Noname manuscript No.
(will be inserted by the editor)

An adaptive control study for the DC motor using meta-heuristic algorithms

Alejandro Rodríguez-Molina[1] · Miguel Gabriel Villarreal-Cervantes[1*] · Mario Aldape-Pérez[1]

Received: date / Accepted: date

Abstract In this work, a comparative study of different meta-heuristic techniques in the adaptive control for the speed regulation of the DC motor with parameters uncertainties is presented. The adaptive control is established as the on-line solution of a constrained dynamic optimization problem. Several adaptive strategies based on Differential Evolution (DE), Particle Swarm Optimization (PSO), Bat Algorithm (BAT), Firefly Algorithm (FFA), Wolf Search Algorithm (WSA) and Genetic Algorithm (GA) are proposed in order to on-line tune the parameters of the DC motor control. Simulation results show that proposed adaptive control strategies are a viable alternative to regulate the speed of the motor subject to different operation scenarios. The statistical analysis given in this work shows the features and the differences among strategies, their feasibility to set them up experimentally and also a new hybrid strategy to efficiently solve the problem. In addition, comparative analysis with a robust control approach reveal the advantages of the adaptive strategy based on meta-heuristic techniques in the velocity regulation of the DC motor.

Keywords Adaptive control · heuristic techniques · optimization problem · parameter estimation

Alejandro Rodríguez-Molina, Miguel G. Villarreal-Cervantes, Mario Aldape-Pérez
Tel.: +52 55 57296000
arodriguezm0902@alumno.ipn.mx,
{mvillarrealc, maldape}@ipn.mx
[1] Posgraduate Department
Instituto Politécnico Nacional, CIDETEC
Mexico city, 07700, Mexico.

1 Introduction

The DC motors are base elements in almost any engineering application that requires movement. These elements are presented in vehicles (Bitar et al. 2015), in robotic manipulators (Li et al. 2013) and in many industrial tools. The use of DC motors has several advantages, some of them are: their operation simplicity by varying their input voltage, their relatively low cost and the great diversity of designs that can fit to any prototype. Almost all applications that use DC motors require an accurate operation (Yang and Chou 2009) and sometimes, an efficient operation in terms of energetic consumption (Raslavičius et al. 2017). Nevertheless, one of the main problems faced by engineers is the presence of parametric uncertainties. Parametric uncertainties are due to several causes such as a variational operating environment, the presence of noise and the wear of the plant after a long and continuous use. Parametric uncertainties may be variations in DC motor parameters, noise in the input/output signal, etc., usually unpredictable and responsible of inaccuracy in the control system. In (Mao et al. 2003) for example, the friction induced by a DC motor had a negative effect in the positioning accuracy of an aero-static slider.

Over time, many efforts in search for a solution to this problem have been made. One way to handle the effect of parametric uncertainties when their behavior is well known (deterministic parametric uncertainties), is by designing an optimal control system or by a correct off-line adjusting of its parameters (Hashem Zadeh et al. 2016). However, this kind of solutions is not so efficient when there are no confidence about the behavior of the uncertainties.

An approach that deal with a set of bounded uncertainties (Liu and Yao 2016) is the robust control approach where the information about the differences between the actual system and the system model (model uncertainty) are used to design a controller. Till the date, many robust control algorithms have been developed for the speed regulation task of a DC motor (Song and Jia 2016; Kim et al. 1997; Linares-Flores et al. 2012; Orozco et al. 2012). Despite the high performance of robust controllers under unknown disturbances, the main disadvantage of this approach lies in the assumption of bounded uncertainties behavior.

Another way to handle parametric uncertainties is by using the adaptive control approach. Adaptive control refers to estimate the plant parameters on-line using the feedback of the system (Landau et al. 2011). Then, estimated parameters are used in calculating the control signal that allows a desired system operation. Unlike robust control approach, adaptive control is not limited by or does not need information about the uncertainties bounds.

Nowadays, there are many approaches from the adaptive control theory. In no-linear control, several adaptive control laws have been designed and are based on the system model (Slotine and Li 1991) and some controller tuning techniques have been developed to improve the effectiveness in control systems subject to uncertainties (Yavuz et al. 2012). In (Kwan et al. 1996) an adaptive control law was developed to control the speed of a motor with parameter variations.

Another approach to adaptive control is based on the use of artificial intelligence (AI) techniques for online tuning of linear controllers. In (Ahn and Truong 2009), a set of fuzzy rules was used to tune the gains of a PID controller on-line. The resulting controller was used in force control of an electro-hydraulic system. In (Le et al. 2013), a neural network was used to obtain the gains of a controller on-line for the control of a parallel robot manipulator of two degrees of freedom. The AI approach have shown to be effective in handling parametric uncertainties; nevertheless, a disadvantage of the used AI techniques is the required system information from the earlier to be adjusted and trained off-line, in other words, they need a collection of information previously acquired, about inputs and outputs of the system to be controlled, under different operation scenarios.

With the goal of finding the controller parameters that improve the accuracy of the controlled system operation, researches have been chosen a different control approach based on the formulation of an optimization problem, which requires optimization techniques to be solved (Dasgupta and McGregor 1992; Mori and Kita 2000). There are many computational methods to solve optimization problems, among them there are the bio-inspired meta-heuristic techniques which base their operation in natural processes. This kind of techniques has taken on great importance in the scientific community, because of their ability to find appropriate solutions with reasonable computation cost in highly non-linear optimization problems (Reeves and Sons 1993), which are commonly presented in the real world problems (Xu et al. 2016). Another important feature of these techniques, is that several mechanisms can be added in order to handle constraints presented in almost any real world problem (Mezura-Montes and Coello 2011). An application of this kind of techniques in control area is shown in (Fister et al. 2016), where the parameters of a PID controller are tuned off-line to control a SCARA robot by using different meta-heuristic techniques. After a comparative analysis among techniques, Particle Swarm Optimization (PSO) proved to be the best alternative to solve this particular problem. In (Villarreal-Cervantes and Alvarez-Gallegos 2016), several variants of the Differential Evolution algorithm were used to obtain the optimal gains of a PID controller off-line. The tuned PID parameters were used in the control of a experimental parallel robot prototype in order to validate the proposed tuning method. One way to obtain the optimal gains of a controller, when the model of the system is unknown, is presented in (Mishra et al. 2015). In that work, the parameters of the PI controller to control a valve were obtained off-line by using current system data which is handled by Differential Evolution. The obtained parameters were set in the real controller and they remained fixed through the execution time. Nevertheless, one of the main issues in the bio-inspired meta-heuristic techniques is that they does not guarantee the optimality condition, such that, they require a comparative analysis to find the most suitable bio-inspired meta-heuristic technique for solving a particular problem.

In adaptive control, meta-heuristic techniques are used to find appropriate controller parameters on-line. In (Lin et al. 2004), the parameters of a PID controller were adjusted on-line by using a Genetic Algorithm. The obtained controller was able to minimize the position error of a linear induction motor. The above work shows the performance of the parameter adjusting strategy based in a single meta-heuristic technique, but there are no evidence of the behavior of different commonly used or novel techniques under the optimization approach to adaptive the control parameter. When the control problem to be solved requires a quick response, one of the big problems that must be afford by the adaptive control approach based on meta-heuristic techniques, is the convergence time of the algorithms, which usually exceeds the minimum time required to update the control signal.

The estimation of the controller parameters on-line through the optimization approach and by using meta-heuristic techniques may reduce the negative effect of the parametric uncertainties. Nevertheless, a formal study which reveal the performance of such strategies and the viability in uncertain scenarios has not been carried out. Hence, in this work, the optimization approach is used for the adaptive control of a DC motor. The use of several adaptive control strategies based on different meta-heuristic techniques is proposed to estimate the control parameters on-line. The main goal of the proposed controllers is to minimize the error in the speed regulation of the motor. All techniques are adjusted and implemented pursing their real implementation feasibility, and the optimization technique parameters are obtained by using the *irace* package in order to make a fair comparative analysis. Through a statistical analysis from simulated results, the qualities and differences among adaptive control strategies in different scenarios of DC motor uncertainties are shown in order to find better alternatives for use under this approach.

The main contribution of this work is the presentation of an adaptive control strategy based on meta-heuristic optimization, which shows to be efficient in the compensation of uncertainties presented in the DC motor and feasible in terms of practical experimental setting up. Another contribution lies in the statistical study of different meta-heuristic techniques used in the control strategy, which aids to select the most representative features that promote the searching of better solutions in order to propose more efficient techniques and also reveals their applicability in the problem of speed regulation of the DC motor.

The rest of the paper is organized as follows: In Section 2, the closed-loop system related with the DC motor dynamics and control system is presented. The adaptive control strategy based on an on-line constrained dynamic optimization problem is stated in Section 3. In Section 4 the meta-heuristic optimization techniques and the new hybrid proposal are described. The comparative analysis is performed and discussed in Section 5. Finally, in Section 6 the conclusions are drawn.

2 DC Motor dynamics and control system

The dynamic model of the DC motor is described by (1) and its electro-mechanic diagram is shown in Fig. 1, with $p = [p_1, p_2, ..., p_6]^T = \left[\frac{b_0}{J_0}, \frac{k_m}{J_0}, \frac{k_e}{L_a}, \frac{R_a}{L_a}, \frac{1}{L_a}, \frac{\tau_L}{J_0}\right]^T$, $\bar{p} = [\bar{p}_1, \bar{p}_2, ..., \bar{p}_6]^T = \left[\frac{\bar{b}_0}{J_0}, \frac{\bar{k}_m}{J_0}, \frac{\bar{k}_e}{L_a}, \frac{\bar{R}_a}{L_a}, \frac{1}{L_a}, \frac{\bar{\tau}_L}{J_0}\right]^T$ the current and estimated vectors of DC motor parameters respectively, where θ_m, $\dot{\theta}_m$, $\ddot{\theta}_m$ are the angle, the angular velocity and the angular acceleration of the shaft respectively, i_a is the armature current, J_0 is the rotor moment of inertia, k_m is the torque constant, b_0 is the viscous friction constant, τ_L is the load torque, R_a is the armature resistance, L_a is the armature inductance, k_e is the electromotive force constant and V is the input voltage.

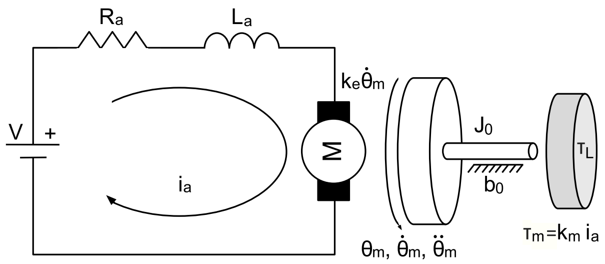

Fig. 1 Electro-mechanic diagram of the DC motor.

$$\begin{aligned} \tfrac{d\dot{\theta}_m}{dt} &= p_2 i_a - p_1 \dot{\theta}_m - p_6 \\ \tfrac{di_a}{dt} &= p_5 V - p_4 i_a - p_3 \dot{\theta}_m \end{aligned} \quad (1)$$

The dynamic model in (1) can be expressed as $\dot{x} = f(p, x(t), u(t))$, where the current state vector is proposed as $x = [x_1, x_2, x_3]^T = \left[\theta_m, \dot{\theta}_m, i_a\right]^T$ and the control signal $u = V$ is given in (2).

$$u = \frac{1}{\bar{p}_2\bar{p}_5}\left(v + \bar{p}_1\left(\bar{p}_2 x_3 - \bar{p}_1 x_2 - \bar{p}_6\right)\right) + \frac{\bar{p}_3}{\bar{p}_5}x_2 + \frac{\bar{p}_4}{\bar{p}_5}x_3 \quad (2)$$

The controller shown in (2) is used to regulate the speed of the DC motor, where $v = k_p e - k_d \dot{x}_2$, k_p and k_d are the proportional and derivative gains, $e = \omega_r - x_2$, ω_r is the desired speed and $\dot{x}_2 = p_2 x_3 - p_1 x_2 - p_6$.

3 Adaptive control strategy based on on-line optimization approach

The adaptive control strategy proposed in this work is shown in Fig. 2. The aim of this control strategy is to reduce the error in the speed regulation task of the DC motor under parametric uncertainties by obtaining the control parameters based on the motor output.

The control parameters that are obtained in this approach are those related to the estimation of the motor output.

It is important to mention that the on-line optimization method for the control tuning, requires solving a constrained dynamic optimization problem (CDOP) at each integration time $\triangle t$, in order to give the optimum parameters to the control signal.

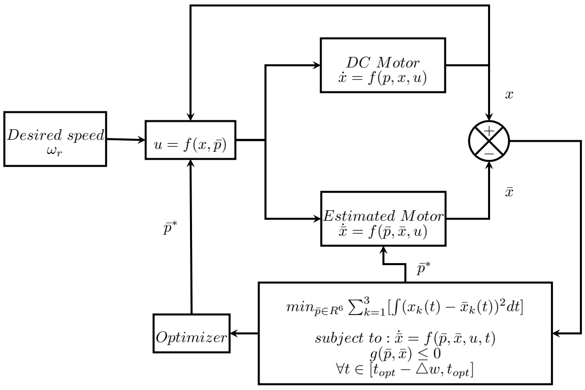

Fig. 2 Proposed control strategy.

The estimation of the motor output to find the control parameters in the CDOP requires an estimated DC motor model with an estimated state vector $\bar{x}$. Hence, the aim of the optimization problem is to find the vector of estimated parameters of the DC motor which minimizes the value of the objective function given in (3). The function in (3) equitably weights the error among each of the actual and estimated state variables in the time interval $\Omega \in [t_{opt} - \triangle w, t_{opt}]$, in which t_{opt} is the time instant when the optimization process is performed and $\triangle w$ is the time interval in which the past states of the motor and of the dynamic model are used in the error calculation.

$$\mathbf{min}_{\ \bar{p} \in R^6}\ J = \sum_{k=1}^{3} \left[\int_{t \in \Omega} \left(x_k(t) - \bar{x}_k(t)\right)^2 dt \right] \tag{3}$$

Additionally, the optimization problem is constrained by the dynamic of the current DC motor in (4) and of the estimated model in (5), by the initial conditions of the state variables in (6) and by the maximum and minimum bounds of the control signal in (7).

$$\dot{x} = f\left(p, x, u, t\right) \tag{4}$$

$$\dot{\bar{x}} = f\left(\bar{p}, \bar{x}, u, t\right) \tag{5}$$

$$\bar{x}(t_{opt} - \triangle w) = x(t_{opt} - \triangle w),\ x(0) = x_0 \tag{6}$$

$$u_{min} \leq u(t_{opt}) \leq u_{max} \tag{7}$$

4 Meta-heuristic optimizers

In this study, six different meta-heuristic techniques reported in the literature, are used to find a solution to the optimization problem stated above and are detailed next. In addition, one hybrid meta-heuristic technique is proposed to enhance the capabilities in searching reliable solutions.

4.1 Differential Evolution

Differential Evolution (DE) bases its operation in the process of natural evolution (Price et al. 2005). Algorithm 1 shows the *rand/1/bin* variant of DE. In this variant an initial population with NP individuals is generated randomly in the search space. During G_{max} generations, the individuals in the population can mutate using (8) which F is the mutation rate randomly selected per generation in range $[F_{min}, F_{max}]$. Also, three randomly selected individuals are recombined using (9) where CR is the crossover rate and $r_1 \neq r_2 \neq r_3 \neq i$ are the random indexes of the parents. For a new generation, the individuals are selected according to (10). In the last generation, the best individuals will be found in population.

Algorithm 1 Differential Evolution

$G = 0$
Generate initial population X_0 with NP individuals
Evaluate X_0
while $G < G_{max}$ **do**
    for each $x_i \in X_G$ **do**
        Generate a mutant individual v_i (see (8))
        Generate a child individual u_i (see (9))
    end for
    Select individuals for G+1 (see (10))
    $G = G + 1$
end while

$$v_i = x_{r1} + F(x_{r2} - x_{r3}) \tag{8}$$

$$u_{i,j} = \begin{cases} v_{i,j} & \textbf{if } rnd(0,1) \leq CR \\ x^G_{i,j} & \textbf{otherwise} \end{cases} \tag{9}$$

$$x_i = \begin{cases} u_i & \textbf{if } f(u_i) < f(x_i) \\ x_i & \textbf{otherwise} \end{cases} \tag{10}$$

4.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) emulates the collaborative behavior of many species in search of food (Kennedy and Eberhart 1995). Algorithm 2 shows the basic operation of PSO where NP members of a particle swarm are initially distributed in the search space. During G_{max} generations, particles modify their velocity using (12) where C_1 and C_2 are factors that weight the knowledge of the best known position by a particle and by the swarm, and ω is a factor which reduce the velocity of all particles (Shi and Eberhart 1998) obtained with (11) within the range $[V_{min}, V_{max}]$. Using its velocity, each particle modify its position using (13) and at the end of the cycle, the particles will be in the best places.

Algorithm 2 Particle Swarm Optimization

$G = 0$
Generate initial swarm X_0 with NP particles
Evaluate X_0
For each particle x_i initialize its best known position $x_i^{best} = x_i$
Initialize the best known position of the swarm x_{swarm}^{best}
Initialize the velocity of each particle $\dot{x}_i$
while $G \leq G_{max}$ **do**
  Update the velocity factor w (see (11))
  for each $x_i \in X_G$ **do**
    Update its velocity $\dot{x}_i$ (see (12))
    Update its position x_i (see (13))
    Update its best known position $x_i^{best} = x_i$
  end for
  Update the best known position of the swarm x_{swarm}^{best}
  $G = G + 1$
end while

$$\omega = V_{\max} - \frac{G}{G_{max}} \left(V_{\max} - V_{\min}\right) \quad (11)$$

$$\begin{aligned} \dot{x}_i = \omega \dot{x}_i + rnd(0,1) \cdot C_1 \left(x_i^{best} - x_i\right) \\ + rnd(0,1) \cdot C_2 \left(x_{swarm}^{best} - x_i\right) \end{aligned} \quad (12)$$

$$x_i = x_i + \dot{x}_i \quad (13)$$

4.3 Bat Algorithm

Bat Algorithm (BAT) is based on the behavior of a group of bats in search of prey by echo-location (Yang 2010). Algorithm 3 shows the behavior of BAT where a group of NP bats is distributed in the search space. During G_{max} generations, each bat uses a fixed amplitude of its echolocation signal by adjusting the frequency Q_i in the range $[f_{min}, f_{max}]$ as in (14). Q_i is used to drive the search of a bat to the best position using (15) and (16). Depending on its pulse rate r_i, each bat can make a random search around the best position using (17). Each bat can fly randomly as shown in (18) with $\bar{A}$ the average loudness of bats. If the new position of the bat improves its conditions, the bat reduces its loudness A_i and increases its pulse rate r_i using (19) and (20) with $\alpha, \gamma \in [0, 1]$ and r_0 the maximum frequency. At the end generation, the bats will be near their prey.

Algorithm 3 Bat Algorithm

$G = 0$
Generate initial population X_0 with NP bats
Evaluate X_0
Initialize the best known position of the population x_{swarm}^{best}
Initialize the velocity of each bat $\dot{x}_i$
while $G \leq G_{max}$ **do**
  for each $x_i \in X_G$ **do**
    Update its pulse frequency f_i (see (14))
    Update its velocity $\dot{x}_i$ (see (15))
    Update its position x_i (see (16))
    Fly close to x_{swarm}^{best} (see (17))
    Fly randomly (see (18))
  end for
  Update the best known position of the population x_{swarm}^{best}
  $G = G + 1$
end while

$$f_i = f_{\min} + rnd(0,1) \cdot (f_{\max} - f_{\min}) \quad (14)$$

$$\dot{x}_i = \dot{x}_i + f_i \left(x_i - x_{swarm}^{best}\right) \quad (15)$$

$$x_i = x_i + \dot{x}_i \quad (16)$$

$$x_i = \begin{cases} x_{swarm}^{best} + rnd(0,1) \cdot A_i & \textbf{if } rnd(0,1) > r_i \\ x_i & \textbf{otherwise} \end{cases} \quad (17)$$

$$x_i = \begin{cases} x_{new} = x_i + rnd(0,1) \cdot \bar{A} & \textbf{if } f(x_{new}) < f(x_i) \\ x_i & \textbf{otherwise} \end{cases} \quad (18)$$

$$A_i = \alpha A_i \quad (19)$$

$$r_i = r_0 \left(1 - \exp\left(-\gamma G\right)\right) \quad (20)$$

4.4 Firefly Algorithm

Firefly Algorithm (Yang 2009) bases its operation on the behavior of fireflies in matchmaking. Algorithm 4 shows the operation of FFA. FFA uses a population of NP fireflies distributed in the search space. During G_{max} generations, each firefly can move toward the fireflies with greater luminescence (based on the value of the objective function) using (22), (23) and (24) where r is the euclidean distance between the positions of a pair of fireflies, γ is an absorption coefficient, $\beta_{min} \in [0, 1]$ is the minimum value of β and α is a step size. The step size is reduced in each generations as shown in (21) with $w \in [0, 1]$. At the end of the algorithm, the fireflies will have the higher luminescence.

Algorithm 4 Firefly Algorithm

$G = 0$
Generate initial population X_0 with NP fireflies
while $G \leq G_{max}$ **do**
  Evaluate X_G
  Sort X_G ascending based on the luminescence of the fireflies (value of the objective function)
  for each $x_i \in X_G$ **do**
    for each $x_j \in X_G$ **do**
      Move x_i towards x_j (see (23) and (24))
    end for
  end for
  $G = G + 1$
end while

$$\alpha = w\alpha \tag{21}$$

$$\beta = (1 - \beta_{\min}) \exp\left(-\gamma r^2\right) \tag{22}$$

$$x_{new} = x_i + \beta\left(x_j - x_i\right) + \alpha\left(rnd(0, 1) - \frac{1}{2}\right) \tag{23}$$

$$x_i = \begin{cases} x_{new} \ \textbf{if} \ f(x_j) < f(x_i) \\ x_i \quad\quad \textbf{otherwise} \end{cases} \tag{24}$$

4.5 Wolf Search Algorithm

Wolf Search Algorithm (WSA) is based on the behavior of groups of wolves in search of food (Tang et al. 2012). Algorithm 5 shows the operation of WSA where a group of NP wolves are distributed in the search space. During G_{max} generations, each wolf can move within its radius of visibility using (25) and (30) where v is the range of visibility and α is the speed factor of the wolves. Each wolf looks for the best near partner within its radius of visibility according to (26), if there is any, the wolf moves toward it using (27) and (30) with r the euclidean distance. If there is no partner, the wolf moves again randomly. If a wolf feels threatened, it can escape to a new position in a wider radius using (28) and (29) where s is the step size for escape and p_a is the probability of not finding threat. At the end of the algorithm, the wolves will be near the food.

Algorithm 5 Wolf Search Algorithm

$G = 0$
Generate initial population X_0 with NP wolves
Evaluate X_0
while $G \leq G_{max}$ **do**
  for each $x_i \in X_G$ **do**
    Look for food (see (25) and (30))
    for each $x_j \in X_G$ **do**
      Look for the best near partner (see (26))
    end for
    if $x_{best} \neq x_i$ **then**
      Move toward x_{best} (see (27) and (30))
    else
      Look for food (see (25) and (30))
    end if
    Escape (see (28) and (29))
  end for
  $G = G + 1$
end while

$$x_{new} = x_i + rnd(-1, 1) \cdot \alpha v \tag{25}$$

$$x_{best} = \begin{cases} x_j \ \textbf{if} \ f(x_j) < f(x_i) \ \textbf{and} \ dist\left(x_i, x_j\right) \leq \alpha v \\ x_i \quad\quad\quad\quad\quad\quad otherwise \end{cases} \tag{26}$$

$$x_{new} = x_i + e^{-r^2}\left(x_{best} - x_i\right) + rnd(-1, 1) \cdot \alpha v \tag{27}$$

$$x_{new} = x_i + rnd(-1, 1) \cdot sv \tag{28}$$

$$x_i = \begin{cases} x_{new} \ \textbf{if} \ rnd(0, 1) < P_a \ \textbf{and} \ f(x_{new}) < f(x_i) \\ x_i \quad\quad\quad\quad\quad\quad otherwise \end{cases} \tag{29}$$

$$x_i = \begin{cases} x_{new} \ \textbf{if} \ f(x_{new}) < f(x_i) \\ x_i \quad\quad \textbf{otherwise} \end{cases} \tag{30}$$

4.6 Genetic Algorithm

Genetic Algorithms (GAs) are stochastic optimization techniques whose operation is based in the processes of natural selection and evolutionary genetics (Sivanandam and Deepa 2007). Algorithm 6 shows the working of a Genetic Algorithm (GA) in which initially, a group of NP chromosomes have different genotypic configurations coded with real numbers. Along G_{max} generations, NP pairs of parents are selected by tournament as shown in Algorithm 7, where TS is the number of contestants. Each pair of parents is combined to generate a child chromosome using a heuristic crossover method (Michalewicz et al. 1994) according with (31) and (32), where CR is the crossover rate. A child chromosome can mutate by using a non-uniform mutation operation (NUM) (Michalewicz 1995) shown in (33), (34) and (35), where F is the mutation rate and $b = 1$ is a factor that controls the non-uniformity of the mutation. At the end of the algorithm, the chromosomes will have better qualities.

Algorithm 6 Genetic Algorithm

$G = 0$
Generate initial population X_0 con NP chromosomes
Evaluate X_0
while $G < G_{max}$ **do**
    for each $x_i \in X_G$ **do**
        Select two parents x_j and x_k by tournament (see Algorithm 7))
        Generate child u_i (see (31) and (32))
        Genera mutant v_i (see (33) and (34))
    end for
    Replace parents with mutants
    $G = G + 1$
end while

Algorithm 7 Selection by tournament for GA.

function SELECTION()
    $rand = random(1, NP)$
    $x_{best} = x_{rand}$
    $t = 0$
    while $t < TS$ **do**
        $rand = rnd(1, NP)$
        if $f(x_{rand}) < f(x_{best})$ **then**
            $x_{best} = x_{rand}$
        end if
        $t = t + 1$
    end while
    return x_{best}
end function

$$u_i = \begin{cases} x_{child} \ \textbf{if} \ rnd(0,1) \leq CR \\ x_j | x_k \quad \textbf{otherwise} \end{cases} \tag{31}$$

$$x_{child} = \begin{cases} rnd(0,1) \cdot (x_j - x_k) + x_j \ \textbf{if} \ f(x_j) < f(x_k) \\ rnd(0,1) \cdot (x_k - x_j) + x_k \quad \textbf{otherwise} \end{cases} \tag{32}$$

$$v_i = \begin{cases} x_{mut} \ \text{if} \ rnd(0,1) \leq F \\ u_i \quad \textbf{otherwise} \end{cases} \tag{33}$$

$$x_{mut,j} = \begin{cases} u_{i,j} + \triangle\left(G, U_j - u_{i,j}\right) \ \textbf{if} \ rnd(0,1) < 0.5 \\ u_{i,j} - \triangle\left(G, u_{i,j} - L_j\right) \quad \textbf{otherwise} \end{cases} \tag{34}$$

$$\triangle\left(G, y\right) = y\left(1 - rnd(0,1)^{\left(1 - \frac{G}{G\max}\right)^b}\right) \tag{35}$$

4.7 Proposed hybridization

According to the statistical study presented later in this paper, a hybridization of the most promising metaheuristic techniques is proposed in order to enhance their capabilities in searching reliable solutions. The hybridization consists in including to the PSO a change in the velocity formula related to the essence of the DE. This modifications consider a different topology with a binomial crossover operation and the inclusion of a selection mechanism into the position and velocity of the particle.

The general operation of the PSO/DE hybridization is observed in Algorithm 8. In this proposal, NP particles of a swarm are initially distributed in the search space. During G_{max} generations, particles estimate the future velocity. The topology of the velocity equation is given in (36) where the main difference is the inclusion of two random selected particles or nodes into the graph denoted by indexes $r_1 \neq r_2 \neq i$. The terms C_1 and C_2 are factors that weight the knowledge of the best known position by a particle and by the swarm, C_3 weight the position of two randomly selected particles.

$$\begin{aligned} v_i = \omega \dot{x}_i + rnd(0,1) \cdot C_1\left(x_i^{best} - x_i\right) \\ + rnd(0,1) \cdot C_2\left(x_{swarm}^{best} - x_i\right) \\ + rnd(0,1) \cdot C_3\left(x_{r_1} - x_{r_2}\right) \end{aligned} \tag{36}$$

The future velocity is estimated through a binomial crossover process where exchanges some components of the velocity equation between (36) and $\omega \dot{x}_i$. This process is expressed in (37), where CR is the probability of changing the current velocity and ω is a factor which reduce the velocity of all particles obtained with (11) within the range $[V_{min}, V_{max}]$.

$$\dot{x}_{i,j}^{next} = \begin{cases} \omega v_{i,j} & \textbf{if } rnd(0,1) \leq CR \\ \omega \dot{x}_{i,j} & \textbf{otherwise} \end{cases} \tag{37}$$

The future position is estimated according to (38).

$$x_i^{next} = x_i + \dot{x}_i^{next} \tag{38}$$

After estimating its velocity and position, the elitist selection mechanism decides if the updated velocity and position of the $i-th$ particle ($\dot{x}_i^{next}$ and x_i^{next}) or the previous calculated one ($\dot{x}_i$ and x_i) pass to the next generation, i.e., it decides if it is better to move to the next states or wait in the same state for the next generation. This mechanism is represented in (39) and (40). Using the appropriate decisions, each particle will be in the best position at the end of the cycle.

$$x_i = \begin{cases} x_i^{next} & \textbf{if } f(x_i^{next}) < f(x_i) \\ x_i & \textbf{otherwise} \end{cases} \tag{39}$$

$$\dot{x}_i = \begin{cases} \dot{x}_i^{next} & \textbf{if } f(x_i^{next}) < f(x_i) \\ \dot{x}_i & \textbf{otherwise} \end{cases} \tag{40}$$

Algorithm 8 PSO/DE hybridization

$G = 0$
Generate initial swarm X_0 with NP particles
Evaluate X_0
For each particle x_i initialize its best known position $x_i^{best} = x_i$
Initialize the best known position of the swarm x_{swarm}^{best}
Initialize the velocity of each particle $\dot{x}_i$
while $G \leq G_{max}$ **do**
    Update the velocity factor w (see (11))
    for each $x_i \in X_G$ **do**
        Estimate its possible future velocity $\dot{x}_i^{next}$ (see (36) and (37))
        Estimate its possible future position x_i^{next} (see (38))
        Decide where to move (see (39) and (40))
        Update its best known position $x_i^{best} = x_i$
    end for
    Update the best known position of the swarm x_{swarm}^{best}
    $G = G + 1$
end while

4.8 Constraint handling

To handle the constraints of the optimization problem, the criterion of Deb was used in each algorithm to decide whether one solution is better than another (Deb 1998). Additionally to the criterion of Deb shown in the first three points of the following list, the fourth point is included to handle two solutions whose features do not allow to distinguish if one is better than other:

- Any feasible is preferred to any infeasible solution.
- Among two feasible solutions, the one having better objective function value is preferred.
- Among two infeasible solutions, the one having smaller constraint violation is preferred.
- Among two infeasible solutions with the same constraint violation, one of them is preferred randomly with same probability.

5 Results

5.1 Experiment design

In this work, three experiments were made based on different kind of parametric uncertainties in the DC motor. In each experiment, the efficacy of the strategies AC-DE, AC-PSO, AC-BAT, AC-FFA, AC-WSA, AC-GA and AC-PSO/DE (adaptive controllers based in DE, PSO, BAT, FFA WSA, GA and PSO/DE respectively) is proven.

The aim of each adaptive control strategy is regulating the speed of a DC motor to $\omega_r = 52.35\ rad/s$ within a time period of $t \in [0, 3]s$. The current motor has the following nominal parameters: $L_a = 102.44 \times 10^{-3}\ H$, $R_a = 9.665\ \Omega$, $k_m = 0.3946\ Nm$, $k_e = 0.4133\ v/rad$, $b_0 = 5.85 \times 10^{-4}\ Nms$ and $J_0 = 3.45 \times 10^{-4}\ Nms^2$. Additionally, a discontinuous load is implemented, i.e., the use of a torque load $\tau_L = 0.05\ Nm$ when $t \in [1, 2]s$ is proposed and a torque load $\tau_L = 0\ Nm$ is used otherwise. The gains of the controller in (2) are set as $k_p = 2500$ and $k_d = 100$ and are obtained by trial and error procedure. For each adaptive strategy, a sampling time of $\triangle t = 5\ ms$ is taken and the time interval $\triangle w = 50\ ms$ is proposed. For the time interval $t \in [0, \triangle w)$, a constant control signal $u = 20\ V$ is used, whereas for $t \geq \triangle w$ the control signal in (2) is given. Furthermore, the bounds of the control signal are set as $-50\ V \leq u \leq 50\ V$.

In the first experiment named as EX1, there are no parametric uncertainties in the DC motor. For the second experiment EX2, the DC motor parameters vary dynamically 10% from their nominal parameters as: $L_a(t) = L_a + 0.1 L_a sin(\pi t)$, $R_a(t) = R_a + 0.1 R_a sin(2\pi t/3)$, $k_m(t) = k_m + 0.1 k_m sin(2\pi t)$, $k_e(t) = k_e + 0.1 k_e sin(2\pi t)$, $b_0(t) = b_0 + 0.1 b_0 sin(\pi t)$, $J_0(t) = J_0 + 0.1 J_0 sin(2\pi t/3)$. In the last experiment EX3, the DC motor parameters vary dynamically as in EX2 and in addition a random noise signal is added to the vector of states for each sampling instant $\triangle t$ in order to simulate the noise in the sensor signals. The noise signal has the following maximum values: ± 0.01, ± 0.1 and ± 0.001 for the angular position x_1, angular velocity x_2 and current x_3 respectively.

In order to provide a fair comparison among the different adaptive control strategies, the parameter tuning for each algorithm was made by the *irace* package of the statistical analysis software *R* (López-Ibáñez et al. 2016), using the conditions of the first experiment. Table 1 shows the obtained parameters for each algorithm. The stop criterium is the number of evaluation of the objective function. 1500 evaluations is set as the stop criterium in the adaptive control strategy based on meta-heuristic algorithms. The lower and upper bounds of the vector of design variables $\bar{p}$ are shown in Table 2.

All experiments were performed on a PC with a 3.2 GHz *i5-6500* processor. The simulations were developed using *C++* programming language and were compiled using *Microsoft Visual C++*.

5.2 Discussion

This section is divided into three discussion cases named as the Case A, the Case B and the Case C. In the Case A, the comparisons of the adaptive control strategies based on the six different meta-heuristic techniques (AC-DE, AC-PSO, AC-BAT, AC-FFA, AC-WSA and AC-GA) are detailed in order to know the most reliable strategies to solve the speed regulation problem of the DC motor under parametric uncertainties. Those comparisons are related with meta-heuristic techniques reported in the literature. Then, in the Case B, the proposed AC-PSO/DE is discussed and compared to the six meta-heuristic techniques reported in the literature. For the Case C, comparisons with a robust control approach are given.

5.2.1 Case A

For each experiment 100 different executions were performed. Tables 3, 4 and 5 contain the results obtained from the execution of each experiment using the different adaptive control strategies. In these tables, the standard deviation of the error velocity $std(\|e\|)$, the magnitude average for the velocity error $\overline{\|e\|}$ and for the control signal $\overline{\|u\|}$ in the time interval $t \in [\triangle w, 3]s$ from the 100 executions are shown. The best ($\|e\|_{best}$, $\|u\|_{best}$) and the worst ($\|e\|_{worst}$, $\|u\|_{worst}$) execution with respect to the magnitude of the velocity error and the control signal are also included. In addition, the mean time to compute the simulation results in the time period $t \in [0, 3]s$ for 100 executions is displayed in $\overline{t_{exec}}$, and also the convergence time of the meta-heuristic strategies to find a new control design parameter vector at each integration time Δt is given in $\overline{t_{exec}}/n$, where n is the number of times that the meta-heuristic strategies find a new control design parameter vector in $t \in [0, 3]s$. Boldface is used to represent the best results for each column. Fig. 3 shows the motor speed behavior and the control signal applied at each time instant for the best run of each adaptive control strategies for the experiments EX1, EX2 and EX3.

It is observed in Table 3 that AC-DE is the controller that more effectively regulates the speed of the DC motor for the experiment EX1 when there are no uncertainties. For the experiment EX2 where the DC motor parameters are dynamical, Table 4 shows that AC-PSO is more effective to compensate uncertainties. In the last experiment EX3, where additionally a noise signal is included into the system, Table 5 shows that AC-PSO is also the best controller. Fig. 4 shows more clearly the behavior of the speed regulation error for the best execution of each controller. It can also be observed that almost all adaptive control strategies have a good performance in the DC motor speed regulation. When a torque load is introduced to the motor, the speed regulation error for all controllers does not surpass $\pm 5\%$ of the desired speed with exception of AC-BAT, which sometimes surpasses $\pm 10\%$. When the torque load remains fixed or is null, the speed regulation error for all controllers with exception of AC-BAT do not surpass $\pm 2\%$ of the desired speed.

In terms of the control signal average value ($\overline{\|u\|}$), it can be seen in Tables 3, 4 and 5 that all control strategies for the three experiments EX1, EX2 and EX3 have a similar energy consumption. Nevertheless, there can be noticed in $\overline{\|u\|}_{worst}$ that in some executions of the adaptive control strategies, the consumption is more than twice the control signal average value $\overline{\|u\|}$. This fact indicates that some executions of the adaptive control strategies provide violations in the maximum control signal constraint, which means that those strategies are not viable to be implemented experimentally. Based on the constraint violation of the control signal from the 100 executions, the strategies that violated such constraints in the three experiments are AC-BAT, AC-WSA and AC-GA.

To ensure that one controller works better than another, it is necessary to check the statistical validity of the results using a non-parametric test as in (Derrac et al. 2011). Tables 6, 7 and 8 show the results of the Wilcoxon test to compare all controllers by pairs. The winner between each pair of strategies is shown in boldface. Wilcoxon test was applied to the 100 $\|e\|$

Table 1 Algorithm parameters (obtained with *irace*).

Algorithm	Parameters
DE	$NP = 25$, $CR = 0.92$, $F_{min} = 0.29$, $F_{max} = 0.89$
PSO	$NP = 25$, $C_1 = 0.5$, $C_2 = 0.4$, $V_{min} = -1.0$, $V_{max} = 0.66$
BAT	$NP = 25$, $A_0 = 0.85$, $r_0 = 0.12$, $f_{min} = 0.0$, $f_{max} = 1.51$
FFA	$NP = 25$, $\alpha = 0.86$, $\beta_{min} = 0.77$, $\gamma = 0.15$
WSA	$NP = 25$, $\alpha = 0.19$, $P_a = 0.68$, $s = 4.9$, $v = 1.0$
GA	$NP = 25$, $CR = 0.2$, $F = 0.25$ $TS = 15$
PSO/DE	$NP = 25$, $C_1 = 1.1$, $C_2 = 0.62$, $C_3 = 0.99$, $V_{min} = 0.06$, $V_{max} = 0.1$, $CR = 0.39$

Table 2 Bounds of design variables.

Bound	$\bar{p}_1$	$\bar{p}_2$	$\bar{p}_3$	$\bar{p}_4$	$\bar{p}_5$	$\bar{p}_6$
Upper	2	1200	5	100	10	150
Lower	0	0	0	0	0	-150

samples of the experiments EX1, EX2 and EX3 for each controller. The statistical significance of the test was set as 5% and a *two-sided* alternative hypothesis was selected. The *two-sided* hypothesis establishes that two controllers have different distributions of $\|e\|$. Additionally, the rank sums R_+ and R_- expose the best algorithm of a pair when the statistical significance denoted by $p - value$ do not surpasses 5%. In Table 6, it can be observed that the distribution of $\|e\|$ for AC-DE controller is different to distributions of the other controllers. In addition, the rank sums show that the elements of the samples of AC-DE surpass to almost all elements of the samples of the other controllers, so AC-DE can be considered as the best controller for speed regulation when there are no parametric uncertainties (experiment EX1), followed by the AC-FFA and AC-GA controllers. For Tables 7 and 8, which contain the results of Wilcoxon test over the samples of $\|e\|$ when there are uncertainties (experiments EX2 and EX3), the AC-PSO and AC-FFA controllers showed to be the best ones (based on the rank sums). Table 9 summarizes the number of wins of each controller for every experiment based on the Wilcoxon test. The results indicate that controllers can be ordered from best to worst based on their performance as follows: AC-FFA, AC-PSO, AC-GA, AC-DE, AC-WSA and AC-BAT.

One way to observe the variations of the speed error using different adaptive control strategies is by paying attention to the rank sums of the Wilcoxon test when a controller is compared versus the worst one, which turned out to be AC-BAT (according to Table 9). Based on the information in Tables 6, 7 and 8, it is observed that in some comparisons versus AC-BAT, some strategies have a non-zero value in rank sum. This indicates that some executions of the adaptive control strategies, the error vector norm $\|e\|$ is worse than the best execution of AC-BAT (see Tables 3, 4 and 5), i.e., $\|e\| \geq 51.4368$. The above means that AC-GA, AC-WSA and AC-BAT present more variation in the speed error than the other adaptive control strategies and it can also be confirmed with the value of $std(\|e\|)$ in Tables 3, 4 and 5. The value of $std(\|e\|)$ indicates what extent the speed error remain without changing drastically, i.e., the variation with respect to $\overline{\|e\|}$. Hence, using these criteria, AC-DE, AC-FFA and AC-PSO strategies are more reliable while AC-GA, AC-WSA and AC-BAT are not reliable.

The main objective in the dynamic optimization problem into the adaptive strategies is to track the current state vector x by the estimated state vector $\bar{x}$, and using such vector $\bar{x}$ to provide the adaptive control parameters at each time interval. In Fig. 5 it is verified that the behavior of the current state vector x and the estimated state vector $\bar{x}$ for the AC-FFA controller (which resulted to be the most invariant with respect to the speed regulation error) is similar. In addition in Fig. 6 the behavior of the adaptive control parameters $\bar{p}$ are shown. This behavior is related with the operation of the meta-heuristic techniques, which search into a wide space for solutions that provide the major benefits for a particular problem.

On the other hand, it is an important fact to know whether the adaptive control strategies can be implemented in an embebed system. Based on the column $\overline{t_{exec}/n}$ in Table 3, 4 and 5, only AC-FFA cannot be used in the experimental implementation due to the convergence time of the algorithm excedes the sampling time $\Delta t = 5ms$.

Given the statistical evidence in this work and the feasible to be experimentally implemented without violate constraints and with a computational time less than the sampling time, the AC-PSO and AC-DE are the most viable strategies to solve the speed regulation problem of the DC motor subject to parametric uncertainties. These strategies are the most effective, trustworthy and feasible for experimentally implementation, because of their capacity to regulate the motor speed with a reasonable convergence time of their algorithms.

5.2.2 Case B

The adaptive control strategy based on this PSO and DE hybridization (AC-PSO/DE) was tested over 100 independent runs under the same experiment conditions in EX1, EX2 and EX3.

The results of using AC-PSO/DE are shown in Table 10. The boldface results in Table 10 indicate the outstanding results with respect to the ones obtained from the other adaptive strategies presented in Tables 3, 4 and 5. In EX1, the AC-PSO/DE alternative has an improvement in performance with respect to AC-PSO. The AC-PSO/DE performance in EX1 is also near to the one of AC-DE which turned out to be the best alternative when there are no parametric uncertainties. For EX2 and EX3, the AC-PSO/DE alternative significantly overcomes the others according to the $\overline{\|e\|}$ value and its behavior is more reliable if taking into account the value of $std(\|e\|)$.

The behavior of the best run of AC-PSO/DE can be observed in Fig. 7 and its error in speed regulation is shown in Fig. 8. It must be noticed that the error signals in Fig. 8 are smoother than the error signals obtained by the other adaptive controllers displayed in Fig. 4.

To ensure that the proposed AC-PSO/DE alternative has the most promising behavior, the Wilcoxon test is also performed among AC-PSO/DE and the other alternatives for EX1, EX2 and EX3 again with a statistical significance of 5% and a *two-sided* alternative hypothesis. The results of the Wilcoxon test are shown in Tables 11, 12 and 13. Table 14 summarizes the overall wins of each controller including AC-PSO/DE. The results indicate that proposing AC-PSO/DE can improve the performance provided by the two best adaptive control alternatives presented in section 5.2 (AC-PSO and AC-DE) and also it is feasible to be experimentally implemented due to the convergence time of the algorithm is less than the sampling time.

5.2.3 Case C

Additionally to the behavioral study of the adaptive control strategy based on different meta-heuristic techniques, it is interesting to observe the differences of this approach when compared with others such as the robust control.

For this, a generalized proportional integral observer based robust controller (RC-GPI) that works for a widely class of non-linear systems (Sira-Ramirez et al. 2011) is implemented to perform comparisons.

Table 15 shows the results of using the RC-GPI for the experiments EX1, EX2 and EX3. The values of $\|e\|$ and $\|u\|$ are obtained in the time interval $t \in [\triangle w, 3]s$ for a fair comparison. As it can be noticed, the proposed adaptive control strategy based on different meta-heuristic techniques overcomes the performance of the GPI-RC when comparing the $\|e\|_{worst}$ values in Tables 3, 4, 5 and 10 with the $\|e\|$ values of the GPI-RC in Table 15.

Fig. 9 shows the behavior of the RC-GPI and Fig. 10 shows at close range of the speed regulation error. In Figs. 9 and 10, the convergence of the motor speed to the reference signal is slower than the proposed adaptive control strategy. It also can be noticed in Figs. 9 and 10 that the RC-GPI present some difficulties when there are some noise in the motor states (EX3) and then the error has some oscillations proportional to the magnitude of the noise unlike the proposal.

6 Conclusion and future work

The study of different optimization meta-heuristic techniques in the adaptive control, shows the qualities of each technique in solving the problem of on-line parameter estimation of a DC motor subject to parametric uncertainties. Among the analyzed qualities are the accuracy in speed regulation, the energy consumption, the invariability with respect to error and the computational time required for each technique. The simulation results show that AC-PSO and AC-DE are the most promising adaptive strategies.

Based on the obtained statistical results, a hybridization of the most promising meta-heuristic techniques is proposed and used in the adaptive control strategy. This alternative named as AC-PSO/DE has a significant performance improvement with respect to the most promising adaptive strategies (AC-PSO and AC-DE) in the speed regulation of the DC motor.

The parameter setting of each meta-heuristic technique is a crucial task whether it is intended to get appropriate solutions of an on-line optimization problem. When tuning algorithms for control tasks, it must attempted to maintain a balance between their search ability and their convergence time, so that they can obtain good solutions that fits the time constraint of the control system. If these parameters are not set correctly, the result could be an unpredictable behavior of the algorithm, with slow or premature convergence, which ultimately results in an untrustworthy behavior of the adaptive control strategy. For the above reason, the parameter tuning of each algorithm was carried out using iterative methods provided by the *irace* package of *R* software.

With the information obtained in the present work, it was observed a good estimation of the current state vectors, even with differences between the current motor parameters and the estimated ones. While it is true that the current motor parameter values in the control system minimize the error, there are also different control parameter settings that provide minimum values of the objective function. This way of conceiving the optimization problem has the advantage of not only compensate the uncertainties in the DC motor parameters, but also uncertainties caused by external agents such as noise signals.

Additionally, some comparisons with a robust control approach are performed and reveals that the proposed adaptive control strategy based in different metaheuristic techniques presents some advantages in the velocity regulation problem under dynamic and discontinuous uncertainties.

As a future work, the adaptive control strategy based on a multi-objective problem will be considered in order to manage the trade-off between the error and the energy consumption.

Acknowledgements The authors acknowledge the support of the Secretaría de Investigación y Posgrado (SIP) under the projects 20170783 and 20161030, and of the Consejo Nacional de Ciencia y Tecnología (CONACyT) under the project 281728.

Compliance with Ethical Standards

Funding: This study was funded by SIP (grant numbers 20170783 and 20161030) and by CONACyT (grant number 281728).
Conflict of interest: The authors declare that they have no conflict of interest.
Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

Ahn K, Truong D (2009) Online tuning fuzzy PID controller using robust extended kalman filter. Journal of Process Control 19(6):1011 – 1023

Bitar Z, Sandouk A, Jabi SA (2015) Testing the performances of dc series motor used in electric car. Energy Procedia 74:148 – 159

Dasgupta D, McGregor D (1992) Non-stationary function optimization using the structured genetic algorithm. In: In Proceedings of Parallel Problem Solving from Nature II, PPSN-2, pp 145–154

Deb K (1998) An efficient constraint handling method for genetic algorithms. In: Computer Methods in Applied Mechanics and Engineering, pp 311–338

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1(1):3 – 18

Fister D, Jr IF, Fister I, Šafarič R (2016) Parameter tuning of PID controller with reactive nature-inspired algorithms. Robotics and Autonomous Systems 84:64 – 75

Hashem Zadeh SM, Khorashadizadeh S, Fateh MM, Hadadzarif M (2016) Optimal sliding mode control of a robot manipulator under uncertainty using pso. Nonlinear Dynamics 84(4):2227–2239

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks (1995), p 1942–1948

Kim KH, Baik IC, Chung SK, Youn MJ (1997) Robust speed control of brushless dc motor using adaptive input-output linearisation technique. IEE Proceedings - Electric Power Applications 144(6):469–475

Kwan C, Lewis F, Yeung K (1996) Adaptive control of induction motors without flux measurements. Automatica 32(6):903 – 908

Landau I, Lozano R, M'Saad M (2011) Adaptive Control: Algorithms, Analysis and Applications. Springer Science+Business Media, New York, NY, USA

Le TD, Kang H, Suh Y, Ro Y (2013) An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator. Neurocomputing 116:53 – 61

Li Y, Tong S, Li T (2013) Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven DC motor via backstepping. Nonlinear Analysis: Real World Applications 14(1):483 – 494

Lin F, Shieh H, Shyu K, Huang P (2004) On-line gain-tuning IP controller using real-coded genetic algorithm. Electric Power Systems Research 72(2):157 – 169

Linares-Flores J, Barahona-Avalos JL, Sira-Ramirez H, Contreras-Ordaz MA (2012) Robust passivity-based control of a buck-boost-converter/dc-motor system: An active disturbance rejection approach. IEEE Transactions on Industry Applications 48(6):2362–2371

Liu K, Yao Y (2016) Robust Control: Theory and Applications. Wiley

López-Ibáñez M, Dubois-Lacoste J, Cáceres LP, Birattari M, Stützle T (2016) The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives 3:43 – 58

Mao J, Tachikawa H, Shimokohbe A (2003) Precision positioning of a dc-motor-driven aerostatic slide system. Precision Engineering 27(1):32 – 41

Mezura-Montes E, Coello CAC (2011) Constraint-handling in nature-inspired numerical optimization: Past, present and future. Swarm and Evolutionary Computation 1(4):173 – 194

Michalewicz Z (1995) Genetic algorithms, numerical optimization and constraints. In: Proceedings of the Sixth International Conference on Genetic Algorithms, pp 151–158

Michalewicz Z, Logan T, Swaminathan S (1994) Evolutionary operators for continuous convex parameter space. In: Proceedings of Third Annual Conference on Evolutionary Programming, p 84–97

Mishra P, Kumar V, Rana K (2015) An online tuned novel nonlinear PI controller for stiction compensation in pneumatic control valves. ISA Transactions 58:434 – 445

Mori N, Kita H (2000) Genetic algorithms for adaptation to dynamic environments - a survey. In: Industrial Electronics Society, 2000. IECON 2000. 26th Annual Confjerence of the IEEE, vol 4, pp 2947–2952

Orozco RG, Jimenez EI, Jimenez-Lizarraga M (2012) Comparative study of the speed robust control of a dc motor.

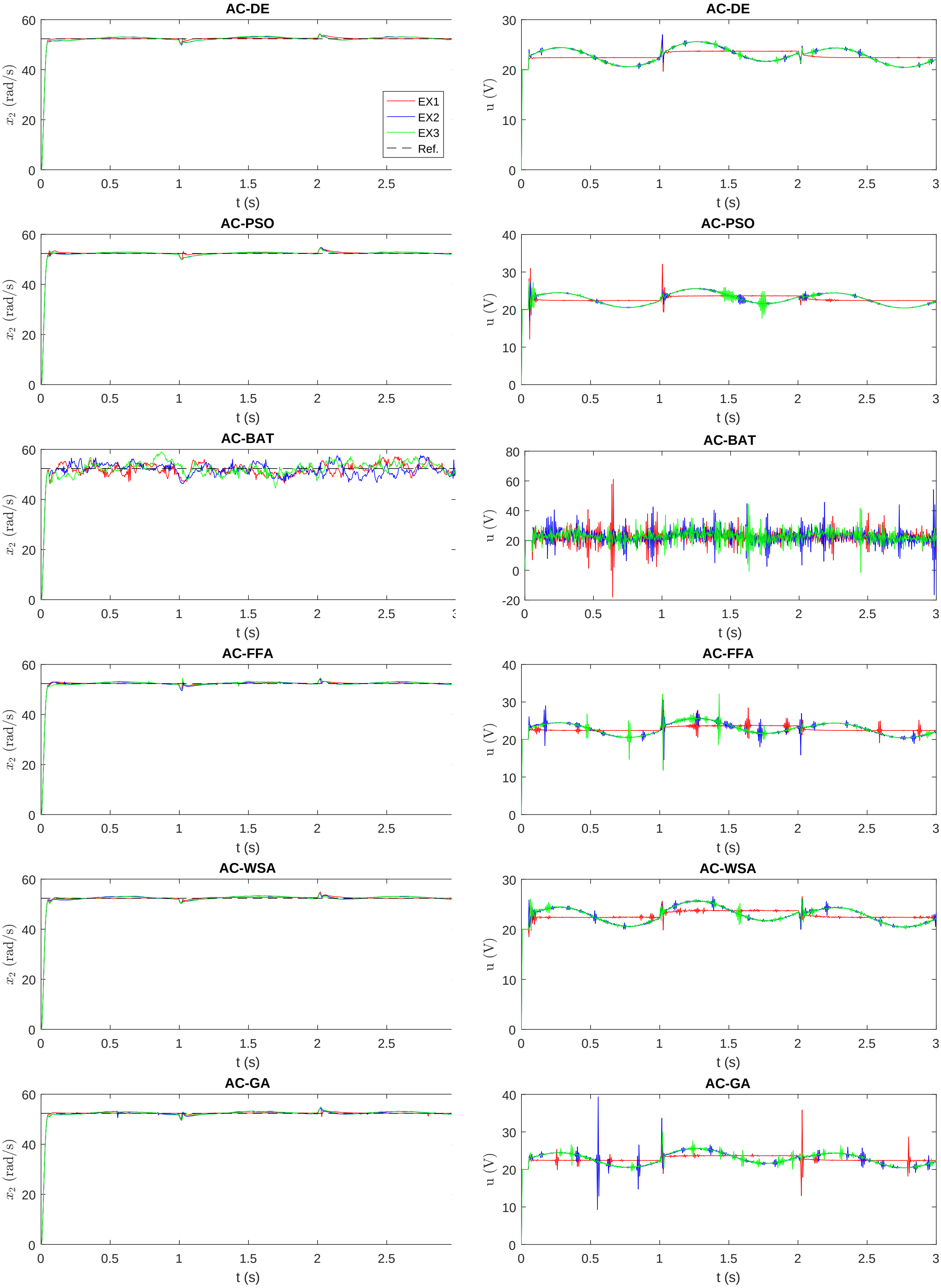

Fig. 3 Performance of adaptive controllers based on different meta-heuristic techniques in the speed regulation for the DC motor in experiments EX1, EX2 and EX3.

Table 3 Results of the regulation problem considering the experiment EX1 i.e., static DC motor parameters.

Controller	$\|e\|_{best}$	$\|e\|_{worst}$	$\overline{\|e\|}$	$std(\|e\|)$	$\|u\|_{best}$	$\|u\|_{worst}$	$\overline{\|u\|}$	$\overline{t_{exec}}(s)$	$\overline{t_{exec}}/n(s)$
AC-DE	7.8592	**13.3111**	**9.7884**	**0.9069**	**555.7000**	**556.0960**	**555.6708**	1.3204	0.0022
AC-PSO	9.3721	22.4361	14.7825	3.0443	556.0610	557.8100	556.4355	**1.1589**	**0.0020**
AC-BAT	51.4368	194.6410	74.9646	15.7524	568.1670	1138.1700	581.1571	2.1191	0.0036
AC-FFA	7.7906	18.7228	11.6178	2.3380	555.7100	556.9890	555.8016	4.0855	0.0069
AC-WSA	9.2605	253.5490	19.0906	28.5732	555.8710	1399.5600	578.4469	2.6772	0.0045
AC-GA	**7.5284**	194.6360	16.3826	21.7629	555.7310	1076.1900	565.2337	1.6688	0.0028

Table 4 Results of the regulation problem considering the experiment EX2 i.e., dynamic DC motor parameters.

Controller	$\|e\|_{best}$	$\|e\|_{worst}$	$\overline{\|e\|}$	$std(\|e\|)$	$\|u\|_{best}$	$\|u\|_{worst}$	$\overline{\|u\|}$	$\overline{t_{exec}}(s)$	$\overline{t_{exec}}/n(s)$
AC-DE	14.2482	21.3301	16.8103	**1.3608**	**556.8310**	**556.2540**	**556.9652**	1.3487	0.0023
AC-PSO	12.5794	21.2095	**14.8619**	1.8995	556.9260	559.2760	557.8749	**1.1040**	**0.0019**
AC-BAT	55.4770	125.9430	78.7608	13.8810	569.6160	831.3740	577.3526	2.2710	0.0039
AC-FFA	**12.1161**	**19.5922**	15.1051	1.5246	557.5290	556.9910	557.1327	3.9670	0.0067
AC-WSA	14.2502	1506.0500	36.8529	152.9521	557.8190	8101.6000	653.8842	2.6936	0.0046
AC-GA	13.0528	92.5447	17.9173	9.4084	557.5710	693.0750	560.1773	1.6606	0.0028

Table 5 Results of the regulation problem considering the experiment EX3 i.e., dynamic DC motor parameters and noise.

Controller	$\|e\|_{best}$	$\|e\|_{worst}$	$\overline{\|e\|}$	$std(\|e\|)$	$\|u\|_{best}$	$\|u\|_{worst}$	$\overline{\|u\|}$	$\overline{t_{exec}}(s)$	$\overline{t_{exec}}/n(s)$
AC-DE	14.8856	**21.3003**	16.7840	**1.1851**	**556.9250**	**555.9230**	**556.9366**	1.3131	0.0022
AC-PSO	**11.8634**	23.7810	**14.9437**	2.0950	557.0500	560.7820	557.6454	**1.1438**	**0.0019**
AC-BAT	58.1029	127.2550	80.4205	12.5628	569.0190	661.9910	573.1904	2.2557	0.0038
AC-FFA	12.3510	22.8277	15.6198	2.1701	557.7670	556.0330	557.1785	4.0406	0.0069
AC-WSA	14.0450	710.0560	25.6302	69.9167	557.4980	4866.6600	604.3889	2.8748	0.0049
AC-GA	13.5486	96.3305	18.6915	11.3825	557.7490	707.7740	560.5306	1.6776	0.0028

Table 6 Results of Wilcoxon test for EX1.

Wilcoxon test	R_-	R_+	*p-value*
AC-DE vs AC-PSO	7	5043	< 0.0001
AC-DE vs AC-BAT	0	5050	< 0.0001
AC-DE vs AC-FFA	718	4332	< 0.0001
AC-DE vs AC-WSA	28	5022	< 0.0001
AC-DE vs AC-GA	596	4454	< 0.0001
AC-PSO vs AC-BAT	0	5050	< 0.0001
AC-PSO vs **AC-FFA**	4426	624	< 0.0001
AC-PSO vs AC-WSA	2823	2227	0.1332
AC-PSO vs **AC-GA**	3721	1329	< 0.0001
AC-BAT vs **AC-FFA**	5050	0	< 0.0001
AC-BAT vs **AC-WSA**	4868	182	< 0.0001
AC-BAT vs **AC-GA**	4949	101	< 0.0001
AC-FFA vs AC-WSA	854	4196	< 0.0001
AC-FFA vs AC-GA	1999	3051	0.1933
AC-WSA vs **AC-GA**	3387	1663	0.0002

Table 7 Results of Wilcoxon test for EX2.

Wilcoxon test	R_-	R_+	*p-value*
AC-DE vs **AC-PSO**	4434	616	< 0.0001
AC-DE vs AC-BAT	0	5050	< 0.0001
AC-DE vs **AC-FFA**	4511	539	< 0.0001
AC-DE vs AC-WSA	1758	3292	0.1332
AC-DE vs **AC-GA**	2977	2073	0.0120
AC-PSO vs AC-BAT	0	5050	< 0.0001
AC-PSO vs AC-FFA	2046	3004	0.0352
AC-PSO vs AC-WSA	459	4591	< 0.0001
AC-PSO vs AC-GA	1087	3963	< 0.0001
AC-BAT vs **AC-FFA**	5050	0	< 0.0001
AC-BAT vs **AC-WSA**	4851	199	< 0.0001
AC-BAT vs **AC-GA**	5049	1	< 0.0001
AC-FFA vs AC-WSA	412	4638	< 0.0001
AC-FFA vs AC-GA	1309	3741	0.0004
AC-WSA vs **AC-GA**	3390	1660	0.0009

In: World Automation Congress 2012, pp 1–6

Price K, Storn RM, Lampinen JA (2005) Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series). Springer-Verlag New York, Inc., Secaucus, NJ, USA

Raslavičius L, Keršys A, Makaras R (2017) Management of hybrid powertrain dynamics and energy consumption for 2wd, 4wd, and HMMWV vehicles. Renewable and Sustainable Energy Reviews 68, Part 1:380 – 396

Reeves CB, Sons (1993) Modern Heuristic Techniques for Combinatorial Problems, vol 1, 1st edn. Wiley

Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, pp 69–73

Sira-Ramirez H, Luviano-Juárez A, Cortés-Romero J (2011) Control lineal robusto de sistemas no lineales diferencialmente planos. Revista Iberoamericana de Automática e Informática Industrial RIAI 8(1):14 – 28

Sivanandam SN, Deepa SN (2007) Introduction to Genetic Algorithms, 1st edn. Springer Publishing Company, Incorporated

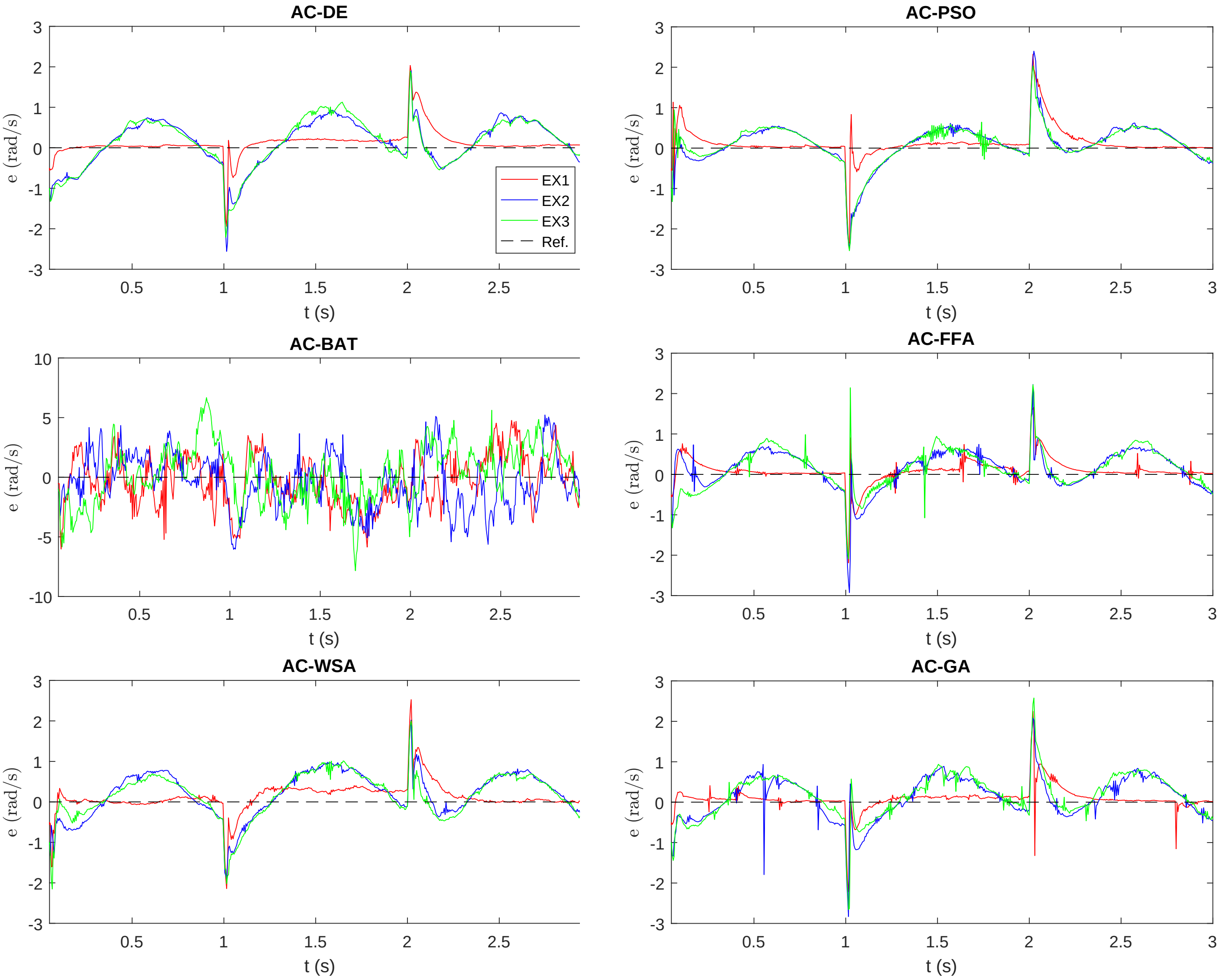

Fig. 4 Error behavior of the adaptive controllers based on different meta-heuristic techniques in the speed regulation of DC motor, when $t \geq \triangle w$ for experiments EX1, EX2 and EX3. AC-DE, AC-FFA and AC-PSO show to be the best controllers for experiments EX1, EX2 and EX3 respectively

Table 8 Results of Wilcoxon test for EX3.

Wilcoxon test	R_-	R_+	*p-value*
AC-DE vs **AC-PSO**	4381	669	< 0.0001
AC-DE vs AC-BAT	0	5050	< 0.0001
AC-DE vs **AC-FFA**	3842	1208	< 0.0001
AC-DE vs AC-WSA	1635	3415	0.0120
AC-DE vs **AC-GA**	2897	2153	0.0210
AC-PSO vs AC-BAT	0	5050	< 0.0001
AC-PSO vs AC-FFA	1763	3287	0.0569
AC-PSO vs AC-WSA	479	4571	< 0.0001
AC-PSO vs AC-GA	923	4127	< 0.0001
AC-BAT vs **AC-FFA**	5050	0	< 0.0001
AC-BAT vs **AC-WSA**	4949	101	< 0.0001
AC-BAT vs **AC-GA**	5045	5	< 0.0001
AC-FFA vs AC-WSA	760	4290	< 0.0001
AC-FFA vs AC-GA	1579	3471	0.0066
AC-WSA vs **AC-GA**	3380	1670	0.0009

Slotine JE, Li W (1991) Applied Nonlinear Control, vol 1, 1st edn. Prentice-Hall

Song Q, Jia C (2016) Robust speed controller design for permanent magnet synchronous motor drives based on sliding mode control. Energy Procedia 88:867 – 873

Tang R, Fong S, Yang XS, Deb S (2012) Wolf search algorithm with ephemeral memory. In: Digital Information Management (ICDIM), 2012 Seventh International Conference on, pp 165–172

Villarreal-Cervantes MG, Alvarez-Gallegos J (2016) Off-line PID control tuning for a planar parallel robot using DE variants. Expert Systems with Applications 64:444 – 454

Xu X, Liu J, Li H, Jiang M (2016) Capacity-oriented passenger flow control under uncertain demand: Algorithm development and real-world case study. Transportation Research Part E: Logistics and Transportation Review 87:130 – 148

Yang SF, Chou JH (2009) A mechatronic positioning system actuated using a micro dc-motor-driven propeller–thruster. Mechatronics 19(6):912 – 926

Table 9 Wins of each controller according to Wilcoxon test.

Experiment	AC-DE	AC-PSO	AC-BAT	AC-FFA	AC-WSA	AC-GA
EX1	**5**	1	0	3	1	3
EX2	1	**5**	0	4	1	3
EX3	2	**4**	0	**4**	1	3
Total	8	10	0	**11**	3	9

Table 10 Results of the regulation problem by using the AC-PSO/DE alternative.

Experiment	$\|e\|_{best}$	$\|e\|_{worst}$	$\overline{\|e\|}$	$std(\|e\|)$	$\|u\|_{best}$	$\|u\|_{worst}$	$\overline{\|u\|}$	$\overline{t_{exec}}(s)$	$\overline{t_{exec}}/n(s)$
EX1	9.5292	16.5754	12.0754	1.3076	**555.3320**	556.8170	555.9418	1.5613	0.0027
EX2	**11.9610**	21.4727	**14.1146**	**1.2366**	557.6340	562.3800	557.2886	1.5338	0.0026
EX3	**11.3433**	**19.0645**	**13.8674**	**0.8415**	558.1310	556.9860	557.2309	1.6255	0.0028

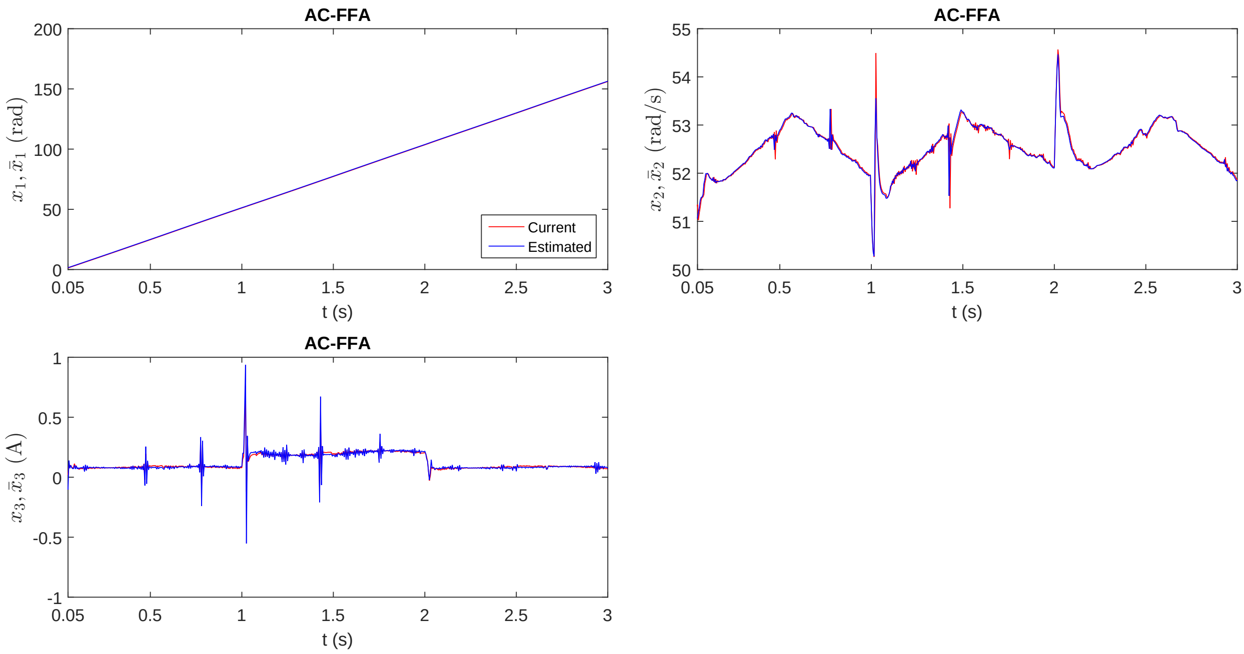

Fig. 5 Behavior of estimated state vector $\bar{x}$ of the DC motor for AC-FFA controller, when $t \geq \triangle w$ for experiment EX3.

Table 11 Results of Wilcoxon test for EX1 including the AC-PSO/DE alternative.

Wilcoxon test	R_-	R_+	*p-value*
AC-DE vs AC-PSO/DE	132.5	4917.5	< 0.0001
AC-PSO vs **AC-PSO/DE**	4392	658	< 0.0001
AC-BAT vs **AC-PSO/DE**	5050	0	< 0.0001
AC-FFA vs AC-PSO/DE	1919	3131	0.0209
AC-WSA vs **AC-PSO/DE**	3966	1084	0.0008
AC-GA vs AC-PSO/DE	2526	2524	0.6173

Table 12 Results of Wilcoxon test for EX2 including the AC-PSO/DE alternative.

Wilcoxon test	R_-	R_+	*p-value*
AC-DE vs **AC-PSO/DE**	4957	93	< 0.0001
AC-PSO vs AC-PSO/DE	3349	1701	0.0569
AC-BAT vs **AC-PSO/DE**	5050	0	< 0.0001
AC-FFA vs **AC-PSO/DE**	3965	1355	< 0.0001
AC-WSA vs **AC-PSO/DE**	5014	36	< 0.0001
AC-GA vs **AC-PSO/DE**	4694	356	< 0.0001

Yang XS (2009) Firefly Algorithms for Multimodal Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 169–178
Yang XS (2010) A New Metaheuristic Bat-Inspired Algorithm, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 65–74
Yavuz H, Stallard TJ, McCabe AP, Aggidis GA (2012) Determination of optimal parameters for a hydraulic power take-off unit of a wave energy converter in regular waves. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy 226:98–111

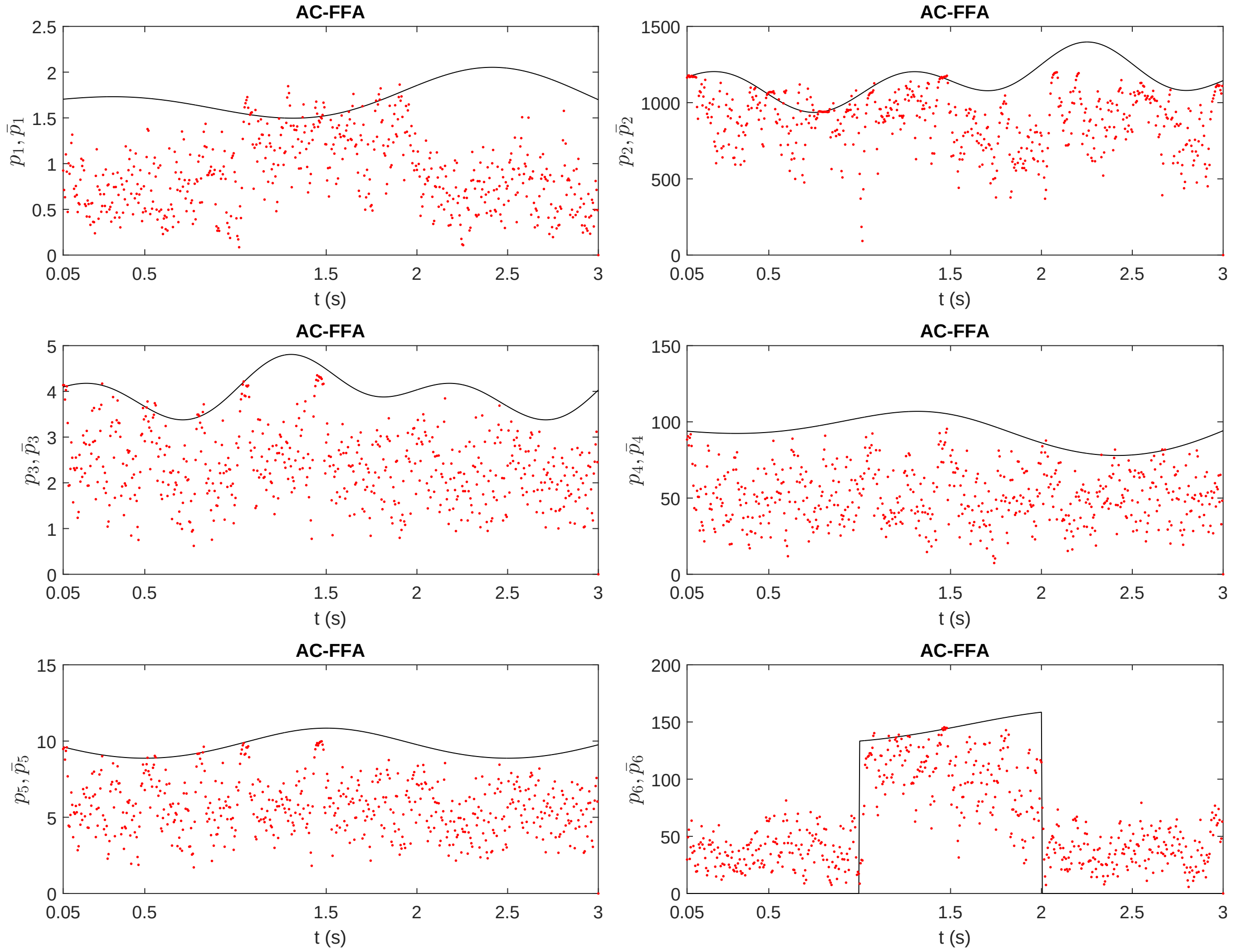

Fig. 6 Behavior of estimated parameters $\bar{p}$ of the DC motor for AC-FFA controller, when $t \geq \triangle w$ for experiment EX3. The solid line indicates the behavior of p and dots are the values of $\bar{p}$ for each instant $\triangle t$.

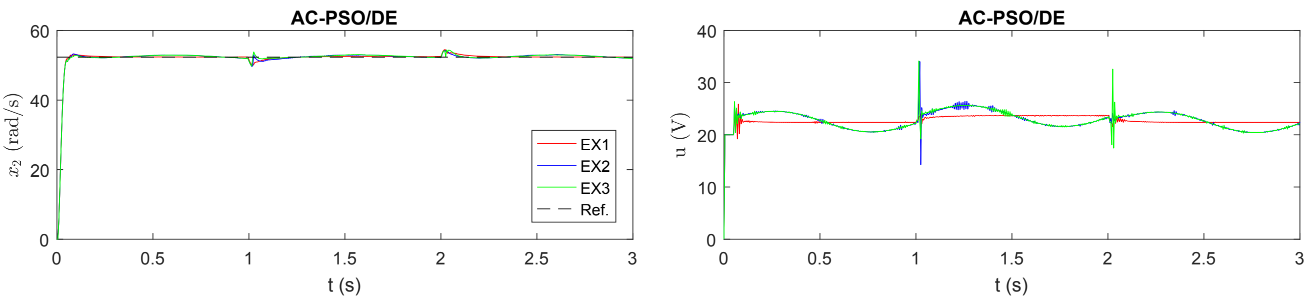

Fig. 7 Performance of adaptive controllers based on PSO/DE hybridization in the speed regulation for the DC motor in experiments EX1, EX2 and EX3.

Table 13 Results of Wilcoxon test for EX3 including the AC-PSO/DE alternative.

Wilcoxon test	R_-	R_+	*p-value*
AC-DE vs **AC-PSO/DE**	5049	1	< 0.0001
AC-PSO vs **AC-PSO/DE**	3769	1281	0.0017
AC-BAT vs **AC-PSO/DE**	5050	0	< 0.0001
AC-FFA vs **AC-PSO/DE**	4442	608	< 0.0001
AC-WSA vs **AC-PSO/DE**	4997	53	< 0.0001
AC-GA vs **AC-PSO/DE**	4871	179	< 0.0001

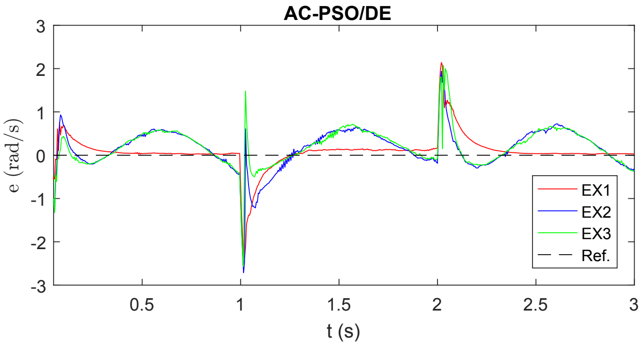

Fig. 8 Error behavior of the adaptive controller based on PSO/DE hybridization in the speed regulation of DC motor, when $t \geq \triangle w$ for experiments EX1, EX2 and EX3.

Table 14 Wins of each controller according to Wilcoxon test including the AC-PSO/DE alternative.

Experiment	AC-DE	AC-PSO	AC-BAT	AC-FFA	AC-WSA	AC-GA	AC-PSO/DE
EX1	**6**	1	0	4	1	3	3
EX2	1	5	0	4	1	3	**5**
EX3	2	4	0	4	1	3	**6**
Total	9	10	0	12	3	9	**14**

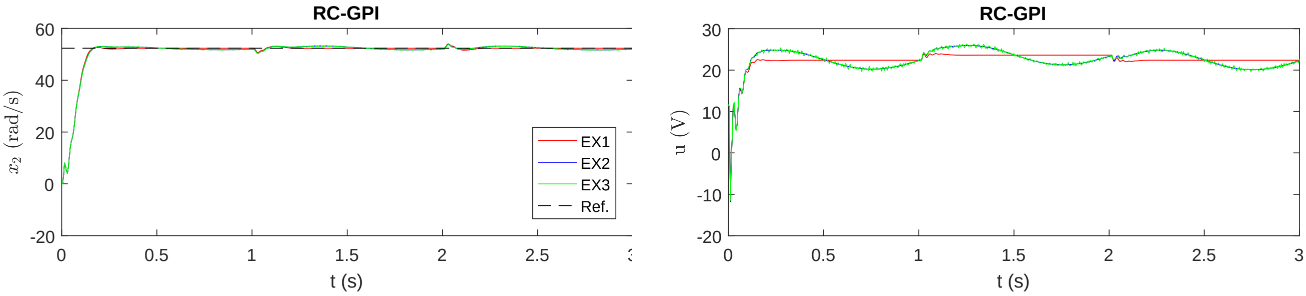

Fig. 9 Performance of the RC-GPI in the speed regulation for the DC motor in experiments EX1, EX2 and EX3 with a zoom (right figure).

Table 15 Results of the regulation problem by using the RC-GPI approach.

Experiment	$\|e\|$	$\|u\|$
EX1	87.3531	551.2878
EX2	89.4196	552.7130
EX3	89.4904	553.5853

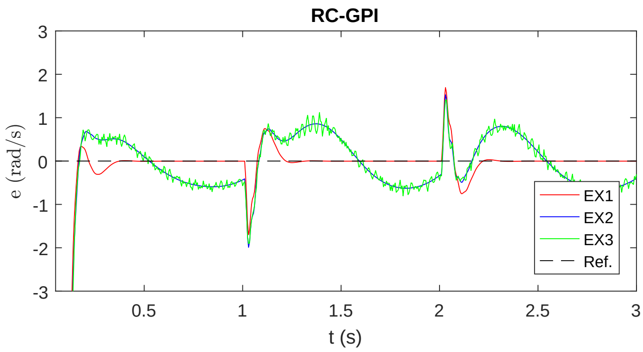

Fig. 10 Error behavior of the RC-GPI in the speed regulation of DC motor, when $t \geq \triangle w$ for experiments EX1, EX2 and EX3.