DESDE 2013
https://repository.uaeh.edu.mx/revistas/index.php/icbi/issue/archive
Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI
ISSN: 2007-6363

Publicación Semestral Pädi Vol. 11 No. Especial 2 (2023) 32–38

Optimal Tuning of a Bounded $e-$Modified Adaptive Control Law using a Particle Swarm Optimization algorithm
Sintonización optima de un control adaptable acotado con $e-$modificación, mediante un algoritmo de Optimización de Enjambre de Partículas

Olga L. Jiménez-Morales [a,*], Diego Tristán-Rodríguez [1], Rubén Garrido [1], Efrén Mezura-Montes [b]

[a] *Dept. of Automatic Control, CINVESTAV-IPN, 07360, Mexico city, Mexico.*
[b] *Artificial Intelligence Research Center ,Universidad Veracruzana, Xalapa, Veracruz, Mexico.*

Abstract

This paper presents the gain tuning of an adaptive control law by means of Particle Swarm Optimization (PSO). The restrictions imposed on the particles in the PSO are obtained from the stability analysis of the adaptive control law. In this way, the PSO produces particles associated with optimal gains that simultaneously guarantee closed-loop stability and the minimization of the Fitness Function. The adaptive controller employs the velocity and acceleration of the desired trajectory signal for constructing the regressor vector used in the update law. In addition, a new technique is proposed for bounding the parameter estimates allowing them to remain within certain prescribed limits. The performance of the adaptive law tuned using the PSO is evaluated by experiments on a low-cost servo system.

Keywords: Adaptive Control, Optimization, PSO, DC servomotor.

Resumen

Este trabajo presenta una estrategia de sintonización de las ganancias de una ley de control adaptable mediante la Optimización por Enjambre de Partículas (PSO). Las restricciones impuestas a las partículas en el PSO se obtienen a partir del análisis de estabilidad de la ley de control adaptable. De esta forma, el PSO produce partículas asociadas a ganancias óptimas que garantizan simultáneamente la estabilidad en lazo cerrado y la minimización de la función objetivo. El controlador adaptable emplea la velocidad y la aceleración de la señal de trayectoria deseada para construir el vector regresor utilizado en la ley de adaptación propuesta. Además, se propone una nueva técnica de acotación para los parámetros estimados que permite mantenerlos dentro de límites prescritos. El desempeño de la ley adaptable sintonizada mediante PSO se evalúa mediante experimentos en un servosistema de bajo costo.

Palabras Clave: Control adaptable, Optimización, PSO, Servomotores de CD.

1. Introduction

Knowledge of the parameters of a model of the plant to be controlled is of great importance for the design of classical and advanced control laws. These parameters are not always exactly known and may exhibit changes over time. A possible solution to this problem is the use of adaptive controllers, which dispense the exact knowledge of the plant parameters and may improve performance in face of parametric changes. A wide variety of adaptive controllers have been reported in the literature, and they are classified as direct and indirect, and are designed for trajectory tracking, reference model tracking or regulation tasks (Åström y Wittenmark, 2013), (Narendra, 2013), (Sastry, 1989), (Sastry y Bodson, 2011).

On the other hand, the effect of disturbances and measurement noise produce what is called parametric drift in the update laws that produce the parameter estimates. This problem may prevent the tracking error from converging to zero or may induce closed-loop instability. In (Narendra y Annaswamy, 2012) (Narendra, 2013), (Sun, 1995) several solutions to this problem is proposed, one of the them is the so called *e*-modification, which adds an extra term to the update law to counteract the

[*] Autor para correspondencia: olga.jimenez@cinvestav.mx
Correo electrónico: olga.jimenez@cinvestav.mx (Olga L. Jiménez-Morales), dtristan@ctrl.cinvestav.mx (Diego Tristán-Rodríguez), garrido@ctrl.cinvestav.mx(Rubén Garrido), emezura@uv.mx (Efrén Mezura-Montes).

Historial del manuscrito: recibido el 17/03/2023, última versión-revisada recibida el 06/06/2023, aceptado el 09/06/2023 publicado el 11/09/2023. **DOI:** https://doi.org/10.29057/icbi.v11iEspecial2.10713

parametric drift.

Another aspect to be considered in the design of an adaptive controller in practice is the construction of the regressor vector used in the adaptive law. In many cases the regressor is designed using measured signals. This approach is unfeasible if the noise level in the measurements is large, which translates into poor performance. Interestingly enough, in (Sadegh y Horowitz, 1990), (Lewis *et al.*, 2003) in the case of the adaptive control of robot manipulators, the Desired Compensation Adaptation Law (DCAL) employs the position, velocity and acceleration of the desired trajectory to build the regressor. This approach allows computing off-line the nonlinear terms related to the inertia and Coriolis matrices as well as the gravity vector thus reducing the computational burden when computing the DCAL.

On the other hand, as far as the adaptive control design is concerned, the gain tuning is performed using classical techniques such as Zigler-Nichols rules, an approach applied to auto-tuners (Astrom, 1987), and statistical or heuristic tuning (Åström *et al.*, 1993). Currently, the use of intelligent optimization techniques in the area of automatic control and mechatronics has become an useful tool for tuning the gains of control algorithms. There are a variety of intelligent optimization techniques in the literature, related to the tuning of classical controllers such as PID (Chang, 2022), (Ibrahim *et al.*, 2019), active disturbance rejection control algorithms (Liu *et al.*, 2022), (Hu y Chen, 2022), and adaptive controllers (Mohideen *et al.*, 2013), (Rodríguez-Molina *et al.*, 2019b), among others.

On the other hand, several metaheuristics including the Genetic Algorithm, Simulated Annealing, and Tabu Search has been used for PID controller tuning applied to bioprocess control (Roeva y Slavov, 2012). An interesting survey on applying intelligent optimization techniques to PID controller tuning is found in (Joseph *et al.*, 2022) whereas reference (Rodríguez-Molina *et al.*, 2020) gives an account on how to apply multiobjective optimization to controller tuning.

The aforementioned works give rise to the objective of this work. It is intended to design a new adaptive control algorithm applied to the control of servomotors which has the following features. The proposed algorithm employs only noise-free velocity and acceleration of the desired trajectory in the regressor vector. On the other hand, the proposed adaptive controller is based on a new smooth bounding technique for the parameter estimates combined with the standard *e* modification. In addition, Particle Swarm Optimization techniques are used for the tuning of the adaptive controller gains.

The outline of the paper is as follows. Section 2 describes the mathematical model and the parametrization of the dynamical model of a servo system. Section 3 is devoted to the proposed adaptive controller. Section 4 presents the Particle Swarm Optimization algorithm. The Optimization of the Parameters is developed in Section 5. Experiments with a laboratory prototype are reported in Section 6. The paper ends with some concluding remarks.

2. Mathematical Model of a Servo System

The model of the servo system under study assumes that a power amplifier working in current mode drives a DC motor, the latter endowed with a position sensor. The model is described as:

$$J\ddot{y} = -f\dot{y} + ku + \bar{d} \tag{1}$$

where y, $\dot{y}$ and $\ddot{y}$ are respectively the DC motor angular position, velocity and acceleration, u is the control voltage, J corresponds to the DC motor and load inertias, f is the viscous friction coefficient, k is the input gain, which depends on the power amplifier gain and the motor torque constant, and $\bar{d}$ corresponds to bounded external disturbances.

Multiplying (1) by $\frac{1}{k}$ gives:

$$\frac{J}{k}\ddot{y} = -\frac{f}{k}\dot{y} + u + \frac{\bar{d}}{k} \tag{2}$$

which has the next alternative writing:

$$\theta_1\ddot{y} + \theta_2\dot{y} = u + d \tag{3}$$

with $\theta_1 = \frac{J}{k} > 0$, $\theta_2 = \frac{f}{k} > 0$ and $d = \frac{\bar{d}}{k}$ where $|d| \leq D$.

3. Proposed Adaptive Controller

3.1. Control law

The following algorithm computed using known parameters is proposed to control the servo model (3):

$$u = k_P e + Kr + \theta_1\ddot{y}_d + \theta_2\dot{y}_d \tag{4}$$

with

$$\begin{aligned} e &= y_d - y \\ r &= e + \dot{e} \\ \dot{r} &= \dot{e} + \ddot{e} \end{aligned} \tag{5}$$

Variable $e = y_d - y$ is the tracking error and the terms y_d, $\dot{y}_d$ and $\ddot{y}_d$ are the desired reference signal and its first and second time derivatives. The feedback part of controller (4) is equivalent to a standard Proportional Derivative (PD) control law:

$$k_P e + Kr = k_P e + K(\dot{e} + e) = (k_P + K)e + K\dot{e} \tag{6}$$

and the proportional and derivative gains correspond to $k_p = k_P + K$ and $k_d = K$ respectively.

Now, assume that the control law is computed using parameter estimates $\hat{\theta}_1$ and $\hat{\theta}_2$ instead of θ_1 and θ_2:

$$u = Kr + k_P e + \hat{\theta}_1\ddot{y}_d + \hat{\theta}_2\dot{y}_d \tag{7}$$

The dynamics of the closed-loop system is obtained by substituting (7) into (3):

$$\theta_1\ddot{y} + \theta_2\dot{y} = Kr + k_P e + \hat{\theta}_1\ddot{y}_d + \hat{\theta}_2\dot{y}_d + d \tag{8}$$

Adding and subtracting $\theta_1\ddot{y}_d$ and $\theta_2\dot{y}_d$ in (8), considering the tracking error e and its time derivative, and the parametric errors defined as:

$$\tilde{\theta} = \begin{bmatrix} \tilde{\theta}_1 \\ \tilde{\theta}_2 \end{bmatrix} = \begin{bmatrix} \hat{\theta}_1 - \theta_1 \\ \hat{\theta}_2 - \theta_2 \end{bmatrix} \tag{9}$$

produce:

$$\theta_1\ddot{e} + \theta_2\dot{e} = -Kr - k_P e - \tilde{\theta}_1\ddot{y}_d - \tilde{\theta}_2\dot{y}_d - d \tag{10}$$

Substituting $\ddot{e} = \dot{r} - \dot{e}$ and $\dot{e} = r - e$ defined in (5) into (10) yields:

$$\theta_1 \dot{r} + \theta_2 r = \theta_1 \dot{e} + \theta_2 e - \tilde{\theta}_1 \ddot{y}_d - \tilde{\theta}_2 \dot{y}_d - Kr - k_P e - d \quad (11)$$

Define:

$$\phi = \begin{bmatrix} (\ddot{y}_d + \dot{e}) & (\dot{y}_d + e) \end{bmatrix}^T \quad (12)$$

$$\phi_d = \begin{bmatrix} \ddot{y}_d & \dot{y}_d \end{bmatrix}^T \quad (13)$$

$$\theta = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T \quad (14)$$

$$\begin{aligned} \Phi &= \phi^T \theta - \phi_d^T \theta \\ &= \theta_1(\ddot{y}_d + \dot{e}) + \theta_2(\dot{y}_d + e) - \theta_1 \ddot{y}_d - \theta_2 \dot{y}_d \\ &= \theta_1 \dot{e} + \theta_2 e \end{aligned} \quad (15)$$

The above definitions allow writing (11) as follows:

$$\theta_1 \dot{r} + \theta_2 r = \Phi - \phi_d^T \tilde{\theta} - Kr - k_P e - d \quad (16)$$

where $\tilde{\theta}$ is defined in (9).

3.2. Update law

In order to proceed consider the next assumption:

Assumption 1: Parameters $\theta_i, i = 1, 2$ are bounded as $\theta_{\min i} \leq \theta_i \leq \theta_{\max i}$ and the bounds are known.

Now, define the next function:

$$\theta_i = f(\eta_i) = \sigma_i[1 + \tanh(\eta_i)] + \theta_{\min i} \quad (17)$$

$$\sigma_i = \frac{1}{2}[\theta_{\max i} - \theta_{\min i}], \quad i = 1, 2 \quad (18)$$

for $\theta_i, \eta_i, i = 1, 2$. Function (17) guaranties that for any $\eta_i \in \mathbb{R}$ the value of $\theta_i \in \Omega_i = [\theta_{\min i}, \theta_{\max i}]$, $i = 1, 2$. Function $\tanh(\cdot)$ corresponds to the hyperbolic tangent.

The function:

$$V_{si} = \ln\cosh(\hat{\eta}_i) - \ln\cosh(\eta_i) - (\hat{\eta}_i - \eta_i)\tanh(\eta_i) \quad (19)$$

linked to (17), is positive definite with a minimum $\hat{\eta}_i = \eta_i$ $i = 1, 2$.

Moreover, note also that:

$$\begin{aligned} \tilde{\theta}_i &= \hat{\theta}_i - \theta_i \\ &= \sigma_i[1 + \tanh(\hat{\eta}_i)] + \theta_{\min i} \\ &\quad - \sigma_i[1 + \tanh(\eta_i)] + \theta_{\min i} \\ &= \sigma_i[\tanh(\hat{\eta}_i) - \tanh(\eta_i)] \end{aligned} \quad (20)$$

Bearing in mind function (17), computing the estimate $\hat{\theta}_i$ of θ_i for $i = 1, 2$ is performed as follows:

$$\begin{aligned} \hat{\theta}_i &= \sigma_i[1 + \tanh(\hat{\eta}_i)] + \theta_{\min i} \\ \sigma_i &= \frac{1}{2}[\theta_{\max i} - \theta_{\min i}], \quad i = 1, 2 \\ \dot{\hat{\eta}}_1 &= \gamma_1 \ddot{y}_d r - \gamma_1 \kappa \hat{\theta}_1 |r| \\ \dot{\hat{\eta}}_2 &= \gamma_2 \ddot{y}_d r - \gamma_2 \kappa \hat{\theta}_2 |r| \end{aligned} \quad (21)$$

3.3. Stability issues

The stability analysis of the closed-loop system is performed by taking into account the error dynamics (16) and the update laws (21). To this end, the next Lyapunov function candidate:

$$V = \frac{1}{2}\theta_1 r^2 + \frac{1}{2}k_P e^2 + \frac{\sigma_1}{\gamma_1}V_{s1} + \frac{\sigma_2}{\gamma_2}V_{s2} \quad (22)$$

allows obtaining the following conditions:

The term:

$$\lambda = \theta_2 + K - \frac{1}{2k_P}(\theta_2 - \theta_1)^2 - \theta_1 \quad (23)$$

is positive if:

$$K > \theta_{\max 1} + \frac{1}{2k_P}(\theta_{\max 1} + \theta_{\max 2})^2 \quad (24)$$

Besides, the Lyapunov function time derivative (22) is negative definite if

$$|r| > \frac{\Lambda}{\lambda} \quad (25)$$

where

$$\Lambda = D + \frac{1}{4}\kappa\theta_{\max 1}^2 + \frac{1}{4}\kappa\theta_{\max 2}^2 \quad (26)$$

which allows concluding that the solutions of the closed-loop system are uniformly ultimately bounded.

4. Particle Swarm Optimization (PSO) algorithm

Created in 1995 by Kennedy and Eberhart, this algorithm is based on the mathematical abstraction of a simplified social model behavior of bird flocks (Kennedy y Eberhart, 1995). The potential solutions that the PSO generates are represented as a swarm of particles moving towards a search space defined by the optimization problem. The movement of the particles is called flight and the main idea is that every particle finds a place in the search space that minimizes a cost function.

It is worth mentioning that the PSO algorithm is one of the most widely applied swarm intelligence methods due to its ability to solve many complex optimization problems in different areas (Wang *et al.*, 2018). It is also important to highlight that unlike the PSO algorithms employed in the literature to tune adaptive controllers Rodríguez-Molina *et al.* (2019c); Chang (2022); Rodríguez-Molina *et al.* (2019a), the PSO algorithm presented here considers the stability conditions of the closed-loop system to define the set of feasible search solutions.

4.1. Theory

The PSO algorithm with inertia weight ω-PSO algorithm (Sidorov, 2018) is composed of N particles of dimension L. The particles are defined as $z_n(k) = [z_{n,1}, \ldots, z_{n,L}]^T$ where $n \in [1, \ldots, N]$. For every particle, there exists a velocity vector $v_n(k) = [v_{n,1}, \ldots, v_{n,L}]^T$. The following second order discrete-time dynamic system describes the evolution of the algorithm:

$$\begin{aligned} v_n(k+1) &= \omega v_n(k) \quad (27) \\ &\quad + c_1 \mathrm{rand}()(\mathrm{pBest}(k,n) - z_n(k)) \\ &\quad + c_2 \mathrm{rand}()(\mathrm{gBest}(k) - z_n(k)) \\ z_n(k+1) &= z_n(k) + v_n(k+1) \quad (28) \end{aligned}$$

where the initial conditions are $z_{n,l}(0) = \text{rand}()$ and $v_{n,l}(0) = 0$ such that $l \in [1, \dots, L]$ and rand() is the uniformly distributed random number function. The parameter $\omega \in [0, 1)$ is called the inertia weight and $c_1 \in [0, 1)$, $c_2 \in [0, 1)$ are called the learning factors. The terms pBest $(k, n) \in \mathbb{R}^L$ and gBest $(k) \in \mathbb{R}^L$ are defined as follows:

$$\text{pBest}(k, n) = \arg \min_{0 \le s \le k} J(z_n(s)) \tag{29}$$

$$\text{gBest}(k) = \arg \min_{0 \le s \le k, 1 \le j \le n} J(z_j(s)) \tag{30}$$

and $J(\cdot)$ is a fitness function. The pseudocode of the PSO algorithm is shown in Fig. 1.

The PSO algorithm doesn't have the ability to take into account the limits of the search space in the particles to ensure closed-loop stability, then, it is important to define a set of feasible solutions. This set is called $\Omega \in \mathbb{R}^D$. In this work, the Projection Boundary method (Juarez-Castillo *et al.*, 2019) is implemented in the PSO algorithm to limit the particle positions to the set Ω. In this technique if an element of the solution $z_n(k+1)$ falls out of Ω, the Projection Boundary method projects it into the boundary of Ω:

$$z_{n,d}(k+1) = \begin{cases} z_{n,d}(k+1) & \text{if } \min(\Omega_d) < z_{n,d}(k+1) < \max(\Omega_d) \\ \min(\Omega_d) & \text{if } \min(\Omega_d) > z_{n,d}(k+1) \\ \max(\Omega_d) & \text{if } \max(\Omega_d) < z_{n,d}(k+1) \end{cases} \tag{31}$$

where $d = 1, \dots, D$.

5. Parameter optimization

Carrying out the optimization requires performing a dynamic simulation of the servo system, an due to this fact, it is necessary to use the parameter estimates $\hat{\theta}_{1I}$ and $\hat{\theta}_{2I}$ of θ_1, θ_2 obtained according to (36).

Algorithm 1 ω-PSO algorithm

```
Create a random initial swarm z_n(k) ∀n
Realize the dynamic simulation and evaluate the fitness
  function J(z_n(k)) ∀n
Calculate pBest(k, n) ∀n
Calculate gBest(k)
while stop condition == false do
    for n = 1 to N do
        Compute the velocity vector v_n(k + 1)
        Perform the flight z_n(k + 1)
        Realize the dynamic simulation and evaluate the
  fitness function J(z_n(k))
        Calculate pBest(k, n)
    end for
    Calculate gBest(k)
end while
```

Figure 1: ω-PSO algorithm

The parameters to estimate using the PSO algorithm are k_P and K from the control law (7) and the adaptation law parameters γ_1, γ_2 and κ in (21). Thus, the particles of the PSO algorithm are defined as:

$$z_n := [k_P, K, \gamma_1, \gamma_2, \kappa]^T \tag{32}$$

On the other hand, the fitness function J is defined as follows:

$$J = \int_{T_1}^{T_2} \Big(w_1|e| + w_2|\dot{e}| + w_3 \exp(|\hat{\theta}_1 - \hat{\theta}_{1I}|) + w_4 \exp(|\hat{\theta}_2 - \hat{\theta}_{2I}|) + w_5|u| + w_6 \Big|\frac{du}{dt}\Big| \Big) dt \tag{33}$$

where $T_1 = 0$, $T_2 = 100$, $\exp(\cdot)$ is the exponential function, the weights $w_1 = 0.3, w_2 = 0.3, w_3 = 0.1, w_4 = 0.1, w_5 = 0.1, w_6 = 0.1$ define the importance given to every integrand, e is the tracking error and $\dot{e}$ is the velocity error, u is the control signal and the last term corresponds to its time derivative.

The set of feasible solutions is defined as follows taking into account the stability analysis:

$$\Omega = \Big(k_P > 0, \gamma_1 > 0, \gamma_2 > 0, \kappa > 0, \infty > K > \theta_{\max 1} + \frac{1}{2k_P}(\theta_{\max 1} + \theta_{\max 2})^2 \Big) \tag{34}$$

Note that the stability condition (24) is used to define the set Ω.

The tuning procedure of the parameters of the PSO is performed using the irace package, which gives a number of $n = 10$ particles, a value of $NP = 400$ iterations, an inertia weight of $\omega = 0.9$, and values of the learning parameters of $c_1 = 0.9$ and $c_2 = 0.8$. Each run of the algorithm stops after 4000 (n×NP) evaluations.

The results of the optimization are depicted in Table 1. A sampling of 30 independent runs with the same initial parameters is carried out to observe the behavior of the PSO algorithm. The results of the tests are displayed in Table 2. This table shows that the test produces minimum, median, and average values very similar and a standard deviation close to zero. Therefore, the behavior of the PSO algorithm is considered consistent.

Table 1: Parameters obtained by means of the PSO algorithm.

Control algorithm gains	**Adaptation law variables**
$k_P = 1.44$	$\gamma_1 = \gamma_2 = 7.5$
$K = 1.1$	$\kappa = 0.002$

Table 2: Statistical results.

	Evaluation of J
Minimum	112.71
Median	112.76
Mean	113.00
Standard Deviation	0.4886

6. Experiments Results

6.1. Laboratory prototype

The Laboratory prototype used for implementing and testing the proposed controller consists of a personal computer and a Quanser Consulting Q2-USB data acquisition board (see Fig. 2). The control algorithm is coded in the MATLAB/SIMULINK programming platform under the Quanser QUARC real-time environment with a sampling time of 1 ms and the Euler01 integration method. The control signal output produced by the data acquisition board is processed through a linear amplifier integrated circuit LM675 from Texas Instruments, which drives a Makeblock 180 smart encoder motor (see Fig. 3) whose load corresponds to an inertia disk.

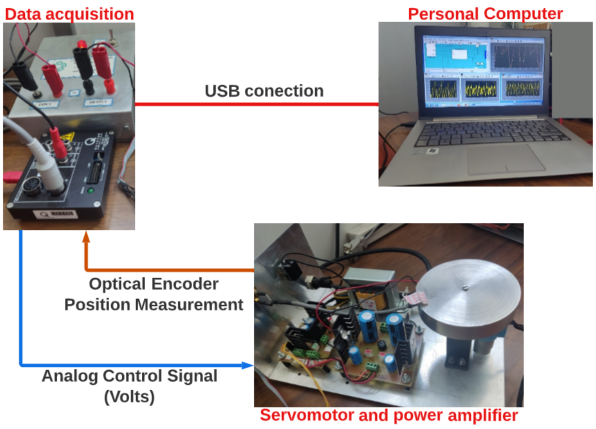


Figure 2: Laboratory prototype.

Rated voltage	7.4 V
No-load current	240 mA
Rotation speed	178 ± 10 RPM
Start torque	5 kg·cm
Feedback	Optical encoder
Rotation	unlimited
Encoder accuracy	360 ppr
Weight	0.0615 kg
Availability	highly available
Cost (USD)	30 USD

Figure 3: Makeblock servomotor and its technical characteristics (Makeblock, 2019).

6.2. *Experimental results*

The next second-order dynamic system generates a reference signal y_d and its first and second-time derivatives:

$$\ddot{y}_d = -2\zeta\omega_n\dot{y}_d - \omega_n^2 y_d + \omega_n^2 r_m \tag{35}$$

where $\zeta = 1$ and $\omega_n = 5$ to make the reference signal y_d as close as possible to the reference defined as $r_m = 0.5\sin(0.3t) + 0.3\sin(0.2t) + 0.1\sin(0.7t)$. This signal is amplified using a gain of 2 and processed by a low-pass filter with a cut-off frequency of 5 rad/s.

The parameters presented in Table 1 are used to implement the adaptive control law algorithm (7) using the adaptation law (21). The initial condition are set to $\eta_1(0) = -10$ and $\eta_2(0) = 10$. To obtain the $\theta_{\min i}$ and $\theta_{\max i}$ values shown in Table 3, which are used in the evaluation of the proposed adaptive controller the following relationships are used:

$$\begin{aligned}\theta_{\min 1} &= \frac{1}{2}\hat{\theta}_{1I}\\ \theta_{\max 1} &= 2\hat{\theta}_{1I}\\ \theta_{\min 2} &= \frac{1}{2}\hat{\theta}_{2I}\\ \theta_{\max 2} &= 2\hat{\theta}_{2I}\end{aligned} \tag{36}$$

where $\hat{\theta}_{1I} = \dfrac{1}{\hat{\theta}_{LS2}}$ and $\hat{\theta}_{2I} = \dfrac{\hat{\theta}_{LS1}}{\hat{\theta}_{LS2}}$. $\hat{\theta}_{LS1}$ and $\hat{\theta}_{LS2}$ are parameters previously estimated using the off-line Least Squares algorithm (further details are found in (Morales *et al.*, 2022)).

To assess the performance of the controller (7) the following performance criteria are used: the integral squared error (ISE) and the integral of the absolute value of the control signal variation (IACV). These indices are expressed as follows:

$$ISE = \int_{T_1}^{T_2} k[e(t)]^2 dt \tag{37}$$

$$IACV = \int_{T_1}^{T_2} \left|\frac{du(t)}{dt}\right| dt \tag{38}$$

where k represents a scaling factor and $\{T_1, T_2\}$ defines a time interval where the performance indexes are computed. For the comparative study a value of $k = 100$ is used with $T_1 = 20s$ and $T_2 = 30s$. Table 4 shows the corresponding results.

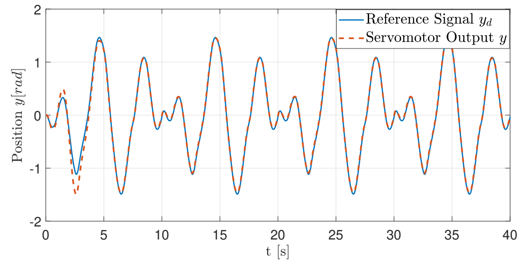
Fig. 4(a) depicts the reference and the output signals of the proposed adaptive controller (7), (21). It is noted that during the first 5 seconds there is a large tracking error. Afterwards, the tracking error decreases. This can be corroborated in Fig. 4(c) where the error signal exhibits a large peak and the decreases. In addition, the ISE index in Table 4 is very small. Fig. 4(b) shows that the control signal never exceeds the upper and lower limits of the power amplifer of $\pm 7.4V$. Besides, the tuning generated by the PSO algorithm produces very low levels of chattering. This is the result of taking into account the term $|\frac{du}{dt}|$ in the cost function. If this term is not included in (33) large leves of chattering could appear in the control signal, which may degrade the closed-loop performance.

Table 3: Limits used in the implementation of the Adaptive Control law.

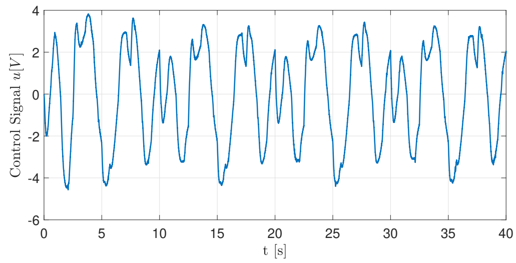
Bounds on the parameter estimates	
$\theta_{\min 1} = 0.04$	$\theta_{\max 1} = 0.16$
$\theta_{\min 2} = 0.78$	$\theta_{\max 2} = 3.12$

Table 4: Performance of the adaptive controller using update law (7), (21).

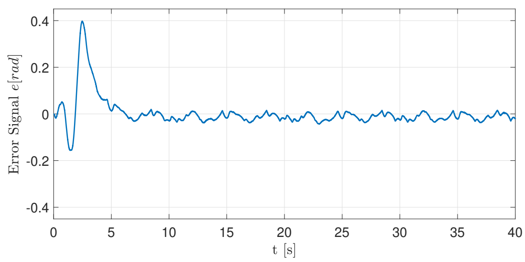
Control Law	**ISE**	**IACV**
Proposed adaptive controller (7), (21)	0.0036	114



(a) Servomotor position output vs desired reference signal.

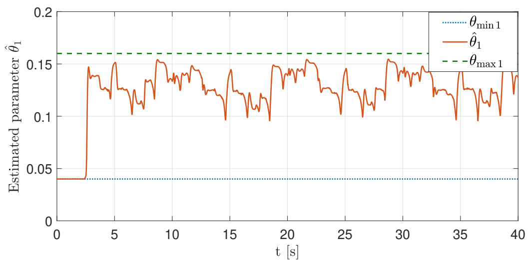


(b) Control signal.

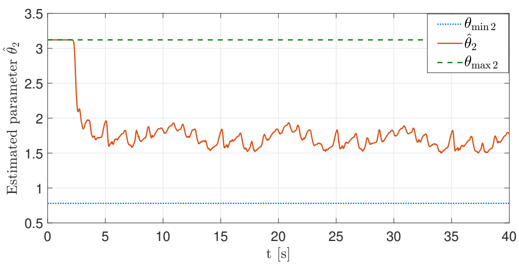


(c) Position error.

Figure 4: Proposed adaptive controller (7), (21).



(a) Estimated parameter $\hat{\theta}_1$ bounded by $\theta_{\min 1}, \theta_{\max 1}$.



(b) Estimated parameter $\hat{\theta}_2$ bounded by $\theta_{\min 2}, \theta_{\max 2}$.

Figure 5: Estimated parameters $\hat{\theta}$ using the adaptive law (21).

Fig. 5 shows the graph of the parameter estimates produced by the updated law (21). They remain within the bounds $\theta_{\min i}$ and $\theta_{\max i}$ and converge to almost constant values. Note that the estimate $\hat{\theta}_2$ remains in its lower limit $\theta_{\max}$ less than 5 s. This behavior is due to the large initial condition in η_2. However, after this time period $\hat{\theta}_2$ evolves towards an almost constant value. A similar comment applies to $\hat{\theta}_1$.

7. Conclusion

The results reported in this work indicate that the trajectories of the closed-loop system are uniformly and ultimately bounded. Moreover, the experiments show that the proposed adaptive controller with the parameters tuned by the PSO algorithm produces a small tracking error. Furthermore, as a novelty, the PSO algorithm employed here uses the theoretical stability results to build the set of feasible solutions. Future work includes extending the results to more complex systems including robot manipulators and quadrotors, which may have multiple terms to tune. Moreover, in the case of quadrotors, optimization is also required to cope with limited energy resources. A potential limitation of the proposed optimization method is that the stability conditions of more complex systems may be complicated and difficult to use to build the set of feasible solutions. Another potential problem is the fact that for larger systems, tuning more parameters would require the development of more advanced PSO algorithms. Moreover, the computational burden increases due to a larger number of parameters and a larger set of feasible solutions to explore.

References

Astrom, K. J. (1987). Adaptive feedback control. *Proceedings of the IEEE*, 75(2):185–217.

Åström, K. J., Hägglund, T., Hang, C. C., y Ho, W. K. (1993). Automatic tuning and adaptation for pid controllers-a survey. *Control Engineering Practice*, 1(4):699–714.

Åström, K. J. y Wittenmark, B. (2013). *Adaptive control*. Courier Corporation.

Chang, W.-D. (2022). An improved particle swarm optimization with multiple strategies for pid control system design. *International Journal of Modeling and Optimization*, 12(2).

Hu, J. y Chen, W. (2022). Design of active disturbance rejection controller for dynamic positioning based on improved particle swarm optimization. *Mathematical Problems in Engineering*, 2022.

Ibrahim, M. A., Mahmood, A. K., y Sultan, N. S. (2019). Optimal pid controller of a brushless dc motor using genetic algorithm. *Int J Pow Elec & Dri Syst ISSN*, 2088(8694):8694.

Joseph, S. B., Dada, E. G., Abidemi, A., Oyewola, D. O., y Khammas, B. M. (2022). Metaheuristic algorithms for pid controller parameters tuning: Review, approaches and open problems. *Heliyon*, pp. 309–399.

Juarez-Castillo, E., Acosta-Mesa, H., y Mezura-Montes, E. (2019). Adaptive boundary constraint-handling scheme for constrained optimization. *Soft Computing*, pp. 1–34.

Kennedy, J. y Eberhart, R. (1995). Particle swarm optimization. En *Proceedings of ICNN'95 - International Conference on Neural Networks*, volumen 4, pp. 1942–1948.

Lewis, F. L., Dawson, D. M., y Abdallah, C. T. (2003). *Robot manipulator control: theory and practice*. CRC Press.

Liu, X., Gao, Q., Ji, Y., Song, Y., y Liu, J. (2022). Active disturbance rejection control of quadrotor uav based on whale optimization algorithm. En *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 351–356. IEEE.

Makeblock (2019). 180 optical encoder motor. https://store.makeblock.com/180-optical-encoder-motor.

Mohideen, K. A., Saravanakumar, G., Valarmathi, K., Devaraj, D., y Radhakrishnan, T. (2013). Real-coded genetic algorithm for system identification and tuning of a modified model reference adaptive controller for a hybrid tank system. *Applied Mathematical Modelling*, 37(6):3829–3847.

Morales, O. J., Tristán-Rodríguez, D., y Garrido, R. (2022). Experimental evaluation of an adrc law using a teaching platform. En *2022 XXIV Robotics Mexican Congress (COMRob)*, pp. 78–83.

Narendra, K. S. (2013). *Adaptive and learning systems: theory and applications*. Springer Science & Business Media.

Narendra, K. S. y Annaswamy, A. M. (2012). *Stable adaptive systems*. Courier Corporation.

Rodríguez-Molina, A., Mezura-Montes, E., Villarreal-Cervantes, M. G., y Aldape-Pérez, M. (2020). Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem. *Applied Soft Computing*, 93:106–342.

Rodríguez-Molina, A., Villarreal-Cervantes, M. G., y Aldape-Pérez, M. (2019a). An adaptive control study for the dc motor using meta-heuristic algorithms. *Soft Computing*, 23:889–906.

Rodríguez-Molina, A., Villarreal-Cervantes, M. G., Mezura-Montes, E., y Aldape-Pérez, M. (2019b). Adaptive controller tuning method based on online multiobjective optimization: A case study of the four-bar mechanism. *IEEE transactions on cybernetics*, 51(3):1272–1285.

Rodríguez-Molina, A., Villarreal-Cervantes, M. G., Mezura-Montes, E., y Aldape-Pérez, M. (2019c). Adaptive controller tuning method based on online multiobjective optimization: A case study of the four-bar mechanism. *IEEE transactions on cybernetics*, 51(3):1272–1285.

Roeva, O. y Slavov, T. (2012). Pid controller tuning based on metaheuristic algorithms for bioprocess control. *Biotechnology & Biotechnological Equipment*, 26(5):3267–3277.

Sadegh, N. y Horowitz, R. (1990). An exponentially stable adaptive control law for robot manipulators. *IEEE Transactions on Robotics and Automation*, 6(4):491–496.

Sastry, S. (1989). *Adaptive Control; Stability, Convergence and Robustness*. Prentice Hall Information and System Sciences Series.

Sastry, S. y Bodson, M. (2011). *Adaptive Control: Stability, Convergence and Robustness*. Courier Corporation.

Sidorov, G. (2018). *Artificial Intelligence*. Alfa-Omega.

Sun, P. A. I. J. (1995). *Robust Adaptive Control*. Prentice Hall.

Wang, D., Tan, D., y Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft computing*, 22(2):387–408.