Adaptive control of a DC motor based on swarm intelligence

Consuelo V. García-Mendoza
Instituto Politécnico Nacional
ESCOM
D.F., México
Email:consuelo.varinia@gmail.com

Miguel G. Villarreal-Cervantes
and Gabriel Sepúlveda-Cervantes
Instituto Politécnico Nacional
CIDETEC
D.F., México
Email:{mvillarrealc,gsepulvedac}@ipn.mx

Ollin Peñaloza-Mejía
Instituto Tecnológico de Sonora
Sonora, México
Email:ollin.penaloza@itson.edu.mx

***Abstract*—The effect of uncertainties in the stabilization of the velocity control of a DC motor is compensated by proposing an adaptive control based on Particle Swarm Optimization (ACPSO). The ACPSO is based on an on-line dynamic optimization problem with dynamic constraints. The empirical analysis based on modifying the inertia weight of the ACPSO indicates that it can effectively regulate the motor's velocity and hence the ACPSO can be another alternative to control the DC motor under parametric uncertainties. Simulation results verify the proposed approach.**

***Keywords* - Particle swarm optimization; adaptive control; DC motor; heuristic algorithms**

I. Introduction

The control system is a crucial factor to develop a task in a mechatronic system. Frequently, cascade control systems are used for mechatronic systems due to this achieve fast disturbance rejection. One of the basic cascade control system has two control loops. The outer control loop determines the set point for the inner control loop. When the actuators of the mechatronic system are Direct Current (DC) motors, then the inner control loop involves the control of the DC motors. In [1] a cascade control system is implemented in a mobile robot. Bounded control is proposed for the outer control loop and the proportional integral control is chosen, as the inner control loop, for the velocity control of the DC motors of the mobile robot wheels. Hence, the control of the DC motor for the control loop of cascade control systems is an important element to be considered for the performance of mechatronic systems. The parametric uncertainties such as changes in the parameters of the DC motor, variations in the load, etc. affect the performance of the control system. In the last decades, this fact has motivated to search different control strategies to improve the behavior of the control system of a DC motor under parametric uncertainties. The adaptation of the control system has been performed by using different approaches: The control parameter adaptation is gotten from the Lyapunov analysis [2], [3], the adaptation is obtained by using neural networks for the online tune of the control parameters [4], the gradient descent method is applied to update the PID control gains [5]. In this paper the online adaptation of the control system parameters of a DC motor is done by establishing a dynamic optimization problem (DOP).

In the last years some efforts in the search of the solution in nonlinear optimization problems have been done. Because of the importance for finding the ”best” solution in a nonlinear optimization problem, several optimization techniques have been developed. The optimization techniques can be classified as gradient based algorithms and meta-heuristic based algorithms. Gradient based algorithms have the main drawbacks that have a fast convergence and can easily be trapped in local minima. In addition, only continuos optimization problem can be solved with these algorithms. On the other hand meta-heuristic algorithms present better performance with discontinuos and nonlinear spaces. Genetic algorithms (GA) are meta-heuristic ones with an acceptable convergence rate. However, the high computational complexity makes the real time implementation (laboratory testing) a difficult task. On the other hand, there are other proposals based on the meta-heuristic, they are grouped as ”Swarm Intelligence Algorithms” where the main characteristic is the collective behavior of decentralized and self-organized systems. Particle swarm optimization (PSO) is a population based optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [6], inspired by social behavior of bird flocking or fish schooling. PSO has advantages over other evolutionary techniques, such as rapid convergence and few parameters which are tunning. PSO computational complexity is related to the interaction between the particles, the number of particles and the number of iterations required to find the solution.

In the last 20 years, there have been developed several versions of PSO. Among the difference of the proposed PSO are the changes in the constriction coefficients, changes in the interaction of particles using different population topologies (static or dynamic) [7], minimization of the population size and the number of iterations to find solution [8], and hybrid

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.50

schemes [9], with the main goal of reducing the computational complexity and increasing the convergence rate of PSO.

From 2007, it has been reported more applications of PSO in the control loop of mechatronic system. For example in [10], PSO is used to calculate the coefficients of impedance that maintain dynamic stability in the mobile manipulator with flexible base, when this have contact with objects in unknown conditions. Each particle in the workspace is assumed to have these impedance coefficients and cost function is the integration of the difference between contact force and desired force. In [11] a modified PSO is proposed to solve the problem fuzzy predictive control considering the control of the Continuous Stirred Tank Reactor (CSTR). In [9] for the control of Ultrasonic Motors (USMs), an intelligent PID control method using Neuronal Network (NN) combined with type PSO is developed.

In this paper the analysis if the inertia weight in the online adaptation of the control system parameters of a DC motor based on PSO algorithm is done establishing dynamic optimization problem. Empirical analysis with simulation results validate the proposed approach under the effects of uncertainties in the parameters of the DC motor.

II. Optimization problem

The dynamic model of the DC motor [12] can be described in the state variable vector $\tilde{x} = [\tilde{x}_1, \tilde{x}_2]^T = [\dot{q}_m, \frac{k_m i_a - b_o q_m - \tau_L}{J_o}]^T$ with the input control vector $\tilde{u} = V_{in}$ as it is observed in (1)-(2), where $p = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{bmatrix}^T = \begin{bmatrix} b_0/J_0 & k_m/J_0 & k_e/L_a & R_a/L_a & 1/L_a & \tau_L/J_0 \end{bmatrix}^T \in R^6$, L_a is the armature inductance, k_m is the torque constant, k_e is the back electromotive force (back emf), R_a is the armature resistance, b_o is the viscous friction coefficient of the motor shaft bearing, J_o is the inertia torque of the motor rotor, V_{in} is the armature voltage, i_a is the armature current, τ_L is the load torque and q_m, $\dot{q}_m$, $\ddot{q}_m$ are the position, velocity and acceleration of the rotor, respectively.

$$\dot{\tilde{x}}_1 = \tilde{x}_2 \tag{1}$$

$$\dot{\tilde{x}}_2 = p_2 p_5 \tilde{u} + p_1 p_6 + \tilde{x}_1 (p_1^2 - p_2 p_3) - i_a (p_2 p_4 + p_1 p_2) \tag{2}$$

It is important to note that the armature current is obtained by solving the differential equation $\dot{i}_a = p_5 \tilde{u} - p_4 i_a - p_3 \dot{q}_m$.

An inverse dynamic control $\tilde{u}(t)$ is proposed to regulate the velocity of the DC motor and it is given in (3), where $e = w_r - \tilde{x}_1(t)$ is the error between the desired angular velocity w_r, and the current angular velocity $\tilde{x}_1(t)$, $\dot{e} = \dot{w}_r - \dot{\tilde{x}}_1(t)$ is the error between the desired angular acceleration $\dot{w}_r$ and the current angular acceleration $\dot{\tilde{x}}_1(t)$, $\ddot{w}_r$ is the rate of change of the desired angular acceleration, k_p, k_d are the control gains and $\bar{p} \in R^6$ is the estimated vector of p.

$$\tilde{u} = \frac{\ddot{w}_r + k_p e + k_d \dot{e} + \bar{p}_1 \bar{p}_2 x_3 - \bar{p}_1^2 x_2 + \bar{p}_1 \bar{p}_6}{\bar{p}_2 \bar{p}_5} + \frac{\bar{p}_3 x_2}{\bar{p}_5} + \frac{\bar{p}_4 x_3}{\bar{p}_5} \tag{3}$$

Defining the time space Ω as $\Omega = \{\lambda \in R \mid \lambda \in [t_1,\ t_n] \subseteq t,\ t_1 = t_n - \Delta w, \Delta w > \Delta t\}$ and the estimated dynamics of the DC motor as $\dot{\bar{\tilde{x}}} = \bar{f}(\bar{\tilde{x}}(t), \tilde{u}(t), \bar{p})$ then, the dynamic optimization problem consists on finding the optimum control design variable vector $\bar{p}^* = [\bar{p_1}^*, \bar{p_2}^*, \bar{p_3}^*, \bar{p_4}^*, \bar{p_5}^*, \bar{p_6}^*]^T$ that minimizes J (4) which is the error between the DC motor model and an estimated one such that $\bar{p}^*$ compensate the nonlinear effects on the parameter vector p of the DC motor, subject to the DC motor dynamics (5), the estimated DC motor dynamics (6), bounds in the control signal (7) and bounds in the design variable vector (8), where $\tilde{u}_{Max}$, $\tilde{u}_{Min}$ are the upper and lower control bounds and $\bar{p}_{Max}$, $\bar{p}_{Min}$ are the upper and lower design variable bounds. Hence, the general formulation of the dynamic optimization problem is stated as in (4)-(8).

$$\begin{aligned} \underset{\bar{p}^*}{Min}\ J = \quad & \int_{t\in\Omega} (\tilde{x}_1(t) - \bar{\tilde{x}}_1(t))^2 dt + \int_{t\in\Omega} (\tilde{x}_2(t) - \bar{\tilde{x}}_2(t))^2 dt \\ & + \int_{t\in\Omega} (i_a(t) - \bar{i}_a(t))^2 dt \end{aligned} \tag{4}$$

Subject to:

$$\frac{d\tilde{x}}{dt} = f(\tilde{x}(t), \tilde{u}(t), p), \qquad x(0) = [0,0,0]^T \tag{5}$$

$$\frac{\bar{\tilde{x}}}{dt} = f(\bar{\tilde{x}}(t), \tilde{u}(t), \bar{p})\Big|_{t\in\Omega}, \quad \bar{\tilde{x}}(t_1) = \tilde{x}(t_1) \tag{6}$$

$$\tilde{u}_{Min} \le \tilde{u}(t_n) \le \tilde{u}_{Max} \tag{7}$$

$$\bar{p}_{Min} \le \bar{p} \le \bar{p}_{Max} \tag{8}$$

In Fig. 1 the closed-loop system of the adaptive control based on PSO (ACPSO) is shown.

III. Particle swarm optimization algorithm

The algorithm PSO is inspired by the social behavior of flocks and fish schools foraging algorithm, developed in 1995 by psychologist - sociologist Jammes Kennedy and Russell Eberhart electronic engineer.

PSO consists in the initialization of the swarm of particles and the iterative process to solve the optimization problem.

A. Initialization

The algorithm PSO creates a swarm of NP particles $p0_{i,G=0} = [p0_{1,i,0}, p0_{2,i,0}, ..., p0_{j,i,0}, ..., p0_{D,i,0}, ...]^T \in \mathbb{R}^D$, $\forall\ i = 1, 2, ...NP$ by the generation $G = 0$. The particles are randomly initialized $p0_{j,i,G=0} = p_j^{min} + rand_j(0,1)(p_j^{max} - p_j^{min})$, where $rand_j$ is a random number with uniform distribution in the interval $[0, 1]$. Desing variables D are

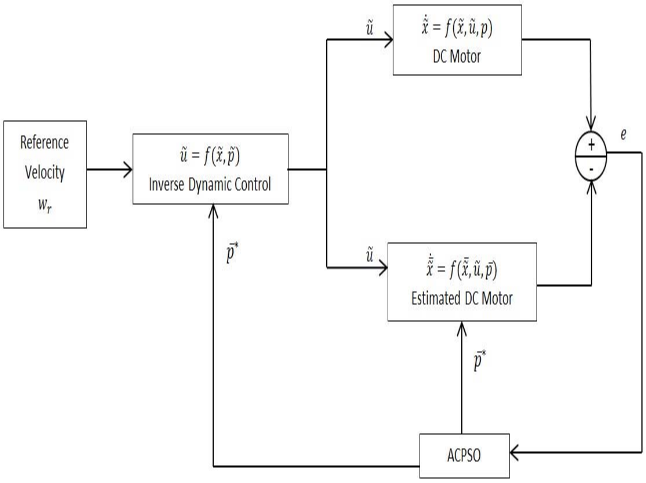


Figure 1: Schematic diagram of the adaptive control of the DC motor based on PSO.

includes in each particle, these variables are bounded by $[p_j^{min}, p_j^{max}]$ $\forall$ $j = 1, 2, ..., D$.

An auxiliary swarm $\vec{p1}_{i,G=0}$ is created and initialized like $\vec{p0}_{i,G=0}$.

The speed of the particles $v_{i,G=0} = [v_{1,i,0}, v_{2,i,0}, ..., v_{j,i,0}, ..., v_{D,i,0}, ...]^T \in \mathbb{R}^D$, $\forall$ $i = 1, 2, ...NP$ is bounded by $[v_j^{min}, v_j^{max}]$ $\forall$ $j = 1, 2, ..., D$ and randomly initialized $v_{j,i,G=0} = v_j^{min} + rand_j(0, 1)(v_j^{max} - v_j^{min})$.

The best solution found by each particle $pBest_{i,G=0} = [pBest_{1,i,0}, pBest_{2,i,0}, ..., pBest_{j,i,0}, ..., pBest_{D,i,0}, ...]^T \in \mathbb{R}^D$ is initialized $pBest_{j,i,G=0} = 1000$ $\forall$ $j = 1, 2, ...D$ and $\forall$ $i = 1, 2, ...NP$.

The best solution found in the neighborhood $gBest_{i,G=0} = [gBest_{1,i,0}, gBest_{2,i,0}, ..., gBest_{j,i,0}, ..., gBest_{D,i,0}, ...]^T \in \mathbb{R}^D$ is inicialized $gBest_{j,i,G=0} = 0$ $\forall$ $j = 1, 2, ...D$ and $\forall$ $i = 1, 2, ...NP$.

The constriction coefficients $\chi = 0.2$, $U1_{i,G=0} = [U1_{1,i,0}, U1_{2,i,0}, ..., U1_{j,i,0}, ..., U1_{D,i,0}, ...]^T$ and $U2_{i,G=0} = [U2_{1,i,0}, U2_{2,i,0}, ..., U2_{j,i,0}, ..., U2_{D,i,0}, ...]^T \in \mathbb{R}^D$ are initialized $U1_{j,i,G=0} = 0$ and $U2_{j,i,G=0} = 0$ $\forall$ $j = 1, 2, ...D$ and $\forall$ $i = 1, 2, ...NP$.

B. Iterative process

For $G = 1, 2, ..., GenMax$ is evaluated $J(\vec{p0}_{i,G})$ and $J(\vec{pBest}_{i,G})$, $\forall$ $i = 1, 2, ...NP$.

If $J(\vec{p0}_{i,G})$ is better than $J(\vec{pBest}_{i,G})$ according to the constraint handling of Deb [13], $\vec{pBest}_{i,G}$ take the value of $\vec{p0}_{i,G}$. Then $\vec{U1}_{i,G}$ and $\vec{U2}_{i,G}$ are randomly generated as $U1_{j,i,G} = 0.3 + rand_j(0, 1)(0.9 - 0.3)$ and $U2_{j,i,G} = 0.3 + rand_j(0, 1)(0.9 - 0.3)$, $\forall$ $j = 1, 2, ...D$ and $\forall$ $i = 1, 2, ...NP$.

```
BEGIN
  G = 0
  Creating a swarm of particles p0_{i,G=0} ∀ i = 1, 2, ...NP
  Creating a auxiliar swarm of particles p1_{i,G=0}, ∀ i = 1, 2, ...NP
  Initialize v_{i,G=0}, pBest_{i,G=0}, gBest_{i,G=0}, U1_{i,G=0}
  and U2_{i,G=0}, ∀ i = 1, 2, ...NP , and χ
  Evaluate J(gBest_{i,G=0}), ∀ i = 1, 2, ...NP
  For G = 0 to MaxGen
    For i = 1 to NP
      Evaluate J(p0_{i,G}) and J(pBest_{i,G})
      If J(p0_{i,G}) is better than J(pBest_{i,G}) (according to Deb)
        pBest_{i,G} = p0_{i,G}
      EndIf
      For j = 1 to D
        U1_{j,i,G} = 0.3 + rand_j(0, 1)(0.9 − 0.3)
        U2_{j,i,G} = 0.3 + rand_j(0, 1)(0.9 − 0.3)
        v_{i,j,G} = χ * v_{i,j,G} + U1_{j,i,G} * (pBest_{i,j,G} − p0_{i,j,G})
        +U2_{j,i,G} * (gBest_{i,j} − p0_{i,j,G})
        v_{i,j,G} is limit between v_j^{min} and v_j^{max}
        p1_{i,j,G} = p0_{i,j,G} + v_{i,j,G}
        p1_{i,j,G} is limit between p_j^{min} and p_j^{max}
      EndFor
      Evaluate J(p1_{i,G})
      If J(p1_{i,G}) is better than J(p0_{i,G}) (according to Deb)
        p0_{i,G} = p1_{i,G}
      EndIf
    EndFor
    Update gBest_{i,G}, ∀ i = 1, 2, ...NP
    G = G + 1
  EndFor
END
```

Figure 2: Pseudocode algorithm PSO.

With χ, $\vec{pBest}_{i,G}$ and $\vec{gBest}_{i_G}$ is update $\vec{v}_{i,G}$ and bounded by $[v_j^{min}, v_j^{max}]$, $\forall$ $i = 1, 2, ...NP$ and $\forall$ $j = 1, 2, ...D$.

Also $\vec{p1}_{i,G}$ is update by $\vec{v}_{i,G}$ and $\vec{p0}_{i,G}$, and limited by $[p_j^{min}, p_j^{max}]$, $\forall$ $i = 1, 2, ...NP$ and $\forall$ $j = 1, 2, ...D$.

$J(\vec{p1}_{i,G})$ is evaluated . If $J(\vec{p1}_{i,G})$ is better than $J(\vec{p0}_{i,G})$ according to the constraint handling of Deb, $\vec{p0}_{i,G}$ take the value of $\vec{p1}$.

Finally is update the best solution $\vec{gBest}_{i,G}$, found in the neighborhood of $\vec{p0}_{i,G}$, $\forall$ $i = 1, 2, ...NP$ in a star topology.

C. Selection mechanism

The selection mechanism proposed by Deb can be set as follows: The constraint handling of Deb [13] are the selection mechanism proposes that the fitness $J(\vec{a})$ is better than the fitness $J(\vec{b})$ if any of the following conditions is met:

- $\vec{a}$ and $\vec{b}$ are feasible and $\vec{a}$ dominates $\vec{b}$.
- $\vec{a}$ is feasible and $\vec{b}$ is not feasible.
- $\vec{a}$ and $\vec{a}$ are not feasible, but $\vec{a}$ have fewer restrictions violated.

The pseudocode of PSO is show in figure.

IV. Results and Discussion

In this work, the PSO algorithm is programmed in Matlab on a Windows 7 platform. Computational experiments were performed on a PC with a 2.83 GHz Intel Core Quad 2 processor and 4 GB of RAM. The parameters of the PSO algorithm are proposed as follows: the swarm size consists of 50 particles i.e., $NP = 50$, the acceleration coefficients given by ϕ_1 and ϕ_2 are set to be in the interval $(0.3, 0.9)$ and the velocity v is chosen accordingly to the bounds of the design variable vector $\bar{p}$, i.e., $v_{max} = p^{max}/2$ and $v_{min} = p^{min}/2$. The stop criterion is when the number of generations is fulfilled i.e., $G_{Max} = 15$ and the back time interval is selected as $\Delta w = 50ms$.

The parameters of the simulation results provided by the Euler method are set as: final time $t_f = 1.5$, integration step $\Delta t = 5ms$ with an initial condition $x(0) = [0, 0, 0]^T$. The uncertainties in the DC motor are chosen such that its parameters varies sinusoidally at 10% from their nominal values and the load torque of $\tau_L = 0.05Nm$ is applied at the interval $[0.7, 1.1]s$. The nominal values of the DC parameters are shown in Table I. The controller gains k_p =34524, $k_d = 368$ are proposed with the velocity reference $w_r = 52.35rad/s$.

Showing the effects of the inertia weight in the adaptive control of DC motor based on PSO algorithm, four different experiments are proposed. Those experiments consist on using different inertia weight as follows: $w = [0.2, 0.4, 0.6, 0.8]$. Ten independent runs are carried out for each experiment.

The results of independent runs for each experiment are presented in Table II. The last column indicates the convergence time of the PSO algorithm in each integration step of the closed-loop system. All experiments have a convergence time around $0.51s$. Future research involves the laboratory testing of the ACPSO and hence the reduction of the convergence time around the sampling time (at least $5ms$). The term $mean(|\dot{e}|)$ and $std(|\dot{e}|)$ are the mean and the standard deviation of the absolute velocity error $|\dot{e}|$ in the time interval from the setting time to the final time. Based on the results in Table II, it is observed that the lowest velocity error's average ($mean(|\dot{e})|$) is given by using the inertia weight value of 0.2 in experiment 1 and the lowest standard deviation's average ($std(|\dot{e})|$) is given with the inertia weight value of 0.4 in experiment 2. Those are marked in boldface in Table II. The highest value of the velocity error average and the standard deviation average is found by assigning the value of 0.8 in the inertia weight for the experiment 4. This indicates that the inertia weight of the PSO algorithm affects the behavior of the control performance in the stabilization

Nominal Parameters	**Value**	**Unit**
J_0	0.000345	Nms^2
k_m	0.394600	Nm
b_0	0.000585	Nms^2
R_a	9.665000	Ω
k_e	0.413300	$V/rads$
L_a	0.102440	H
τ_L	0	Nm

Table I: Nominal parameters of the DC Motor.

Inertia	Run	$mean(\|\dot{e}\|)$	$std(\|\dot{e}\|)$	Convergence time $[h]$
0.2	1	0.3373	0.4237	0.5119
0.2	2	0.3747	0.6204	0.5224
0.2	3	0.3452	0.5201	0.5121
0.2	4	0.3212	0.4362	0.5082
0.2	5	0.3108	0.4882	0.5129
0.2	6	0.3587	0.5538	0.5199
0.2	7	0.3773	0.5117	0.5185
0.2	8	0.3389	0.5255	0.5158
0.2	9	0.3629	0.5079	0.5110
0.2	10	0.3247	0.4197	0.5110
Average		**0.3451**	0.5007	0.5143
0.4	1	0.3464	0.4648	0.5106
0.4	2	0.3039	0.3409	0.5189
0.4	3	0.3706	0.6108	0.5186
0.4	4	0.3432	0.4485	0.5188
0.4	5	0.4054	0.8620	0.5100
0.4	6	0.3440	0.3808	0.5080
0.4	7	0.3594	0.5512	0.5204
0.4	8	0.3768	0.4618	0.5135
0.4	9	0.3481	0.4013	0.5189
0.4	10	0.3279	0.3571	0.5208
Average		0.3525	**0.4879**	0.5158
0.6	1	0.3494	0.3933	0.5133
0.6	2	0.3766	0.4633	0.5078
0.6	3	0.418	0.5134	0.5097
0.6	4	0.3491	0.4619	0.5168
0.6	5	0.3517	0.4037	0.5219
0.6	6	0.4458	1.0422	0.5094
0.6	7	0.3879	0.4952	0.5172
0.6	8	0.3879	0.4952	0.5179
0.6	9	0.3954	0.4907	0.5127
0.6	10	0.3455	0.3791	0.5111
Average		0.3807	0.513	0.5137
0.8	1	0.4689	0.6541	0.5111
0.8	2	0.4391	0.5677	0.5183
0.8	3	0.4204	0.4718	0.5133
0.8	4	0.5625	1.6499	0.5190
0.8	5	0.4634	0.5904	0.5120
0.8	6	0.3918	0.4606	0.5181
0.8	7	0.3987	0.4932	0.5122
0.8	8	0.3616	0.3983	0.5105
0.8	9	0.5229	1.5652	0.5180
0.8	10	0.4453	0.5901	0.5212
Average		0.4474	0.7441	0.5153

Table II: Performance of the adaptive control based on PSO for the DC motor.

of the motor's velocity.

A better stabilization of the DC motor's velocity under the effect of uncertainties is carried out if the velocity error's average is decreased. In order to statistically confirm that the selection of the inertia weight as $w = 0.2$ in Experiment 1, presents a better performance than others three experiments, non-parametric statistical test are included. The pairwise comparison "Wilcoxon signed ranks test" is used for this purpose and in Table III those results are shown. Therefore, the pairwise statistical comparisons state the following: $i)$ The Experiment 1 outperforms the Experiment 3 and 4 with a level of significance $\alpha = 0.001$. $2)$ When the comparison of the Experiment 1 is with the Experiment 2, the Experiment 1 does not present a significant improvement over Experiment 2. However, the probability of presenting a better stabiliza-

Comparison	R^+	R^-	$p - value$
Experiment 1 versus Experiment 2	34	21	0.278
Experiment 1 versus Experiment 3	55	0	0.001
Experiment 1 versus Experiment 4	55	0	0.001

Table III: Comparison results with wilcoxon signed rank test.

tion in the closed-loop system of the DC motor with the Experiment 1 is 72.2%. In addition, in the Experiment 1 the evolution of the velocity error through the time presents more deviation from the reference, it is observed in the standard deviation.

In Fig. 3 the behavior of the stabilization of the motor's velocity at $52.35 rad/s$ with the control signal based on ACPSO is shown for the run 1 of Experiment 1. It is observed that the ACPSO compensate the changes on both the dynamic parameters and the load of the DC motor such that regulation of the velocity to the desired reference is almost not affected.

V. Conclusion

In this paper an adaptive control based on Particle Swarm Optimization is proposed to control the velocity of the DC motor. The empirical study state that the inertia weight of the ACPSO is an important factor to be considered in order to adequately control the DC motor.

Simulation results show the performance of the ACPSO under parametric uncertainties. The results presented in this paper indicates that this approach can stabilize the motor's velocity in spite of such uncertainties.

Future research involves the laboratory testing of the proposed ACPSO on an experimental platform. The main issue is the convergence time of the ACPSO that must be smaller than the sampling time (at least $5ms$).

Acknowledgment

The authors acknowledges support from COFAA, EDI and SIP of the Instituto Politécnico Nacional through projects no. SIP-20151212 and no. SIP-20152102, and the support from the Consejo Nacional de Ciencia y Tecnología (CONACyT) through project no. 182298.

References

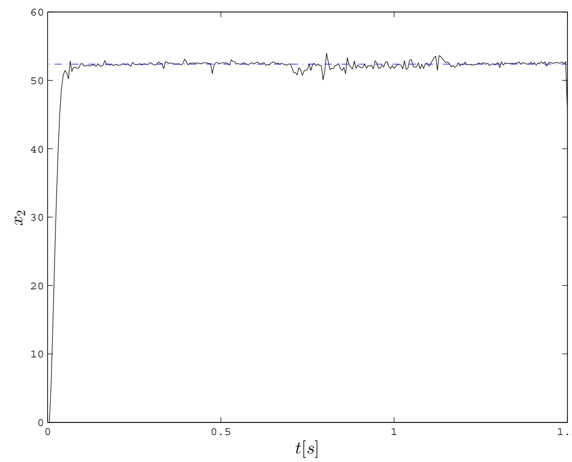
[1] J. Guerrero-Castellanos, M. Villarreal-Cervantes, J. Sánchez-Santana, and S. Ramírez-Martínez, "Seguimiento de trayectorias de un robot móvil (3,0) mediante control acotado," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 11, no. 4, pp. 426–434, 2014.

[2] K. J. Astrom and B. Wittenmark, *Adaptive Control*. Dover Publications, 2008.

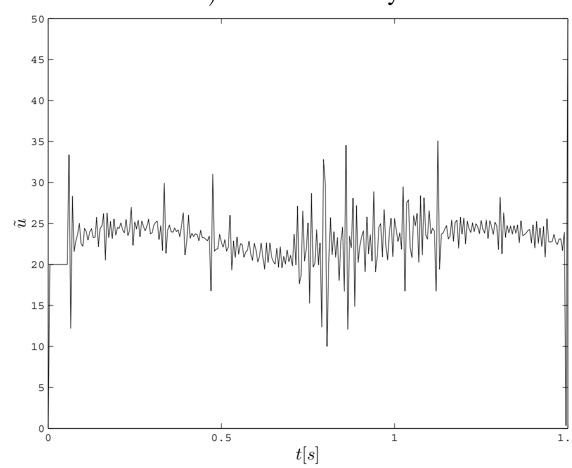
[3] J. Zhou and Y. Wang, "Real-time nonlinear adaptive backstepping speed control for a pm synchronous motor," *Control Engineering Practice*, vol. 13, no. 10, pp. 1259 – 1269, 2005.

[4] K. Nouri, R. Dhaouadi, and N. B. Braiek, "Adaptive control of a nonlinear dc motor drive using recurrent neural networks," *Applied Soft Computing*, vol. 8, no. 1, pp. 371 – 382, 2008.

[5] C.-F. Hsu and B.-K. Lee, "Fpga-based adaptive pid control of a dc motor driver via sliding-mode approach," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11 866 – 11 872, 2011.



$a)$ Motor's velocity



$b)$ Control signal

Figure 3: Behavior of the closed-loop system of the DC motor with the ACPSO.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[7] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[8] O. Sahed, K. Kara, and M. Hadjili, "Pso-based fuzzy predictive control," in *Annual Conference on IEEE Industrial Electronics Society (IECON)*, Oct 2012, pp. 2416–2421.

[9] S. Mu, K. Tanaka, S. Nakashima, D. Alrijadjis, and H. Tomimoto, "Intelligent pid control based on pso type nn for usm," in *IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, Oct 2014, pp. 612–615.

[10] M. Salehi, G. Vossoughi, M. Vajedi, and M. Brooshaki, "Impedance control and gain tuning of flexible base moving manipulators using pso method," in *International Conference on Information and Automation(ICIA)*, June 2008, pp. 458–463.

[11] C. Guo, Y. Zhang, X. You, X. Chen, and Y. Zhang, "Optimal control of continuous annealing process using pso," in *IEEE International Conference on Automation and Logistics (ICAL)*, Aug 2009, pp. 602–605.

[12] K. R., *Electric Motor Drives: Modeling, Analysis and Control*. Prentice Hall, 2001.

[13] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311 – 338, 2000.