indeed, the objective of the function is to decrease the parameter with the change of error. However, the function has an inverse objective to make constant the proportional effect. Hence, the system may not always keep quick reaction against the error as demonstrated by Woo et al. in [31].

2.2.2. Self-tuning via Relative Rate Observer Method RROM

In this self-tuning method [12, 16], the PID-type FC integral and derivative components updating are achieved as follows:

$$\begin{aligned} \beta_k &= \frac{\beta_0}{K_f \delta_k} \\ K_{dk} &= K_{d0} K_{fd} K_f \delta_k \end{aligned} \tag{3}$$

where δ_k is the output of the fuzzy Relative Rate Observer (RRO) K_f is the output scaling factor for δ_k and K_{fd} is the additional parameter that affects only the derivative factor of the FC.

The rule-base for δ_k, as used by Eksin et al. [12] and Güzelkaya et al. [16], is considered for the fuzzy RRO. This fuzzy RRO block has as inputs the absolute values of error $|e_k|$ and the variable r_k, defined subsequently, as shown in Table 1.

$\lvert e_k \rvert / r_k$	S	M	F
S	M	M	L
SM	SM	M	L
M	S	SM	M
L	S	S	SM

Table 1. Fuzzy rule-base for the variable δ_k.

The linguistic levels assigned to the input $|e_k|$ and the output variable δ_k are as follows: L (Large), M (Medium), SM (Small Medium) and S (Small). For the input variable r_k, the following linguistic levels are assigned: F (Fast), M (Moderate) and S (Slow).

The variable r_k, defined in [12, 16] and called normalized acceleration, gives "relative rate" information about the fastness or slowness of the system response as shown in Table 2. It is defined as follows [16]:

$$r_k = \frac{\Delta e_k - \Delta e_{k-1}}{\Delta e^*} = \frac{\Delta\left(\Delta e_k\right)}{\Delta e^*} \tag{4}$$

where Δe_k and $\Delta\left(\Delta e_k\right)$ are the incremental change in error and the so-called acceleration in error given respectively by:

$$\Delta e_k = e_k - e_{k-1} \tag{5}$$

$$\Delta\left(\Delta e_k\right) = \Delta e_k - \Delta e_{k-1} \tag{6}$$

In equation (4), the variable Δe^* is chosen as follows:

$$\Delta e^* = \begin{cases} \Delta e_k & if \quad |\Delta e_k| \geq |\Delta e_{k-1}| \\ \Delta e_{k-1} & if \quad |\Delta e_k| < |\Delta e_{k-1}| \end{cases} \tag{7}$$

Δe^*	$\Delta\left(\Delta e_k\right)$	System response
Positive	Positive	Fast
Positive	Negative	Slow
Negative	Positive	Slow
Negative	Negative	Fast

Table 2. Nature of the system response depending on the variable r_k.

For this RROM self-tuning approach, the uniformly distributed triangular and the symmetrical membership functions, as shown in Figures 2, 3, 4, are assigned for the fuzzy inputs r_k and $|e_k|$, and fuzzy output variable δ_k. The view of the above fuzzy rule-base is illustrated in Figure 5.

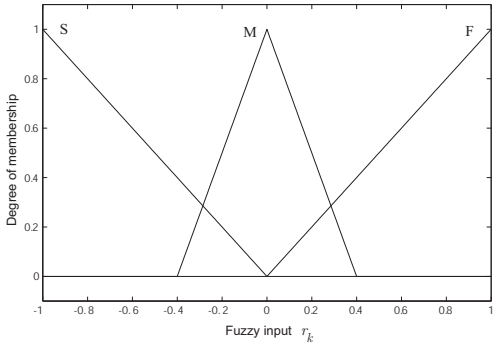

Figure 2. Membership functions for r_k.

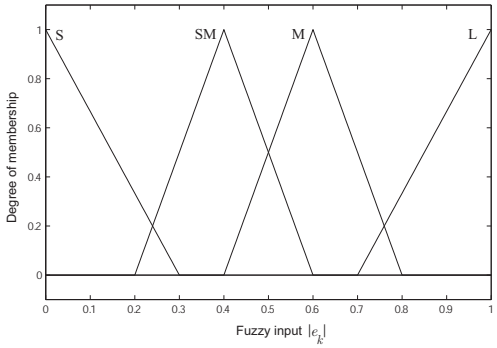

Figure 3. Membership functions for $|e_k|$.

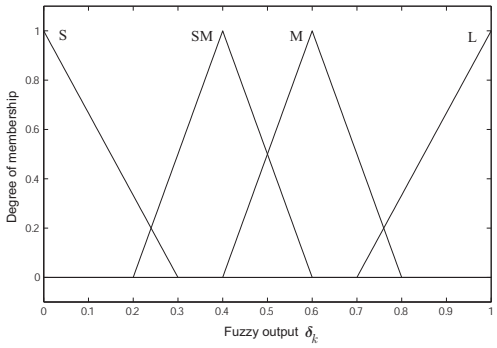

Figure 4. Membership functions for δ_k.

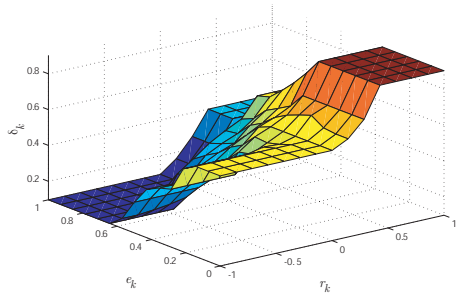

Figure 5. View of the fuzzy rule-base for δ_k.

3. The proposed PSO-based approach

In this section, the problem of scaling factors tuning, for all defined PID-type FC structures, is formulated as a constrained optimization problem which is solved using the proposed PSO-based approach.

3.1. PID-type FC tuning problem formulation

The choice of the adequate values for the scaling factors of each PID-type FC structure is often done by a trials-errors hard procedure. 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 like a promising solution. This tuning problem can be formulated as the following constrained optimization problem:

$$\begin{cases} \underset{\boldsymbol{x}\in\mathbb{D}}{minimize}\, f(\boldsymbol{x}) \\ subjectto \\ g_l(\boldsymbol{x}) \leq 0; \;\; \forall l = 1, \ldots, n_{con} \end{cases} \tag{8}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ the cost function, $\mathbb{D} = \{\boldsymbol{x} \in \mathbb{D}^m; \boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{\max}\}$ the initial search space, which is supposed containing the desired design parameters, and $g_l : \mathbb{R}^m \to \mathbb{R}$ the problem's constraints.

The optimization-based tuning problem consists in finding the optimal decision variables $\boldsymbol{x}^* = \left(x_1^*, x_2^*, \ldots, x_m^*\right)^T$, representing the scaling factors of a given PID-type FC structure, which minimize the defined cost function, chosen as the ISE and IAE performance criteria. These cost functions are minimized, using the proposed constrained PSO algorithm, under various time-domain control constraints such as overshoot D, steady state error E_{ss}, rise time t_r and settling time t_s of the system's step response, as shown in the equations (9), (10) and (11).

Hence, in the case of the PID-type FC structure without self-tuning mechanisms, the scaling factors to be optimized are K_e, K_d, α and β. The formulated optimization problem is defined as follows:

$$\begin{cases} \underset{\boldsymbol{x}=(K_e,K_d,\alpha,\beta)^T\in\mathbb{R}_+^4}{minimize} f(\boldsymbol{x}) \\ subjectto \\ D \leq D^{\max}; t_s \leq t_s^{\max}; t_r \leq t_r^{\max}; E_{ss} \leq E_{ss}^{\max} \end{cases} \tag{9}$$

where $D^{\max}$, $E_{ss}^{\max}$, $t_r^{\max}$ and $t_s^{\max}$ are the specified overshoot, steady state, rise and settling times respectively, that constraint the step response of the PSO-tuned PID-type FC controlled system, and can define some time-domain templates.

In the case of the PID-type FC structure with the EFTM self-tuning mechanism, the scaling factors to be optimized are φ_1, φ_2, ψ_1 and ψ_2. The formulated optimization problem is defined as follows:

$$\begin{cases} \underset{\boldsymbol{x}=(\varphi_1,\varphi_2,\psi_1,\psi_2)^T \in \mathbb{R}_+^4}{minimize} \; f(\boldsymbol{x}) \\ subjectto \\ D \leq D^{\max}; t_s \leq t_s^{\max}; t_r \leq t_r^{\max}; E_{ss} \leq E_{ss}^{\max} \end{cases} \tag{10}$$

For the PID-type FC structure with the RROM self-tuning mechanism, the scaling factors to be optimized are K_f and K_{fd}. The formulated optimization problem is defined as follows:

$$\begin{cases} \underset{\boldsymbol{x}=\left(K_f,K_{fd}\right)^T \in \mathbb{R}_+^2}{minimiser} \; f(\boldsymbol{x}) \\ subjectto \\ D \leq D^{\max}; t_s \leq t_s^{\max}; t_r \leq t_r^{\max}; E_{ss} \leq E_{ss}^{\max} \end{cases} \tag{11}$$

3.2. Particle Swarm Optimization technique

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.

3.2.1. Overview

The PSO technique is an evolutionary computation method developed by Kennedy and Eberhart [9]. This recent meta-heuristic technique is inspired by the swarming or collaborative behaviour of biological populations. The cooperation and the exchange of information between population individuals allow solving various complex optimization problems [10, 25, 27, 28, 30].

Without any regularity on the cost function to be optimized, 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. In comparison with other meta-heuristics, this optimization technique is a simple concept, easy to implement, and a computationally efficient algorithm [10, 27, 30]. The convergence and parameters selection of the PSO algorithm are proved using several advanced theoretical analysis proposed by Ruben and Kamran in [27] and Van den Bergh in [30]. Its stochastic behaviour allows overcoming the local minima problem.

Particle swarm optimisation has been enormously successful in several and various industrial domains [18, 19]. It has been used across a wide range of engineering applications. These applications can be summarized around domains of robotics, image and signal processing, electronic circuits design, communication networks, but more especially the domain of plant control design, as shown in [2–6].

3.2.2. Basic PSO algorithm

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

At each algorithm iteration, the i^{th} particle position, $\boldsymbol{x}^i \in \mathbb{R}^m$, evolves based on the following update rules:

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{x}_k^i + \boldsymbol{v}_{k+1}^i \tag{12}$$

$$\boldsymbol{v}_{k+1}^i = w_{k+1}\boldsymbol{v}_k^i + c_1 r_{1,k}^i \left(\boldsymbol{p}_k^i - x_k^i\right) + c_2 r_{2,k}^i \left(\boldsymbol{p}_k^g - x_k^i\right) \tag{13}$$

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 $[\![0, 1]\!]$,

$\boldsymbol{p}_k^i$: the best previously obtained position of the i^{th} particle,

$\boldsymbol{p}_k^g$: 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 Figure 6, for a two dimensional design space.

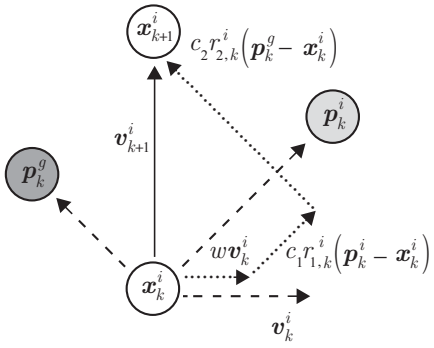

Figure 6. Particle position and velocity updates.

In order to improve the exploration and exploitation capacities of the proposed PSO algorithm, we choose for the inertia factor a linear evolution with respect to the algorithm iteration as given by Shi and Eberhart in [28]:

$$w_{k+1} = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{k_{\max}}\right) k \tag{14}$$

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.

Similarly to other meta-heuristic methods, the PSO algorithm is originally 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 (8) with penalties proportional to the degree of constraint infeasibility. In this paper, the following external static penalty technique is used:

$$\varphi(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{l=1}^{n_{con}} \chi_l \max\left[0, g_l(\boldsymbol{x})^2\right] \tag{15}$$

where χ_l is a prescribed scaling penalty parameters and n_{con} is the number of problem constraints $g_l(\boldsymbol{x})$.

Finally, the basic proposed PSO algorithm can be summarized by the following steps:

1. Define 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. Initialize the n_p particles with randomly chosen positions $\boldsymbol{x}_0^i$ and velocities $\boldsymbol{v}_0^i$ in the search space $\mathbb{D}$. Evaluate the initial population and determine $\boldsymbol{p}_0^i$ and $\boldsymbol{p}_0^g$.
3. Increment the iteration number k. For each particle apply the update equations (12) and (13), and evaluate the corresponding fitness values $\varphi_k^i = \varphi\left(\boldsymbol{x}_k^i\right)$:
   - if $\varphi_k^i \leq pbest_k^i$ then $pbest_k^i = \varphi_k^i$ and $\boldsymbol{p}_k^i = \boldsymbol{x}_k^i$;
   - if $\varphi_k^i \leq gbest_k$ then $gbest_k = \varphi_k^i$ and $\boldsymbol{p}_k^g = \boldsymbol{x}_k^i$;

   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 $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}_k^i}\left\{f\left(\boldsymbol{x}_k^i\right), \forall i, k\right\}$. Otherwise, go to step 3.

3.2.3. The convergence of PSO algorithm analysis

In this part, the proposed PSO algorithm is analysed based on results in [27, 30]. Theoretical conditions for convergence algorithm and parameters choice are established.

Let us replace the velocity update equation (13) into the position update equation (12) to get the following expression:

$$\boldsymbol{x}_{k+1}^i = \left(1 - c_1 r_{1,k}^i - c_2 r_{2,k}^i\right)\boldsymbol{x}_k^i + w\boldsymbol{v}_k^i + c_1 r_{1,k}^i \boldsymbol{p}_k^i + c_2 r_{2,k}^i \boldsymbol{p}_k^g \tag{16}$$

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

$$\boldsymbol{v}_{k+1}^i = -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right)\boldsymbol{x}_k^i + w\boldsymbol{v}_k^i + c_1 r_{1,k}^i \boldsymbol{p}_k^i + c_2 r_{2,k}^i \boldsymbol{p}_k^g \tag{17}$$

The obtained equations (16) and (17) can be combined and written in matrix form as:

$$\begin{bmatrix} \boldsymbol{x}_{k+1}^i \\ \boldsymbol{v}_{k+1}^i \end{bmatrix} = \begin{bmatrix} 1 - \left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \\ -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k^i \\ \boldsymbol{v}_k^i \end{bmatrix} + \begin{bmatrix} c_1 r_{1,k}^i & c_2 r_{2,k}^i \\ c_1 r_{1,k}^i & c_2 r_{2,k}^i \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_k^i \\ \boldsymbol{p}_k^g \end{bmatrix} \tag{18}$$

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

$$\hat{y}_{k+1} = \mathcal{M}\hat{y}_k + \mathcal{N}\hat{u}_k \tag{19}$$

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:

$$\hat{y}_k = \begin{bmatrix} \boldsymbol{x}_k^i \\ \boldsymbol{v}_k^i \end{bmatrix}; \hat{u}_k = \begin{bmatrix} \boldsymbol{p}_k^i \\ \boldsymbol{p}_k^g \end{bmatrix}; \mathcal{M} = \begin{bmatrix} 1 - \left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \\ -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \end{bmatrix}; \mathcal{N} = \begin{bmatrix} c_1 r_{1,k}^i & c_2 r_{2,k}^i \\ c_1 r_{1,k}^i & c_2 r_{2,k}^i \end{bmatrix} \tag{20}$$

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 constants from the k^{th} to the $(k+1)^{th}$ iteration, given the following equilibrium state:

$$\hat{y}_{k+1} - \hat{y}_k = \begin{bmatrix} -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w \\ -\left(c_1 r_{1,k}^i + c_2 r_{2,k}^i\right) & w - 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k^i \\ \boldsymbol{v}_k^i \end{bmatrix} + \begin{bmatrix} c_1 r_{1,k}^i & c_2 r_{2,k}^i \\ c_1 r_{1,k}^i & c_2 r_{2,k}^i \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_k^i \\ \boldsymbol{p}_k^g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{21}$$

which is true only when:

$$\begin{aligned} \boldsymbol{x}_k^i &= \boldsymbol{p}_k^i = \boldsymbol{p}_k^g, \\ \boldsymbol{v}_k^i &= 0 \end{aligned} \tag{22}$$

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

$$\hat{y}_{eq} = \left[\boldsymbol{p}_k^g, 0\right]^T \tag{23}$$

So, the dynamic behaviour of the i^{th} particle can be analysed using the eigenvalues derived from the dynamic matrix formulation (19) and (20), solutions of the following characteristic polynomial:

$$\lambda^2 - \left(1 + w - c_1 r_{1,k}^i - c_2 r_{2,k}^i\right)\lambda + w = 0 \tag{24}$$

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

$$\begin{aligned} &|w| < 1 \\ &c_1 r_{1,k}^i + c_2 r_{2,k}^i > 0 \\ &w + 1 - \tfrac{c_1 r_{1,k}^i + c_2 r_{2,k}^i}{2} > 0 \end{aligned} \tag{25}$$

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:

$$\begin{aligned} &0 < c_1 + c_2 < 4 \\ &\tfrac{c_1 + c_2}{2} - 1 < w < 1 \end{aligned} \tag{26}$$

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

In order to illustrate the efficiency of the proposed PSO algorithm in the resolution of problems (9), (10) and (11), several comparisons with the Genetic Algorithms Optimization GAO-based method [14, 29] are considered. The next section is dedicated to the application of the proposed PSO-tuned PID-FC approaches to an electrical DC drive and a thermal process within a developed real-time framework.

4. Real-time control approach implementation

In this section, all designed PSO-tuned PID-type FC structures are applied to two different systems such as an electrical DC drive and a thermal PT-326 Process Trainer benchmarks. Real-time implementations and experimental results of these control laws are presented and discussed.

4.1. Control of an electrical DC drive benchmark

4.1.1. Plant model description

The considered benchmark is a 250 watts electrical DC drive, as shown in Figure 15. The machine's speed 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). This is achieved using a data acquisition and control system based on a PC computer and a multi-functions data acquisition PCI-1710 board which is compatible with MATLAB/Simulink [1, 17].

The considered electrical DC drive can be described by the following model that is used in the design setup:

$$G(s) = \frac{k_m}{(1+\tau_m s)(1+\tau_e s)} \tag{27}$$

The model's parameters are obtained by an experimental identification procedure and they are summarized in Table 3 with their associated uncertainty bounds. Also, this model is sampled with 10 ms sampling time for simulation and experimental setups.

Parameters	Nominal values	Uncertainty bounds
k_m	0.05	75 %
τ_m	300 ms	75 %
τ_e	14 ms	75 %

Table 3. Identified DC drive model parameters.

4.1.2. Simulation results

For all proposed PSO-tuned PID-type FC structures, product-sum inference and center of gravity defuzzification methods are adopted for the FC block. Uniformly distributed and symmetrical membership functions, are assigned for the fuzzy input and output variables. The associated fuzzy rule-base is given in Table 4.

e_k / Δe_k	N	Z	P
N	NB	N	Z
Z	N	Z	P
P	Z	P	PB

Table 4. Fuzzy rule-base for the output u_{fz}.

The linguistic levels assigned to the input variables e_k and Δe_k, and the output variable u_{fz} are given as follows: N (Negative), Z (Zero), P (Positive), N (Negative), NB (Negative Big) and PB (Positive Big). The view of this rule-base is illustrated in Figure 7.

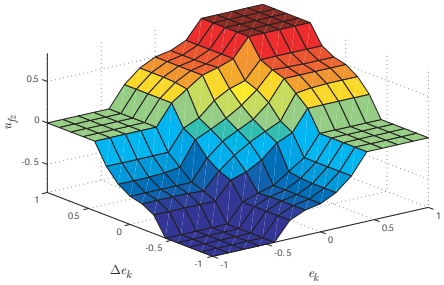

Figure 7. View of fuzzy rule-base for the fuzzy output u_{fz}.

The swarm size algorithm's choice is generally a problem-dependent in PSO framework. However, Eberhart and Shi [10] as well as Poli et al. [25] show that this parameter is often set empirically in relation to the dimensionality and perceived difficulty of a considered optimization problem. They suggest that swarm size values in the range 20-50 are quite common. For this purpose, we have tested the proposed PSO algorithm with different values in this range for the case of PID-type FC structure without self-tuning mechanisms. Globally, all the found results are close to each other. But, best values of the fitness are obtained while using a swarm size equal to 30. Henceforth, this value will be adopted for our following 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 so to validate the proposed approach. We run the proposed algorithm 20 times and feasible solutions are found in 98 % of trials and within an acceptable CPU computation time for the IAE and ISE criterion cases. The obtained optimization results are summarized in Tables 5, 6 and 7. Besides, the fact that the algorithm's 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 performances comparison of PSO- and GAO-based approaches is achieved in the same conditions.

Cost function	Algorithm	Best	Mean	Worst	St. dev.
ISE	PSO	0.0193	0.0304	0.0511	0.018
ISE	GAO	0.1200	0.1780	0.2410	0.050
IAE	PSO	0.0162	0.0261	0.0497	0.016
IAE	GAO	0.1892	0.2835	0.3227	0.066

Table 5. Optimization results from 20 trials of problem (9).

Cost function	Algorithm	Best	Mean	Worst	St. dev.
ISE	PSO	0.0660	0.0765	0.1030	0.015
ISE	GAO	0.0820	0.0912	0.1330	0.012
IAE	PSO	0.0659	0.0838	0.0946	0.014
IAE	GAO	0.0718	0.0814	0.0973	0.013

Table 6. Optimization results from 20 trials of problem (10).

Cost function	Algorithm	Best	Mean	Worst	St. dev.
ISE	PSO	0.0559	0.0792	0.0840	0.013
ISE	GAO	0.0822	0.0936	0.1120	0.015
IAE	PSO	0.0673	0.0861	0.0993	0.016
IAE	GAO	0.0855	0.0905	0.1009	0.008

Table 7. Optimization results from 20 trials of problem (11).

Indeed, the population size, used in the GAO algorithm, is set as 30 individuals and the maximum generation number is 50. However, the GA parameters, used for MATLAB simulations, are chosen as the Stochastic Uniform selection and the Gaussian mutation methods. The Elite Count is set as 2 and the Crossover Fraction as 0.8. The algorithm stops when the number of generations reaches the specified value for the maximum generation.

According to the statistical analysis of Tables 5,6 and 7, we can conclude that the proposed PSO-based approach produces better results in comparison with the standard GAO-based one. Also, while using a Pentium IV, 1.73 GHz and MATLAB 7.7.0, the CPU computation times are about 358 and 364 seconds for ISE and IAE criteria, respectively, for the considered PID-type FC without self-tuning mechanisms structure.

On the other hand, performances on convergence properties of the proposed PSO and the used GAO algorithm, in term of iterations number's required to find the best solution, are compared for the IAE criterion case, as shown in Figures 8 and 9. While using the proposed PSO-based method, we succeed to obtain the optimal solution within only about 28 iterations. However, the GAO-based method finds the same result after 40 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 of the GAO-based approach.

In a typical optimization procedure, the scaling parameters χ_l, given in equation (15), will be linearly increased at each iteration step so constraints are gradually enforced. Generally, the quality of the solution will directly depend on the value of the specified scaling parameters. In this paper and in order to make the proposed approach simple, great and constant scaling penalty parameters, equal to 10^3, are used for simulations. Indeed, simulation results show that with a great values of χ_l, the control system performances are weakly degraded and the effects on the tuning parameters are less meaningful. The PSO algorithm convergence is faster than the case with linearly variable scaling parameters.

The robustness of the proposed PSO algorithm convergence, under variation of the cognitive, social and inertia factor parameters, is analysed based on numerical simulations as shown

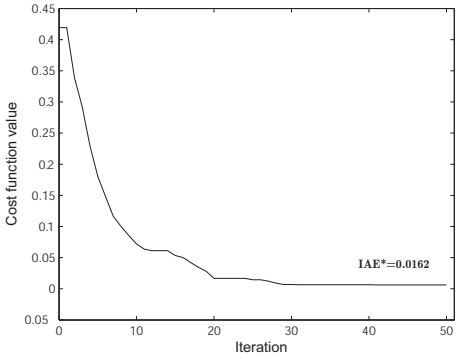

Figure 8. Convergence properties of the proposed PSO algorithm: IAE criterion case.

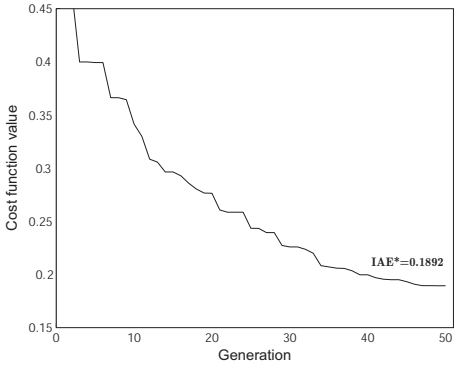

Figure 9. Convergence properties of the standard GAO algorithm: IAE criterion case.

in Figure 10 and Figure 11. The PSO algorithm's convergence is guaranteed within the established domain given by the equation (26).

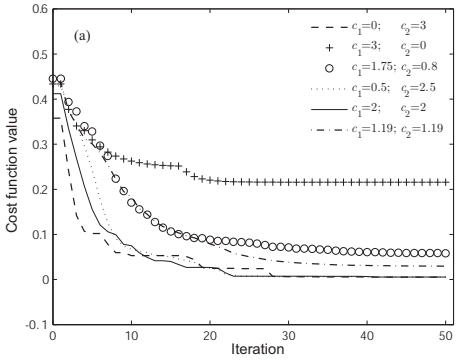

Figure 10. Robustness of the proposed PSO algorithm under variations of the cognitive and social parameters: IAE criterion case.

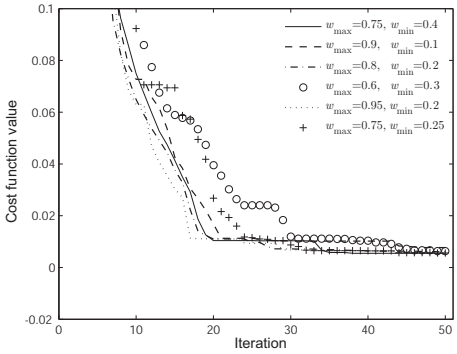

Figure 11. Robustness of the proposed PSO algorithm under variations of the inertia factor: IAE criterion case.

The robust stability of the proposed PSO-tuned PID-type FC approach is analysed while considering external disturbances and model uncertainties. According to uncertain bounds on nominal plant parameters, given in Table 1, we are going to consider the following family of continuous-time transfer functions supposed including the real studied plant:

$$\mathcal{G} = \left\{ \hat{G}(s) = \frac{k_m}{(1+\tau_m s)(1+\tau_e s)}; k_m \in \left[k_m^{\min}, k_m^{\max}\right], \tau_e \in \left[\tau_e^{\min}, \tau_e^{\max}\right], \tau_m \in \left[\tau_m^{\min}, \tau_m^{\max}\right] \right\} \tag{28}$$

Figure 12 shows the step responses of a family of 5 random generated closed-loop uncertain models. The stability robustness of the uncertain plants, under the above considered uncertainty types, is guaranteed for all designed PID-type FC structures.

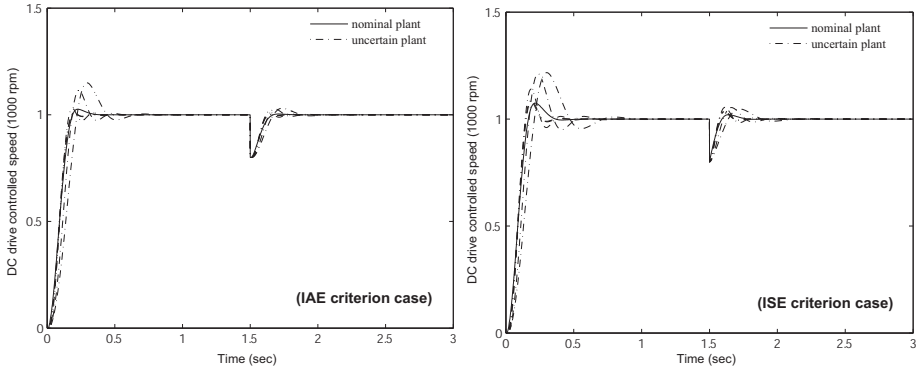

Figure 12. Stability robustness of the PSO-tuned PID-type fuzzy controlled system under model parameters uncertainties and external disturbances.

Finally, the time-domain performances of all proposed PID-type FC structures, are compared for the PSO- and GAO-based design cases as shown in Figure 13.

Besides, Table 8 shows the superiorities of the self-tuning EFTM and RROM PID-type FC structures in relation to the one without self-tuning mechanisms as verified in [16]. Remenber that the considered time-domain constraints for the PID-type FC tuning problems (9), 10)

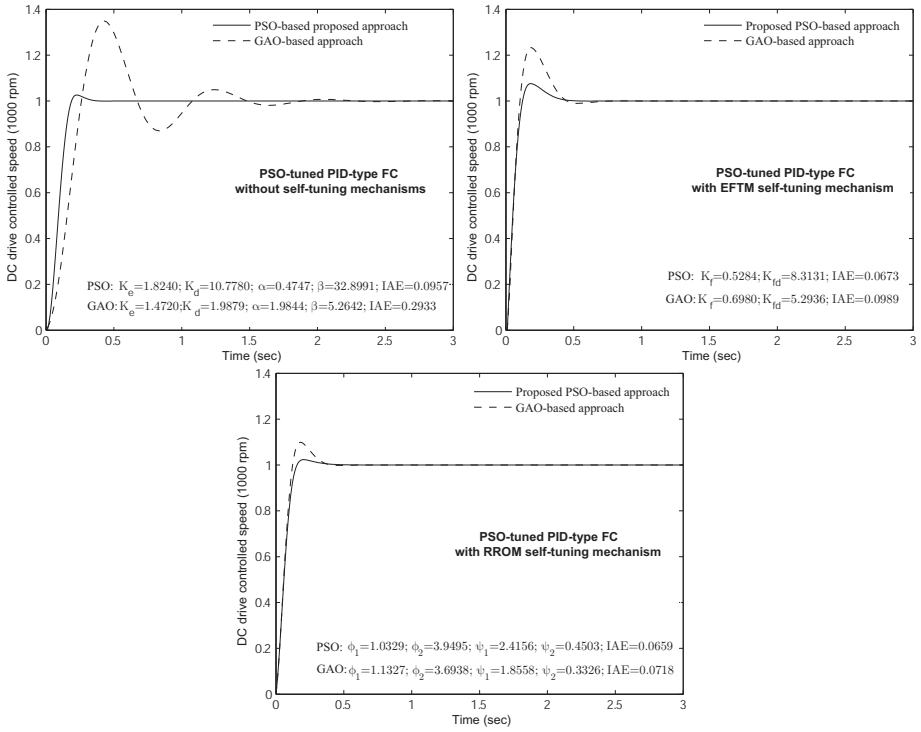

Figure 13. Time-domain performances comparison of all designed PSO- and GAO-tuned PID-type FC structures: IAE criterion case.

and (11) problems, defined in terms of overshoot, steady state, rise and settling times, have been specified as $D^{\max} = 20\%$, $E_{ss}^{\max} = 0$, $t_r^{\max} = 0.25\,\text{sec}$ and $t_s^{\max} = 0.75\,\text{sec}$.

PSO-tuned PID-type FC structure	$D(\%)$	t_r(sec)	t_s (sec)	E_{ss}	CPU computation time (sec)
without self-tuning mechanisms	17.5	0.23	0.49	0	364
with EFTM self-tuning mechanism	15	0.21	0.64	0	370
with RROM self-tuning mechanism	7	0.20	0.68	0	392

Table 8. Performances of the PSO-tuned PID-type FC structures: IAE criterion case.

4.1.3. Experimental setup and results

In order to illustrate the efficiency of the proposed PSO-tuned fuzzy control structures within a real-time framework, the example of the PID-type FC without self-tuning mechanism is considered. The same principle of implementation remains valid for the other PID-type FC structures.

The controlled process is constituted by the single-phase AC-DC power converter and the independent excitation DC motor. A schematic diagram of the experimental setup prepared for testing of the designed controller is shown in Figure 14. The developed experimental

benchmark is given by Figure 15. The designed real-time application acquires input data (speed of the DC drive) and generates control signals for the AC-DC power converter through a thyristors gate drive circuit. This is achieved using a data acquisition and control system based on PC and a multi-function data acquisition PCI-1710 board with 12-bit resolution of A/D converter and up to 100 KHz sampling rate. A thyristors gate drive circuit, based on a multivibrator, is used to generate a triggering burst of high-frequency impulses. A pulse transformer is used to assure the galvanic insulation between the control and power circuits. The acquired speed measure, obtained from tachometer sensor, must be adapted to be applied to the used multi-function PCI-1710 board. The complete electronic circuit diagram of the designed control system is given in [1, 17].

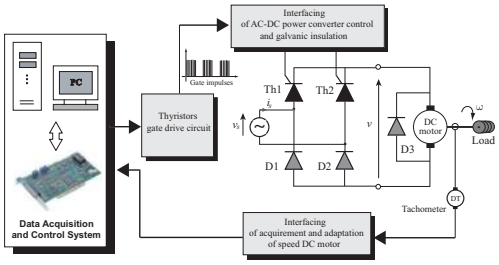

Figure 14. The proposed experimental setup schematic.

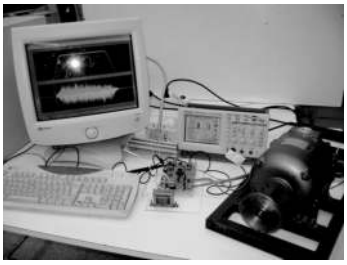

Figure 15. The developed experimental DC drive benchmark.

The multi-function data acquisition PCI-1710 board allows achieving measurement and controlling functions. This target is used to create a real-time application to let the implemented controller system run while synchronized to a real-time clock. The model of the plant was removed from the simulation model, and instead of it, the input device driver (Analog Input) and the output device driver (Analog Output) were introduced as shown in Figure 17. These device drivers close the feedback loop when moving from simulations to experiments. Device driver's blocks include procedures to access the inputs-outputs board. The real-time controller is developed through the compilation and linking stage, in a form of a Dynamic Link Library (DLL) which is then loaded in memory and started-up

The practical implementation of the PSO-tuned PID-type FC approach leads to the experimental results of Figure 17 and Figure 18. The obtained results are satisfactory for a simple, systematic and non-conventional control approach and point out the controller's viability and validate the proposed control approach. The measured speed tracking error of the controlled DC drive is very small (less than 10 % of set point) showing the high performances of the proposed control especially in terms of tracking. On the other hand,

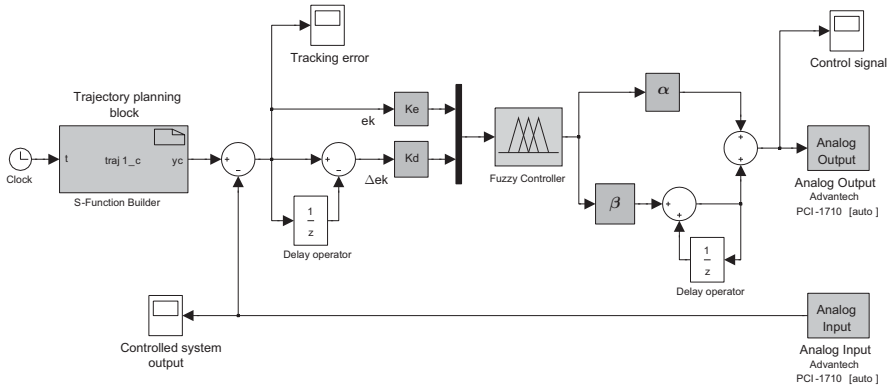

Figure 16. PCI-1710 board based real-time implementation of the proposed PSO-tuned PID-type FC structure.

Figure 18 shows the robustness of the proposed PSO-tuned PID-type FC in rejection of an external load disturbance applied on the controlled system. The dynamic of the disturbance rejection is fast and guaranteed.

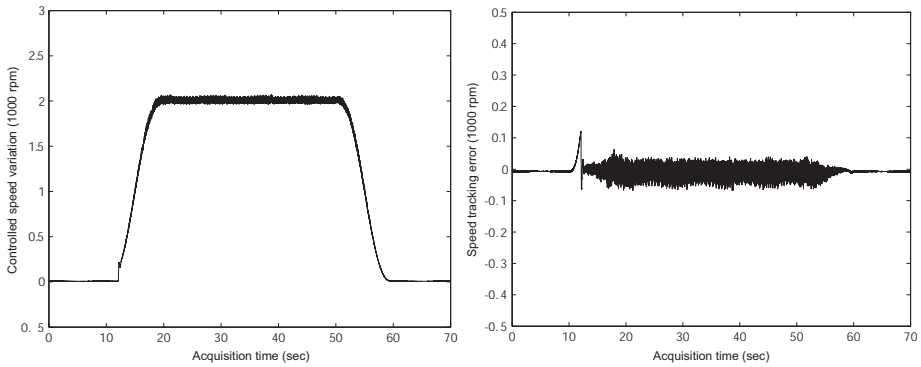

Figure 17. Experimental results of PSO-tuned PID-type FC implementation: fuzzy controller tracking performances.

4.2. Control of a thermal process benchmark

4.2.1. Plant model description

The thermal process to be controlled, shown in the photography of Figure 20, is based on a known PT-326 Process trainer [13], initially developed with an analog control system and modified in order to be digitally controlled. To power the heating resistor, a single-phase AC-AC converter, is developed [7].

In this prototype, the air drawn from atmosphere by a centrifugal blower is injected, through a heating element, in a polypropylene pipe, and rejected in the atmosphere. The amount of air

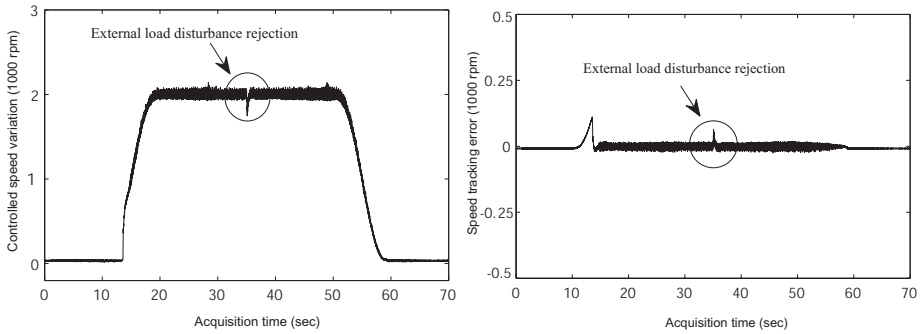

Figure 18. Experimental results of PSO-tuned PID-type FC implementation: fuzzy controller robustness under external load disturbance.

flowing in the pipe can be adjusted by the mean of an inlet throttle attached to the blower. The process consists of heating the air flowing in the pipe to the desired temperature level. The digital control system generates a 40W signal which determines the amount of electrical power supplied to heating resistor made of $10K\Omega/7W$ power resistors. According to these settings, the experimental trials show that the controlled air temperature can be varied up to 20°C from the ambient temperature. The assigned control objective is to regulate the temperature of the air at a desired level, with high tracking performance and under internal disturbances, like model parameters variation, and output disturbances. The temperature sensor can be placed at three different locations on the path of the air flow. A variation in the temperature makes a voltage variation at the sensor's output. The amount of air trough the pipe, adjusted by setting the opening of the throttle, can also be used to generate an output disturbance, in order to test the efficiency of the proposed control system.

The controlled system input is the voltage applied to the AC-AC power electronic circuit feeding the heating resistor, and the output is the air flow temperature in the pipe, expressed by a 50 mV/°C voltage, obtained after amplification of the LM35 temperature sensor's output signal. As shown in [23, 32], this process can be characterized as a non-linear system with a pure time delay. The pure time delay depends on the position of the temperature sensor element inserted into the air stream at any one of the three positions along the tube. When the temperature in the air volume inside the tube is assumed uniform a linear model can be obtained. To identify a numerical model of the considered plant, some experimental trials lead to consider the following transfer function, between the heater input voltage and the sensor's output voltage:

$$G(s) = \frac{k_m}{1+\tau s} e^{-\tau_d s} \tag{29}$$

where k_m is the DC gain system, τ is the time constant system, and τ_d is the time delay system.

This obtained plant model is assumed to be the nominal one and will be adopted in PSO-tuned PID-type FC synthesis step. These model's parameters are obtained by an experimental identification procedure and they are summarized in Table 9 with their associated uncertainty bounds. Also, this model is sampled with 2 sec sampling time for simulation and experimental setups.

Parameters	Nominal values	Uncertainty bounds
k_m	20	50 %
τ	65 sec	50 %
τ_d	1 sec	50 %

Table 9. Identified Thermal Process model parameters.

For this PSO-tuned PID-type fuzzy control example, we represent only the obtained experimental results. For the numerical simulations step, both IAE and ISE criterion, used for the electrical DC drive control, are investigated for this thermal process example. Same problems (9), (10) and (11) are considered and resolved by the developed constrained PSO algorithm.

4.2.2. Experimental setup and results

The developed real-time application acquires air temperature measure and generates control signals for the triac of AC-AC power converter through a gate drive circuit, as shown in Figure 19. This is achieved using a control system based on PC and the used multi-function data acquisition PCI-1710 board. A triac gate drive circuit is used to generate a Pulse Width Modulation (PWM) control signal synchronized with the zero-crossing of the AC voltage. The acquired air temperature measure is scaled before being applied to the used multi-function PCI-1710 board used to create a real-time application to let the implemented controller system run while synchronized to a real-time clock. This leads to experimental results shown in Figure 21 and Figure 22.

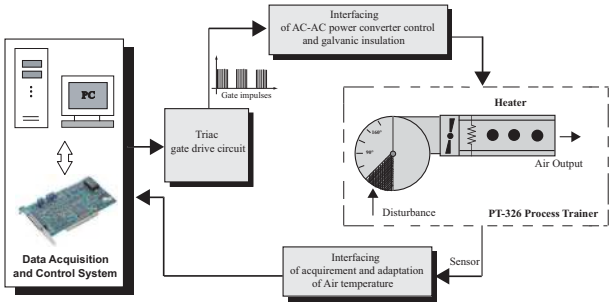

Figure 19. The proposed thermal process experimental setup schematic.

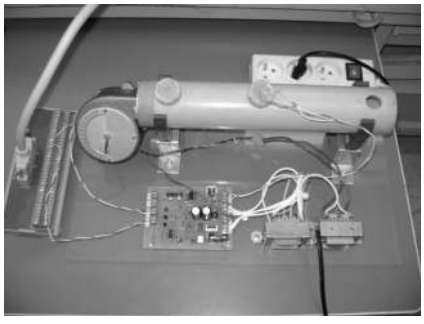

Figure 20. Developed experimental benchmark of the PT-326 Process Trainer.

As shown in Figure 21 and Figure 22, the controlled air temperature of the considered thermal process tracks the desired trajectory with high performance in terms of response speed and precision in the two considered cases. The robustness of the proposed control strategy in term of output static disturbance rejection, which caused by the throttle opening, is improved.

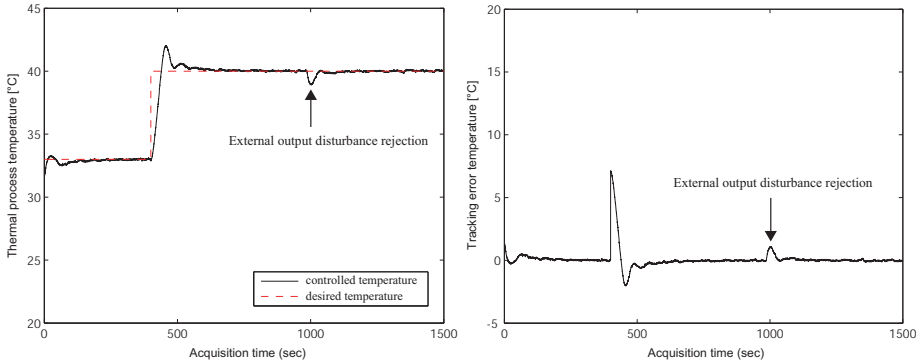

Figure 21. Experimental result of PSO-tuned fuzzy controlled PT-326 Process Trainer: IAE criterion case.

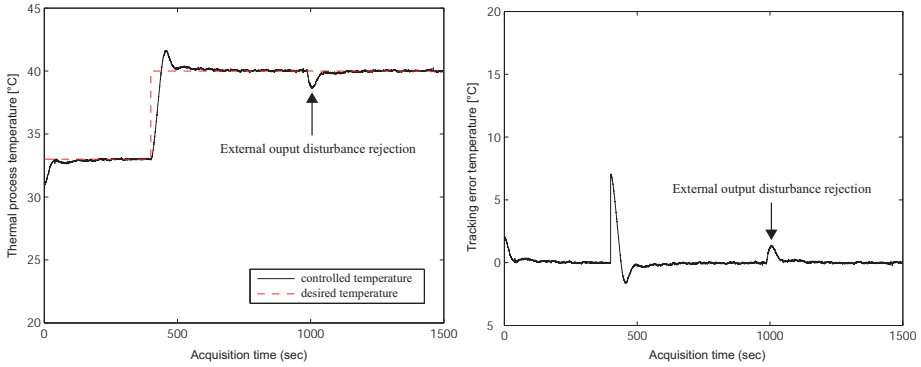

Figure 22. Experimental result of PSO-tuned fuzzy controlled PT-326 Process Trainer: ISE criterion case.

5. Conclusion

In this study, a new method for tuning PID-type FC structures, using a constrained PSO-based technique, is proposed and successfully applied to an electrical DC drive and thermal process within a real-time framework. This efficient tool leads to a robust and systematic fuzzy control design approach. The performances 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 practical implementation of the PSO-tuned PID-type FC approach, for the considered electrical DC drive and the thermal PT-326 Process Trainer benchmarks, leads to several

satisfactory experimental results showing the high performances of the proposed control especially in terms of tracking and robustness.

The PSO-tuned PID-type FC structures robustness, under external influences such as the output static disturbances and parametric uncertainties, are proven. The control design methodology is systematic, practical and simple without need to exact analytic plant model description. The obtained simulation and experimental results show the efficiency in terms of performance and robustness of the proposed fuzzy control approach which can be applied in industrial motor control field.

Author details

S. Bouallègue
Higher Insitute of Industrial Systems of Gabes (ISSIG), Salaheddine Elayoubi Street, 6032 Gabes, Tunisia

J. Haggège and M. Benrejeb
National Engineering School of Tunis (ENIT), BP 37, le Belvédère, 1002 Tunis, Tunisia

All authors are with the Research Laboratory in Automatic Control (LA.R.A) of ENIT.

6. References

[1] Bouallègue, S., Haggège, J., Benrejeb, M., (2012). On a robust real-time $\mathcal{H}_\infty$ controller design for an electrical drive, International Journal of Modelling, Identification and Control, 15 (2), pp. 89-96.

[2] Bouallègue, S., Haggège, J., Ayadi, M., Benrejeb, M., (2012). PID-type fuzzy logic controller tuning based on particle swarm optimization, Engineering Applications of Artificial Intelligence, 25 (3), pp. 484-493.

[3] Bouallègue, S., (2011). Optimisation par essaim particulaire de lois de commande robuste : théorie et mise en œuvre pratique, Editions Universitaires Européennes, ISBN : 978-613-1-59335-2, Saarbrücken, Germany.

[4] Bouallègue, S., Haggège, J., Benrejeb, M., (2010). Structured Loop-Shaping $\mathcal{H}_\infty$ Controller Design using Particle Swarm Optimization. Proceedings of the 2010 IEEE International Conference on Systems, Man, and Cybernetics SMC'10, Istanbul.

[5] Bouallègue, S., Haggège, J., Benrejeb, M., (2010). Structured Mixed-Sensitivity $\mathcal{H}_\infty$ Design using Particle Swarm Optimization. Proceedings of the 7th IEEE International Multi-Conference on Systems, Signals and Devices SSD'10, Amman.

[6] Bouallègue, S., Haggège, J., Benrejeb, M., (2011). Particle Swarm Optimization-Based Fixed-Structure $\mathcal{H}_\infty$ Control Design. International Journal of Control, Automation, and Systems 9 (2), pp. 258-266.

[7] Bouallègue, S., Haggège,J., and Benrejeb, M., (2009). Real-Time $\mathcal{H}_\infty$ Control Design for a Thermal Process, Proceedings of the 10th International conference on Sciences and Techniques of Automatic control & computer engineering STA'2009-ACS, pp. 949-958, Hammamet, Tunisia.

[8] Bühler, H., (1994). Réglage par logique floue, Presses polytechniques et universitaires romandes, Lausanne.

[9] Eberhart, R.C., Kennedy, J., (1995). A New Optimizer Using Particle Swarm Theory. Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, pp. 39-43.
[10] Eberhart, R.C., Shi, Y., (2001). Particle Swarm Optimization: Developments, Applications and Resources. Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, Korea, pp. 81-86.
[11] Eker, I., Torun, Y., (2006). Fuzzy logic control to be conventional method. Energy Conversion and Management 47, pp. 377-394.
[12] Eksin, I., Güzelkaya, M., Gürleyen, F., (2001). A new methodology for deriving the rule-base of a fuzzy logic controller with a new internal structure. Engineering Applications of Artificial Intelligence 14, pp. 617-628.
[13] Feedback, (1980). Process Trainer PT326, Feedback Instruments Limited, Crowborough, United Kingdom.
[14] Goldberg, D.E., (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.
[15] Grigorie, L. (editor), (2011). Fuzzy Controllers, Theory and Applications, ISBN 978-953-307-543-3, InTech, Croatia.
[16] Güzelkaya, M., Eksin, I., Yesil, E., (2003). Self-tuning of PID-type fuzzy logic controller coefficients via relative rate observer. Engineering Applications of Artificial Intelligence 16, pp. 227-236.
[17] Haggège, J., Ayadi, M., Bouallègue, S., Benrejeb, M., (2010). Design of Fuzzy Flatness-based Controller for a DC Drive. Control and Intelligent Systems 38 (3), pp. 164-172.
[18] Korosec, P., (editor), (2010). New Achievements in Evolutionary Computation, ISBN 978-953-307-053-7, 2010, InTech, Croatia.
[19] Lazinica, A. (editor), (2009). Particle Swarm Optimization, ISBN 978-953-7619-48-0, InTech, Croatia.
[20] Lee, C.C., (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller. Part I. IEEE Transactions on Systems, Man, and Cybernetics 20 (2), pp. 404-418.
[21] Lee, C.C., (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller. Part II. IEEE Transactions on Systems, Man, and Cybernetics 20 (2), pp. 419-435.
[22] Mamdani, E.H., (1974). Application of fuzzy logic algorithms for control of simple dynamic plant, Proceedings of the Institute of Electrical Engineering (IEE), 121 (12), pp.1585-1588
[23] Mohd, R., Yeoh Keat, H., Sahnius, U., Norhaliza, A.W., (2007). Modelling of PT326 Hot Air Blower Trainer Kit Using PRBS Signal and Cross-Correlation Technique, Jurnal Teknologi, vol. 42, pp. 9-22.
[24] Passino, K.M., Yurkovich, S., (1998). Fuzzy Control. Addison Wesley Longman, Menlo Park, California.
[25] Poli, R., Kennedy, J., Blackwell, T., (2007). Particle Swarm Optimization: An Overview. Swarm Intelligence, Springer 1, pp. 33-57.
[26] Qiao, W.Z., Mizumoto, M., (1996). PID type fuzzy controller and parameters adaptive method. Fuzzy Sets and Systems 78, pp. 23-35.
[27] Ruben, E.P., Kamran, B., (2007). 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.

[28] Shi, Y., Eberhart, R., (1999). Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation, Washington, pp. 1945-1950.
[29] The MathWorks Inc., (2009). Genetic Algorithm and Direct Search ToolboxTM: User's Guide. Natick.
[30] Van den Bergh, F., (2006). An Analysis of Particle Swarm Optimizers. PhD Thesis, University of Pretoria, Pretoria, South Africa.
[31] Woo, Z-W., Chung, H-Y., Lin, J-J., (2000). A PID type fuzzy controller with self-tuning scaling factors. Fuzzy Sets and Systems 115, pp. 321-326.
[32] Yesil, E., Güzelkaya, M., Eksin, I., Tekin, O.A., (2008). Online Tuning of Set-point Regulator with a Blending Mechanism Using PI Controller, Turk J Elec Engin, 16 (2), pp. 143-157.
[33] Zadeh, L.A., (1994). Soft computing and fuzzy logic, IEEE Software, 11 (6), pp. 48-56.
[34] Zadeh, L.A., (1965). Fuzzy Sets, Information and Control, 8, pp. 338-353.