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, Diego Tristán-Rodríguez, Rubén Garrido, Efrén Mezura-Montes

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...
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 in the regressor features. The proposed algorithm employs only noise-free velocity and its time derivative, and the parametric drift.

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:

\[
J\ddot{y} = -f\dot{y} + ku + \bar{d}
\]

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:

\[
\frac{J}{k}\ddot{y} = -f\frac{\dot{y}}{k} + u + \frac{\bar{d}}{k}
\]

which has the next alternative writing:

\[
\theta_1\dot{y} + \theta_2\ddot{y} = u + d
\]

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):

\[
u = k_pe + Kr + \theta_1\dot{y}_d + \theta_2\ddot{y}_d
\]

with

\[
e = y_d - y
\]

\[
r = e + \dot{e}
\]

\[	f = \dot{e} + \ddot{e}
\]

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:

\[
kpe + Kr = kpe + K(e + \dot{e}) = (kp + K)e + K\dot{e}
\]

and the proportional and derivative gains correspond to \(kp = k_p + K\) and \(kd = K\) respectively.

Now, assume that the control law is computed using parameter estimates \(\hat{\theta}_1\) and \(\hat{\theta}_2\) instead of \(\theta_1\) and \(\theta_2\):

\[
u = Kr + kpe + \hat{\theta}_1\dot{y}_d + \hat{\theta}_2\ddot{y}_d
\]

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

\[
\theta_1\dot{y} + \theta_2\ddot{y} = Kr + kpe + \hat{\theta}_1\dot{y}_d + \hat{\theta}_2\ddot{y}_d + d
\]

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

\[
\hat{\theta} = \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{bmatrix} = \begin{bmatrix} \hat{\theta}_1 - \theta_1 \\ \hat{\theta}_2 - \theta_2 \end{bmatrix}
\]

produce:

\[
\theta_1\dot{e} + \theta_2\ddot{e} = -Kr - kpe - \hat{\theta}_1\ddot{y}_d - \hat{\theta}_2\ddot{y}_d - d
\]
Substituting \( \ddot{e} = \dot{r} - \dot{e} \) and \( \dot{e} = r - e \) defined in (5) into (10) yields:

\[
\theta_1 \ddot{r} + \theta_2 r = \theta_1 \dot{e} + \theta_2 e - \tilde{\theta}_1 \tilde{y}_d - \tilde{\theta}_2 \tilde{y}_d - Kr - k_p e - d
\]

(11)

Define:

\[
\phi = \begin{bmatrix} \tilde{y}_d + \dot{e} \\ \dot{y}_d + e \end{bmatrix}^T
\]

(12)

\[
\phi_d = \begin{bmatrix} \dot{y}_d & \tilde{y}_d \end{bmatrix}^T
\]

(13)

\[
\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^T
\]

(14)

\[
\Phi = \phi^T \theta - \phi_d^T \theta
\]

\[
= \theta_1 (\dot{y}_d + \dot{e}) + \theta_2 (\dot{y}_d + e) - \theta_1 \tilde{y}_d - \theta_2 \tilde{y}_d
\]

(15)

The above definitions allow writing (11) as follows:

\[
\theta_1 \ddot{r} + \theta_2 r = \Phi - \phi_d^T \tilde{\theta} - Kr - k_p e - d
\]

(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

\[
\theta_{\text{min}} \leq \theta_i \leq \theta_{\text{max}}, \quad i = 1, 2
\]

and the bounds are known.

Now, define the next function:

\[
\theta_i = f(\eta_i) = \sigma_i [1 + \tanh(\eta_i)] + \theta_{\text{min}}, \quad i = 1, 2
\]

(17)

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

(18)

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

The function:

\[
V_{\dot{u}} = \ln \cosh(\hat{\eta}_i) - \ln \cosh(\eta_i) - (\hat{\eta}_i - \eta_i) \tanh(\eta_i)
\]

(19)

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

Moreover, note that:

\[
\tilde{\theta}_i = \hat{\theta}_i - \theta_i
\]

\[
= \sigma_i [1 + \tanh(\hat{\eta}_i)] + \theta_{\text{min}}
\]

\[
- \sigma_i [1 + \tanh(\eta_i)] + \theta_{\text{min}}
\]

\[
= \sigma_i [\tanh(\hat{\eta}_i) - \tanh(\eta_i)]
\]

(20)

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

\[
\hat{\theta}_1 = \gamma_1 \eta_{i,r} - \gamma_1 k_b \dot{\theta}_1 |r|
\]

\[
\hat{\theta}_2 = \gamma_2 \eta_{i,r} - \gamma_2 k_b \dot{\theta}_2 |r|
\]

(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:

\[
V = \frac{1}{2} \theta_1 r^2 + \frac{1}{2} k_p e^2 + \frac{1}{\gamma_1} V_{\dot{z}_1} + \frac{1}{\gamma_2} V_{\dot{z}_2}
\]

(22)

allows obtaining the following conditions:

The term:

\[
\lambda = \theta_2 + K - \frac{1}{2k_p} (\theta_2 - \theta_1)^2 - \theta_1
\]

is positive if:

\[
K > \theta_{\text{max}} + \frac{1}{2k_p}(\theta_{\text{max}} + \theta_{\text{max}})^2
\]

(23)

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

\[
|r| > \Lambda
\]

(25)

where

\[
\Lambda = D + \frac{1}{4} k_B^2 \theta_{\text{max}}^2 + \frac{1}{4} k_B^2 \theta_{\text{max}}^2
\]

(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 and 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 \( \omega \)-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:

\[
v_{n}(k + 1) = \omega v_{n}(k)
\]

(27)

\[
+ c_1 \text{rand}() (p\text{Best}(k, n) - z_n(k))
\]

\[
+ c_2 \text{rand}() (g\text{Best}(k) - z_n(k))
\]

\[
z_n(k + 1) = z_n(k) + v_n(k + 1)
\]

(28)
where the initial conditions are \( z_{n,0}(0) = \text{rand}() \) and \( v_{n,0}(0) = 0 \) such that \( l \in [1, \ldots, L] \) and \( \text{rand}() \) 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 \( p\text{Best} \ (k,n) \in \mathbb{R}^L \) and \( g\text{Best} \ (k) \in \mathbb{R}^L \) are defined as follows:

\[
p\text{Best}(k,n) = \arg\min_{0 \leq s \leq k} J(z_n(s)) \tag{29}
g\text{Best}(k) = \arg\min_{0 \leq s \leq k} J(z_k(s)) \tag{30}
\]

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 \( \Omega \). In this technique if an element of the solution \( z_n(k+1) \) falls out of \( \Omega \), the Projection Boundary method projects it into the boundary of \( \Omega \):

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

where \( d = 1, \ldots, D \).

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 \( \hat{\theta}_{1l} \) and \( \hat{\theta}_{2l} \) of \( \theta_1, \theta_2 \) obtained according to (36).

\[
\text{Algorithm 1: } \omega\text{-PSO algorithm}
\]

1. Create a random initial swarm \( z_n(k) \) \forall n
2. Realize the dynamic simulation and evaluate the fitness function \( J(z_n(k)) \) \forall n
3. Calculate pBest \((k, n) \) \forall n
4. Calculate gBest \((k) \)
5. while stop condition = false do
6. for \( n = 1 \) to \( N \) do
7. Compute the velocity function \( v_n(k+1) \)
8. Perform the flight \( z_n(k+1) \)
9. Realize the dynamic simulation and evaluate the fitness function \( J(z_n(k)) \)
10. Calculate pBest \((k, n) \)
11. end for
12. Calculate gBest \((k) \)
13. end while

The parameters to estimate using the PSO algorithm are \( k_P \) and \( K \) from the control law (7) and the adaptation law parameters \( \gamma_1, \gamma_2 \) and \( \kappa \) 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:

\[
J = \int_{T_1}^{T_2} \left( w_1|e| + w_2|\dot{e}| + w_3\exp((\hat{\theta}_1 - \theta_{1l}))/\exp((\hat{\theta}_2 - \theta_{2l})) + w_4|\dot{\theta}_1| + w_5|\dot{\theta}_2| + w_6\frac{|du|}{dt} \right) dt \tag{33}
\]

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:

\[
\Omega := \left\{ k_P > 0, \gamma_1 > 0, \gamma_2 > 0, \kappa > 0, \infty > K > \theta_{\text{max}} + \frac{1}{2k_P} (\theta_{\text{max}} + \theta_{\text{max}}) \right\} \tag{34}
\]

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

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 \((nxNP)\) 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>
<thead>
<tr>
<th>Table 1: Parameters obtained by means of the PSO algorithm.</th>
<th>Table 2: Statistical results.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control algorithm gains</td>
<td>Evaluation of ( J )</td>
</tr>
<tr>
<td>( k_P ) = 1.44</td>
<td>Minimum 112.71</td>
</tr>
<tr>
<td>( \gamma_1 ) = ( \gamma_2 ) = 7.5</td>
<td>Median 112.76</td>
</tr>
<tr>
<td>( K ) = 1.1</td>
<td>Mean 113.00</td>
</tr>
<tr>
<td>( \kappa ) = 0.002</td>
<td>Standard Deviation 0.4886</td>
</tr>
</tbody>
</table>

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.

6.2. Experimental results

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

\[
y_d = -2\zeta\omega_n y_d - \omega_n^2 y_d + \omega_n^2 r_m
\]

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 \( \theta_{\text{min}} \) and \( \theta_{\text{max}} \) values shown in Table 3, which are used in the evaluation of the proposed adaptive controller the following relationships are used:

\[
\begin{align*}
\theta_{\text{min}1} &= \frac{1}{2} \hat{\theta}_{11} \\
\theta_{\text{max}1} &= 2\hat{\theta}_{11} \\
\theta_{\text{min}2} &= \frac{1}{2} \hat{\theta}_{21} \\
\theta_{\text{max}2} &= 2\hat{\theta}_{21}
\end{align*}
\]

where \( \hat{\theta}_{11} = \frac{1}{\hat{\theta}_{LS2}} \) and \( \hat{\theta}_{21} = \frac{\hat{\theta}_{LS1}}{\hat{\theta}_{LS2}} \). \( \hat{\theta}_{LS1} \) 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:

\[
\begin{align*}
\text{ISE} &= \int_{T_1}^{T_2} k[e(t)]^2 dt \\
\text{IACV} &= \int_{T_1}^{T_2} |\frac{du(t)}{dt}| dt
\end{align*}
\]

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. The tuning generated by the PSO algorithm produces very low levels of chattering. This is the result of taking into account the variation (IACV). These indices are expressed as follows:

![Figure 2: Laboratory prototype.](image1)

![Figure 3: Makeblock servomotor and its technical characteristics.](image2)

![Table 3: Limits used in the implementation of the Adaptive Control law.](image3)

![Table 4: Performance of the adaptive controller using update law (7), (21).](image4)
Fig. 5 shows the graph of the parameter estimates produced by the updated law (21). They remain within the bounds $\theta_{\text{min},i}$ and $\theta_{\text{max},i}$ and converge to almost constant values. Note that the estimate $\hat{\theta}_i$ remains in its lower limit $\theta_{\text{min},i}$ less than 5 s. This behavior is due to the large initial condition in $\eta_i$. However, after this time period $\hat{\theta}_i$ 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


