

Intelligent swarm algorithms for optimizing nonlinear sliding mode controller for robot manipulator

Suhad Qasim G. Haddad¹, Hanan A. R. Akkar²

¹Department of Computer Engineering, University of Technology, Iraq, Baghdad, Baghdad, Iraq

²Department of Electrical Engineering, University of Technology, Iraq, Baghdad, Baghdad, Iraq

Article Info

Article history:

Received Nov 28, 2020

Revised Mar 17, 2021

Accepted Mar 27, 2021

Keywords:

Artificial intelligence

Particle swarm optimizer

Sliding mode control

Social spider optimizer

Swarm intelligence

ABSTRACT

This work introduces an accurate and fast approach for optimizing the parameters of robot manipulator controller. The approach of sliding mode control (SMC) was proposed as it documented an effective tool for designing robust controllers for complex high-order linear and nonlinear dynamic systems operating under uncertain conditions. In this work Intelligent particle swarm optimization (PSO) and social spider optimization (SSO) were used for obtaining the best values for the parameters of sliding mode control (SMC) to achieve consistency, stability and robustness. Additional design of integral sliding mode control (ISMC) was implemented to the dynamic system to achieve the high control theory of sliding mode controller. For designing particle swarm optimizer (PSO) and social spider optimization (SSO) processes, mean square error performances index was considered. The effectiveness of the proposed system was tested with six degrees of freedom robot manipulator by using (PUMA) robot. The iteration of SSO and PSO algorithms with mean square error and objective function were obtained, with best fitness for (SSO = $4.4876 e^{-6}$ and (PSO) = $3.4948 e^{-4}$.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Suhad Qasim G. Haddad

Department of Computer Engineering

University of Technology, Al Sinaa Street, Baghdad, Iraq

Email: suhadhadad.sh@gmail.com

1. INTRODUCTION

Programming the multi degrees of freedom (DOF) of robot arms in order to maximize their functionality has recently remained a challenging job, time-consuming and costly operation in particular. Indeed, the coupling effects of multiple joints must be taken into account by controlling robot manipulator parameters [1]. In this work, programmable universal manipulation arm (PUMA) robot, which has six degrees of freedom (6-DOF), all are rotary joints with serial connections, was used as a study case. The first three joints are used to control the robot's handle position. The second three joints are to obtain the orientation of the robot's wrist locus. PUMA robot manipulator is widely used in medical, automotive, education and other important applications. The parameters and dimensions of this robot were all known and documented in different literatures [2]. Figure 1 illustrates the global model of PUMA robot the 6-DOF manipulator.

Designing a stable and strong controller is an important part for sensitive and various applications for robot manipulator. The most common nonlinear model-based controller is sliding mode controller (SMC) which has been properly applied in various applications such as motor control, space system, automatic flight control and finally robot control. It is considered as a powerful advance stable robotic manipulator control system that can achieve and solve the most important tasks in control systems, which are robustness and

stability [3]. Integral sliding mode controller (ISMC) was suggested in this work to boost the manipulator output to accomplish the desired tasks with high stability.

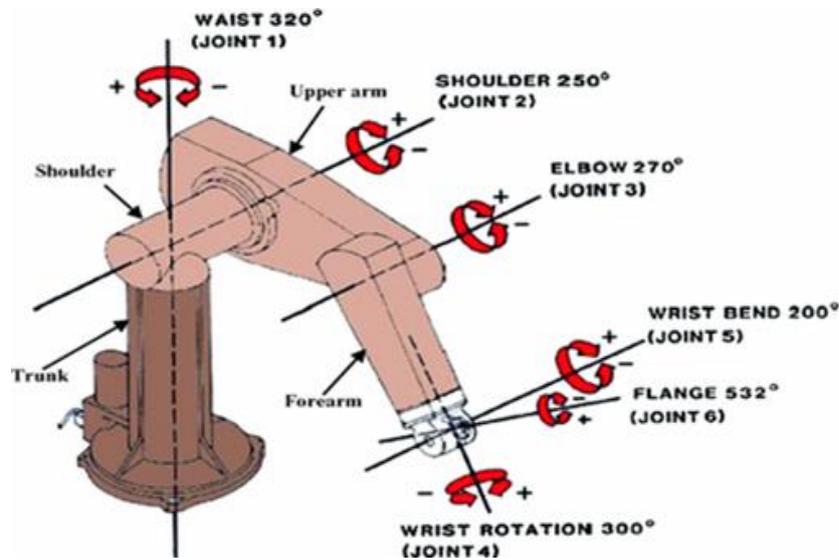


Figure 1. The global model of puma 560 robot manipulator

The essential benefits of the SMC method are achieving of asymptotic reactivity and stability in implementation regarding to all notched internal and external uncertainties and disturbances. The SMC can be split into two key components, which are the discontinuous part (τ_{dis}) that's used to design appropriate tracking performance centered on linear methodology requiring identical fast switching. But it is caused to system instability and the occurrence of chatter. The second part is equivalent controller (τ_{eq}) which is the effect of nonlinear terms that induced reliability and used to fine-tune the sliding surface slopes [4]. In addition, SMC has two major disadvantages, namely are chattering phenomenon in control response, and nonlinear equivalent analogous functional formulation in indeterminate and uncertain dynamic parameters [5]. The rising in switching frequency of the control in discontinuous issue happened due to a phenomenon called (chattering), which can be considered as an unwanted feature that appear in the control behavior of SMC. It is a limited value with restricted frequency of oscillation that can cause some severe mechanical problems, such as distortion and heating of robot manipulator mechanical components. This chattering influences on the stability of the system, due to this reason chattering effect should be cancelled or minimized. So, adjusting the sliding mode control parameters is an important part to reduce the chattering disadvantage and developing a stable coefficient for nonlinear controllers [6]. In recent times, artificial intelligence (AI) techniques have been implemented to overcome the disadvantage in conventional controller design, which is an effective tool for solving many applications with intelligent techniques. These innovative approaches are methods of biological inspired optimization, which can randomly transfer a solution of one candidate into a new one with better fitness function [7]. Researchers have been obsessed with insect or animal species' collective intelligent actions in nature, such as bird flocks, ant colonies, fish schools, bee swarms, where the relation between the collection of insects or animals is referred to as swarm behaviour. This division of AI is often referred to as swarm intelligence (SI), which deals with the mutual behavior of swarms by the active coordination and interaction of individuals without supervision. SI provides a benefits that including scalability, fault tolerance, flexibility, speed, mobility, autonomy, parallelism, and adaptation [8]. In this work, particle swarm optimization (PSO), and social spider optimization (SSO), which are an efficient algorithms of SI, were suggested to adjust the nonlinear coefficients and optimizing the parameters of the proposed controllers. These algorithms can create the perfect solution, with a high quality and a lower computing time than other techniques, with progressively stable assembly characteristics, they efficiently change the parameters with sufficient convergence to achieve impressive acceleration of the system [9].

This paper has been arranged accordingly. Some related works are presented in the second section. The third section deals with the analysis, modelling and simulation of the dynamic model of PUMA robot. Section four concentrates on the strategy of controller model, and the method that used in designing SMC and ISMC for controlling the dynamic part of the robot. Section 5 discusses the method of PSO and SSO

algorithms. The outcome of the debate and simulation is shown in the sixth section. The last section in the paper introduces the conclusion and future work.

2. LITERATURE REVIEW

Researchers have been working hard to control the robot manipulator and to create best solutions for the dynamic model, to obtain stable response with minimum steady state error. Several controllers have been developed with various SI algorithms. A concise summary of previous related works is shown in this section. In [10] mentioned that to solve the trajectory monitoring problem for a nonlinear design of wheeled robot, PSO has been used to get the best values for the parameters of sliding mode control. In [11], the robotic device stability was demonstrated using Lyapunov impedance based technique, with PSO algorithm for optimization the parameters of the control. Jalali [12] introduced an adaptive sliding-mode controller with PSO algorithm for optimizing PUMA robot manipulator, this controller can respond and adapt itself to change system parameters according to the external intervention. PSO algorithm was used to enhance the parameters of the sliding function and minimize the chattering action. Boundjou [13] developed a stable adaptive controller with Particle Swarm Optimizer for automatic and systematic tuning of PID gains for two degrees of freedom robot arm, thus optimizing the cost function of the parameters. In [14], evaluating of SMC efficiency with PID was done, taking into account the ziegler-nichols (ZN) and the PSO tuning algorithm implemented in the sliding surface. The computational and simulation results showed that compared to traditional ZN variables the SMC is able to do well with the PSO variables added to the sliding board.

PID controller auto-tuning system was implemented for PUMA robot manipulators [15]. Two methods of SI multi-objective optimization were tested, namely multi objective cuckoo scan (MOCS) and multi objective particle swarm optimization (MOPSO). Comparison was made between the results of the two algorithms in the case of achieving a predefined trajectory with a reasonable tracking accuracy. Sliding mode controller with boundary layer was presented for dynamic design with control of 6-DOF robot arm manipulator type IRB-120 [16]. The chattering that generates unreliable signal with high frequency oscillation in the sliding mode control was eliminated by using the boundary layer which provides the controller with stabilization and improve the total performance. New technique was presented for robot manipulators with a robust high degree composite, super-twisting of sliding mode controller (HOS-TSMC) [17]. The suggested approach extends the traditional and conventional controller of sliding mode (TSMC) and the approximate state of sliding mode (ESMC) to ameliorate the chatter. To minimize the influence of numerous external disturbances, the non-linear simulator of the saturation integral backstepping controller was implemented and the particle swarm algorithm was used to evaluate the controller parameters [18]. Three types of external disturbances (constant, intermittent, and random disturbances) were used separately during the quadrotor aircraft's flight control. The outcome of the simulation reveals that in anti-disturbance power, the saturation integral backstepping controller was much stronger than the classical backstepping controller and integral back stepping controller. A modern adaptive terminal sliding mode controller with a non-single specialty was used to control the behavior of quad rotor flight [19]. Besides, the controller of backstepping sliding mode, was introduced to control the path of the quadrotor, despite the influence of external disturbances, this leads to quick and reliable observation. The design process was illustrated with Lyapunov principle. Then, the proposed controllers of backstepping, integral sliding mode and the second sliding mode, were investigate with several flights checks. Where the simulation result shows an improvement in the convergence of time, with good robustness and chattering free for control efforts.

Akkar and Haddad [4] introduced a strategy for tuning three types of dynamic control techniques that were developed for the PUMA 560 robot manipulator, which are proportional integral derived (PID), sliding mode control (SMC), and the modified type of integral sliding mode control (ISMC). Intelligent particle swarm optimization (PSO) has been suggested to achieve the best parameters in order to maximize the parameters in the proposed strategies. The PSO optimization approach has improved the parameters of each controller according to the initial values of the particles, such as their swarm size and initial velocity.

3. CONTRIBUTION

In this work, based on the theory of SMC, an integral optimal control law was designed for the 6-DOF PUMA robot manipulator. The integral sliding mode controller (ISMC) was proposed to improve the manipulator output to perform the desired tasks with high stability. Then, particle swarm algorithm for optimization (PSO), and social spider optimization (SSO) algorithm, were suggested to adjust the nonlinear coefficients and optimizing the parameters of the ISMC. Where, adjusting the integral sliding mode control parameters is an important part to reduce the chattering disadvantage and developing a stable coefficient for nonlinear controllers.

4. DYNAMIC DESIGN OF PUMA ROBOT MANIPULATOR

The dynamic equation of multi DOF for PUMA robot manipulator is about the analysis of motion concerning forces. Dynamic design modelling is required for mechanical part model, control and finally in implementing the simulation. It is used to define the parameters of the dynamic and also to explain the relationship between displacement, distance, acceleration and the force that acts on the manipulator of the robot. Lagrange-Euler mathematical model of the robot is used for calculation the control laws and design the dynamic system of the robot. In of a multi DOF for robot manipulator is calculate is being as [20], [21]:

$$A(q)\ddot{q} + N(q, \dot{q}) = \tau \tag{1}$$

Dynamic equation as an outcome for the robot can be described is being as:

$$N(q, \dot{q}) = V(q, \dot{q}) + G(q) \tag{2}$$

$$V(q, \dot{q}) = B(q) [\dot{q} \dot{q}] + C(q) (\dot{q})^2 \tag{3}$$

$$\tau = A(q) \ddot{q} + B(q)[\dot{q} \dot{q}] + C(q)(\dot{q})^2 + G(q) \tag{4}$$

where: $A(q)$:Symmetric positive matrix considered for kinetic energy and inertia matrix,with $n \times n$ dimension. $B(q)$: is for Coriolis torques matrix, with $n \times n (n - 1)/2$ dimension. $C(q)$: is for centrifugal torques, with $n \times n$ dimension. $G(q)$: is for gravity torques, with $n \times 1$ dimension. q : is the joint position (or joint angle), for $q = [q_1, q_2, \dots, q_n]$, \dot{q} :is consider as n- vector for joint velocities, \ddot{q} : is consider as n- vector of accelerations. And τ : is consider as the joint force vector (torque). $[\dot{q}^2]$: that can a vector given by $[\dot{q}_1^2, \dot{q}_2^2, \dots, \dot{q}_n^2]^T$, $[\dot{q}\dot{q}]$: that can a vector given : $[\dot{q}_1\dot{q}_2, \dot{q}_1\dot{q}_3, \dots, \dot{q}_1\dot{q}_n, \dot{q}_2\dot{q}_3, \dots]^T$. The input of the dynamic system is torque matrix in the robotic manipulator arrangement while the outcomes are real variables displacement and joints, as a result it can be written is being as:

$$\ddot{q} = A^{-1}(q) \cdot [\tau - N(q, \dot{q})] \tag{5}$$

All the parameters of the forward and inverse kinematics and matrix of PUMA robot have been computed as mentioned in [2], [22], [23]. In this work only the first, second and third links of PUMA robot manipulator will be taken into consideration. Figure 2 illustrates the block diagram for dynamic and kinematics model for PUMA robot manipulator.

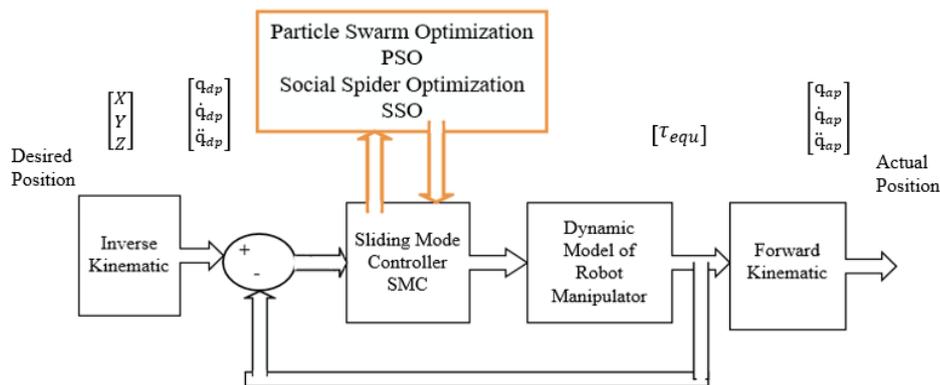


Figure 2. Block diagram for dynamic and kinematics model for PUMA robot

5. METHOD OF DESIGNING SLIDING MODE CONTROL SMC

SMC is applied to impose the system’s state path to pass through the sliding surface, then imposes the state’s system path to "slide" along the switching surface till it stays on the origin [4]. In order to define the equation that control the total design of SMC, the nonlinear input single for dynamic system define is being as:

$$Z_n = \delta(Z) + b(Z)v(t) \tag{6}$$

where (v) is consider as the vector of the control input , Z_n is the n_{th} derivative of the state vector Z . δ is consider as unknown nonlinear function or uncertainty of dynamic, $b(Z)$ is switching [SIGN] function. The key objective of designing SMC is to train the appropriate state desire position Z_d according to the actual joint variables, the trucking error vector will be defined is being as [3], [5].

$$e = \tilde{Z} = Z - Z_d \quad (7)$$

where: Z is for real and actual position, Z_d is the desired position, and \tilde{Z} : is for estimated trucking error vector. According to the theory of the sliding mode controller, sliding surface is the key important part to design this controller, the calculation of time varying for sliding surface $S(z, t)$, and the integral part of sliding surface is shown is being as:

$$S(z, t) = \left(\frac{d}{dt} + \mu\right)^{n-1} (\tilde{Z}) = 0 \quad (8)$$

$$S(z, t) = \left(\frac{d}{dt} + \mu\right)^{n-1} \left(\int_0^t \tilde{Z} dt\right) = 0 \quad (9)$$

μ is the sliding surface slope coefficient and it is positive constant, in this method the main target is to keep the sliding surface slope near to zero. Therefore, best strategies to achieve this is to find the input control v outside the sliding surface and remains on it. To keep the $S(e, \dot{e})$ close to zero, the law control is design to achieve the sliding condition of Lyapunov function [4], [6], is being as:

$$V = \frac{1}{2} S^2 \geq 0 \quad (10)$$

So, the time derivative becomes $\dot{V} = (\dot{S}^2 S)$ and the control u is chosen such that:

$$\dot{S}^2 S \leq \eta |S| \quad (11)$$

where η is consider appositve constant [3], [6], so the sliding surface will have computed is being as:

$$\frac{1}{2} \frac{d}{dt} S^2(z, t) \leq \eta |S(z, t)| \quad (12)$$

when the surface $(S) \approx \text{Zero}$.So , error $e = \hat{Z} = Z - Z_d \approx \text{Zero}$. Let us consider that:

$$S = \dot{e} + \mu e \quad (13)$$

$$\dot{e} = \frac{de}{dt} = \dot{Z} - \dot{Z}_d \quad (14)$$

$$S = (\dot{Z} - \dot{Z}_d) + \mu (Z - Z_d) \quad (15)$$

Derivative of (15), will consider as the change in sliding surface, is being as:

$$\dot{S} = \frac{dS}{dt} = (\ddot{Z} - \ddot{Z}_d) + \mu (\dot{Z} - \dot{Z}_d) \quad (16)$$

$$\text{Since : } \dot{Z} = \delta + v \text{ So : } \dot{S} = \delta + U - \dot{Z}_d + \mu (\dot{Z} - \dot{Z}_d) \quad (17)$$

By imposing $S \rightarrow 0$ also $\dot{S} \rightarrow 0$, If we put $\dot{S} = 0$, in (17), as shown in

$$0 = \delta + U - \dot{Z}_d + \mu (\dot{Z} - \dot{Z}_d) \quad (18)$$

where δ is the uncertainty of dynamic, under this hypothesis we get the best approximation for control (\hat{U}) which can be define as:

$$\hat{U} = -\delta + \dot{Z}_d - \mu (\dot{Z} - \dot{Z}_d) \quad (19)$$

Using the uncertainty switching control law to control the dynamic parameters of sliding mode:

$$U_{dis} = \hat{U} - K(z, t)S_{Sign} \tag{20}$$

where K : is a positive constant function of (z, t) , and S_{Sign} is the switching function define is being as:

$$\begin{cases} S_{Sign} = 1 \text{ if } S > 0 \\ S_{Sign} = 0 \text{ if } S = 0 \\ S_{Sign} = -1 \text{ if } S < 0 \end{cases} \tag{21}$$

Keep the $S(e, \dot{e})$ close to zero in order to satisfy the sliding condition of Lyapunov function. So (9) and (16) can be arranged is being as:

$$\frac{1}{2} \frac{d}{dt} S^2(z, t) = \dot{S} S = [\delta - \hat{\delta} - K S_{SIGN}] S = (\delta - \hat{\delta}) S - K|S| \tag{22}$$

$$S(z, t) = \left(\frac{d}{dt} + \mu\right)^2 \left(\int_0^t \tilde{Z} dt\right) = (\dot{Z} - \dot{Z}_d) + 2\mu(\dot{Z} - \dot{Z}_d) + \mu^2(Z - Z_d) \tag{23}$$

So, the approximation of \hat{U} is computed is being as:

$$\hat{U} = -\hat{\delta} + \ddot{Z}_d - 2\mu(\dot{Z} - \dot{Z}_d) + \mu^2(Z - Z_d) \tag{24}$$

The SMC for robot manipulator with multi DOF is being as. Figure 3 shows the total of the τ_{Total} :

$$\tau_{Total} = \tau_{eq} + \tau_{dis} \tag{25}$$

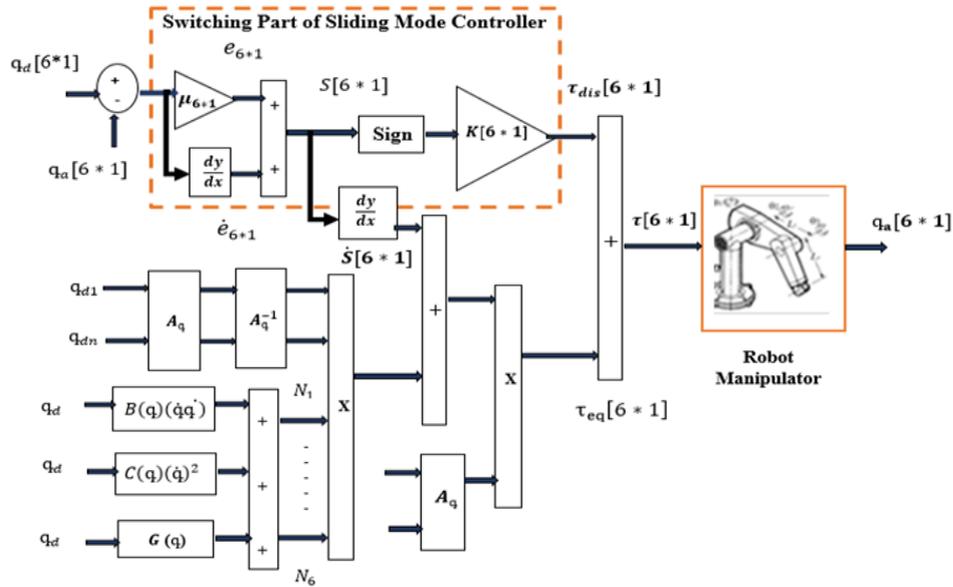


Figure 3. Block diagram for the total equation of the torque in SMC of PUMA robot manipulator

Finally, the dynamic model of PUMA robot manipulator can be computed is being as [4], [6]:

$$\tau_{eq} = [A^{-1} (B + C + G) + \dot{S}] A \tag{26}$$

$$\tau_{dis} = K S_{Sign} \tag{27}$$

$$\tau_{Total} = \tau_{eq} + K S_{Sign} \tag{28}$$

$$\tau_{Total} = [A^{-1} (B + C + G) + \dot{S}] A + K .S_{Sign} \tag{29}$$

5.1. Modified integral approach for sliding mode controller (ISMC)

In order to enhance the performance designed of the proposed controller scheme which is based on SMC, besides improve stability and minimize total error, integral sliding mode controller (ISMC) was designed, our goal is to keep the sliding surface close to zero all the time [4], [6], [10]. The formula that defines ISMC surface is being as:

$$S_{Integral} = \mu e + \dot{e} + \left(\frac{1}{2}\right)^2 \sum e \tag{30}$$

In the SMC model, two parameters (K, μ) must be adjusted. If these parameters are appropriately changed, the controller will reject external disturbances disrupting the tracking path and increase the torque of the robot joints. PSO and SSO are the two algorithms that used in this work to pick the best values to the parameters of both SMC and ISMC. Figure 4 shows the total output torque for ISMC design.

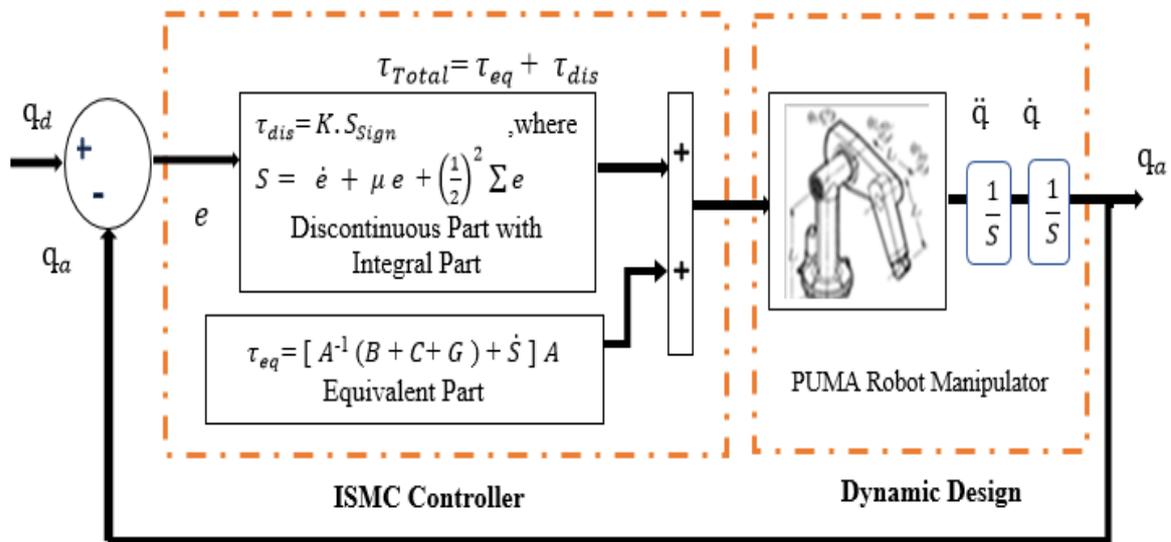


Figure 4. Block diagram illustrates the design of ISMC of PUMA robot manipulator

6. INTELLIGENT SWARM OPTIMIZATION ALGORITHMS

6.1. Particle swarm optimization (PSO)

Particle swarm optimization (PSO) algorithm used initial random solutions called particles, which fly over the search space size by following the obtained optimum particle and their own acquired experiences. The principle of this optimization is to use its best known positions of particles to converge the swarm population in the solution space to a single optimal. The PSO algorithm requires, at each point, changing the location and velocity of the particle to its $Pbest$ and $Gbest$ for each particle, the velocity is modified iteratively by its personal best position, which is found by the particle, and also by the best position found by the particles in its vicinity [24], as shown in (31), (32):

$$\zeta_{i,g}^{(k+1)} = \Phi \cdot \zeta_{i,g}^{(k)} + c_1 \varphi_1 (Pbest_{i,g} - X_{i,g}^{(k)}) + c_2 \varphi_2 (Gbest_{i,g} - X_{i,g}^{(k)}) \tag{31}$$

$$X_{i,g}^{(k+1)} = X_{i,g}^{(k)} + \zeta_{i,g}^{(k+1)} \tag{32}$$

where: i : is for particles number. g : for dimensions' number. Φ : which are two for SMC. X_i : is for g -dimensional position vector ($X_{i1}, X_{i2}, \dots, X_{ig}$). ζ_i : velocity of the particle ($\zeta_{i1}, \zeta_{i2}, \dots, \zeta_{ig}$). $Pbest$: best visited position for the particles. $Gbest$: best position explored in the population. $\varphi_1 \varphi_2$: consider as random integer between 0 and 1. Φ : inertia weight. $c_1 c_2$: positive constants. k : for iteration pointer. Coefficients c_1 and c_2 include the relative weight of the probabilities that accelerate each particle in $Pbest$ and $Gbest$ position. Sufficient choice of inertia weight (Φ) will allow for a balance between global and local exploration, which enables reduced aggregate iteration in order to find a fairly optimum solution. Table 1 illustrates the global pseudo code for PSO [25].

Table 1. Global pseudo code for PSO algorithm

Step No.	Description
1	Generate random initial population and initial parameters (initialization of individuals).
2	While not (The population converges towards the desired and optimum solution or the ultimate iteration obtained).
3	For each dimension in the particle
4	Update particle velocities using (31)
5	Update particles positions using (32)
6	Evaluate fitness function ($X_{i,g}^{(k)}$)
7	If function($X_{i,g}^{(k)}$) < function(Pbest _{i,g}) then (Pbest _{i,g}) ← $X_{i,g}^{(k)}$
8	End If
9	If function($X_{i,g}^{(k)}$) < function(Gbest _{i,g}) then function(Gbest _{i,g}) ← $X_{i,g}^{(k)}$
10	End If
11	End If and Loop (Next generation)

6.2. Social spider optimization (SSO)

The SSO algorithm recognizes two search agents, or two genders of male and female in spider's webs. Each entity is accomplished with a different set of evolutionary operators, based on the gender, which simulate different cooperative activities normally seen in the colony. It is observed that, within a certain range of distance, dominant males mate with their female neighbors and non-dominant males remain in a position close to other network males and rely on nutrients for the latter. In addition to mating, spiders communicate with each other by vibrations and the frequency of the vibrations depends on two important factors: the heaviness of the spider and the distance between the spiders that communicate [26]. The male and female spiders are the search agents whose population can be determined as:

$$N_f = [(0.9 - rand * 0.25) * N_k] \quad (33)$$

$$N_m = N_k - N_f \quad (34)$$

where: N_k , N_f , and N_m denote the total population of all spiders in the communal web, reflecting the number of female spiders and male spiders respectively. The weight that represent the heaviness of each spider in the population W_i is found is being as:

$$W_i = \frac{fitness_i - worst}{best - worst} \text{ for } 0 \leq i \leq N_k \quad (35)$$

where: i : represent the population of the web. N_k : The iteration in the total population. $fitness_i$, $best$ and $worst$ are objective function values. In the common popular web, the spiders connect with each other's by using the vibrations that calculated is being as:

$$V_{i,j} = W_j * e^{-d_{i,j}^2} \quad (36)$$

where: W_j : is related to the heaviness of the spider that transmits the vibration. $d_{i,j}$: is considered as the Euclidean distance computed between the two interacting spiders. The vibrations between spiders in the web may be classified depending on the transmitting spider as shown in:

- The situation where the transmitting spider, S_c is more heavier than the perceiving spider, i , ($W_c > W_i$) contributing to vibration $V_{c,i}$.
- The situation where the transmitting spider, S_b is the best heaviest one in the web which contributing in vibration $V_{b,i}$, ($W_b = \max best (fitness_i)$).
- The situation where the transmitting spider, S_f is a female neighbor contributing in vibration $V_{f,i}$.

The location of the female and male spiders is modified at each iteration of the optimization algorithm using the formula is being as [27]:

$$f_i^{k+1} = \begin{cases} f_i^k + \alpha V_{c,i}(S_c - f_i^k) + \beta V_{b,i}(S_b - f_i^k) + \gamma \left(rand - \frac{1}{2} \right) & \text{if } r_m < PF \\ f_i^k - \alpha V_{c,i}(S_c - f_i^k) - \beta V_{b,i}(S_b - f_i^k) + \gamma \left(rand - \frac{1}{2} \right) & \text{if } r_m \geq PF \end{cases} \quad (37)$$

where: α, β, γ , and r_m are consider as random numbers, that are between [0 or 1]. PF is regarded as the threshold of probability factor that is matched with all the randomly produced numbers while the spiders will

travelled in the space. So, the transition is randomly regulated by the PF factor of likelihood and the motion is generated in relation with the other spiders depending on their vibration through the space of search. In the optimization procedure the male spider m_i^{k+1} , will operated is being as [26]:

$$m_i^{k+1} = \begin{cases} m_i^k + \alpha V_{c,i}(S_f - m_i^k) + \gamma \left(rand - \frac{1}{2} \right) & \text{if } W_{N_{f+i}} > W_{N_{f+m}} \\ m_i^k + \alpha \left(\frac{\sum_{hi=1}^{N_m} m_h^k W_{N_{f+hi}}}{\sum_{hi=1}^{N_m} W_{N_{f+hi}}} - m_i^k \right) & \text{if } W_{N_{f+i}} \leq W_{N_{f+m}} \end{cases} \quad (38)$$

where: $\left(\frac{\sum_{hi=1}^{N_m} m_h^k W_{N_{f+hi}}}{\sum_{hi=1}^{N_m} W_{N_{f+hi}}} - m_i^k \right)$ is considered as the weighted average of the male spider in the web population. The dominant males, as described earlier, will mate with the females in the mating range, which is given is being as:

$$r = \frac{\sum_{j=1}^n (p_j^{high} - p_j^{low})}{2n} \quad (39)$$

where: p_j^{high} and p_j^{low} are assumed to be as the higher and lower initial limits. Table 2 illustrates the pseudo code for SSO algorithm:

Table 2. The pseudo code for SSO algorithm

Step No.	Description
1	Assign the initial parameters value for the SSO algorithm
2	Create the population of spiders and assign memory for them
3	Initial the first position for both male spider and female spider
4	Counting the k iterations in the population (N_k)
5	While $i \leq N_k$
6	Calculate the mating radius for both female and male spiders as in (39)
7	Calculating the spiders' weight as in (35)
8	Calculate the passage of female and male spider's dependent on mutual female and male operators as shown in (37) and (38)
9	Conduct mating between dominant males and females
10	Update solutions if there are heavier spider progenies
11	End the loop

7. RESULTS AND DISCUSSION

In this work, the Matlab/ Simulink structure version R2019b and M-File were used to apply the design strategy of the controller in order to evaluate the performance of the proposed controller design. The sampling time fixed on $\Delta t = 0.001$ s. PSO and SSO algorithms are used to inspect and refine the parameters of the controller (K , μ). The algorithms will take the (X_{desire}) as a reference and check each particle in (X_{actual}) by using the mean square error between them until it reaches the best fitness function with smallest mean square error, is being as:

$$Mse = \frac{1}{N} \sum_{i=1}^N (X_d - X_a)^2 \quad (40)$$

where: N : is number of random samples that were used X_d : is for desire position. X_a : is for actual position. PSO algorithm is initialized with the following parameters: $N=20$ birds, Iteration=100, $c_1=c_2=2$, $\Phi=0.5$, and the k (dimension vector) for SMC and ISMC=2 parameters.

For SSO algorithm the initialized is with the following parameters: Spider=20, Iteration=100, all the random number $[\alpha, \beta, \gamma, r_m] = 1$. The upper female percent=0.9, lower female percent=0.65. The total time for implementation is 100 Sec. Only the first three links of PUMA robot were being considered in this work.

Figure 5 shows the step response for the first, second and third links for the robot, from which it can be seen that ISMC/SSO has the best response with nearly zero maximum overshoot and stable steady state, minimum settling time, and near zero mean square error comparing with the ISMC/PSO. Figure 6 shows the final torque in the third links with ISMC/SSO, and ISMC/PSO techniques, it can be noticed that the torque response becomes more stable besides eliminating the chattering phenomena. Table 3 shows the configuration of the range for the initial values of the parameters (K , μ) for ISMC, the values after optimization with PSO, and the values after optimization with SSO.

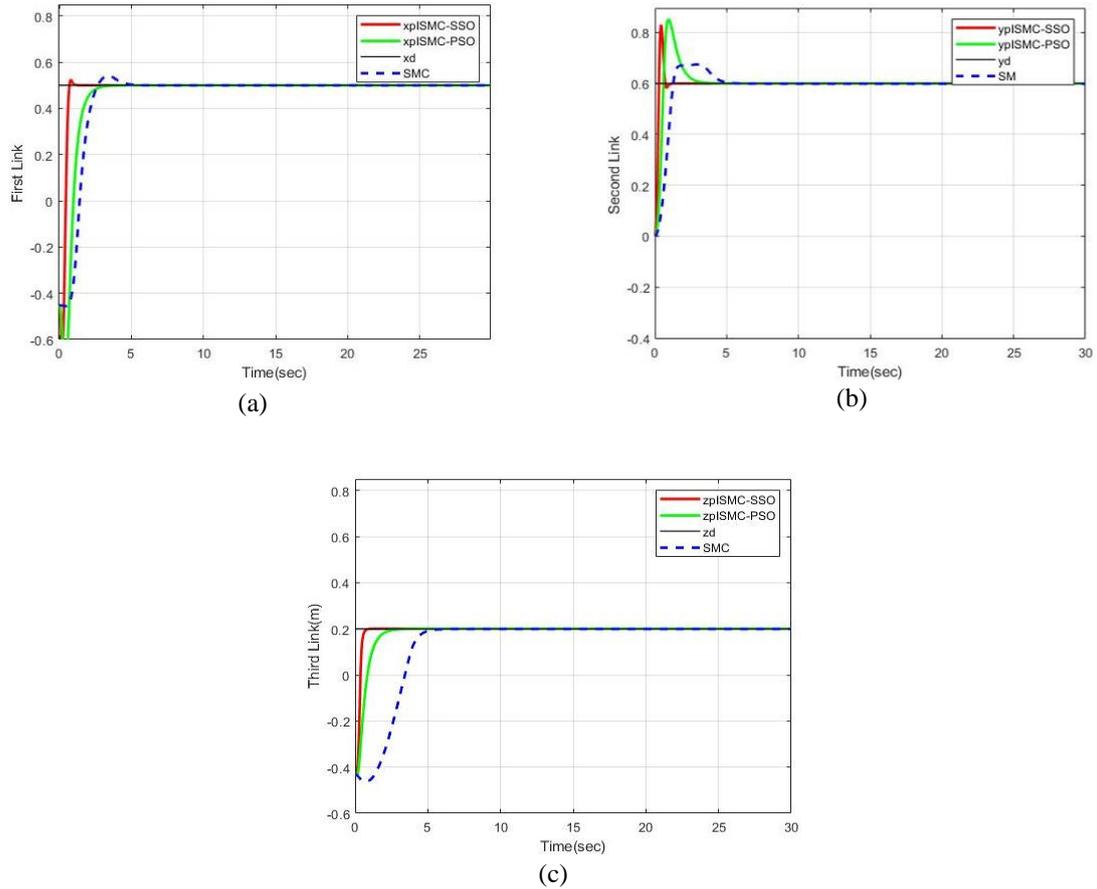


Figure 5. The step response for the first, second and third links (a, b, c) for the robot

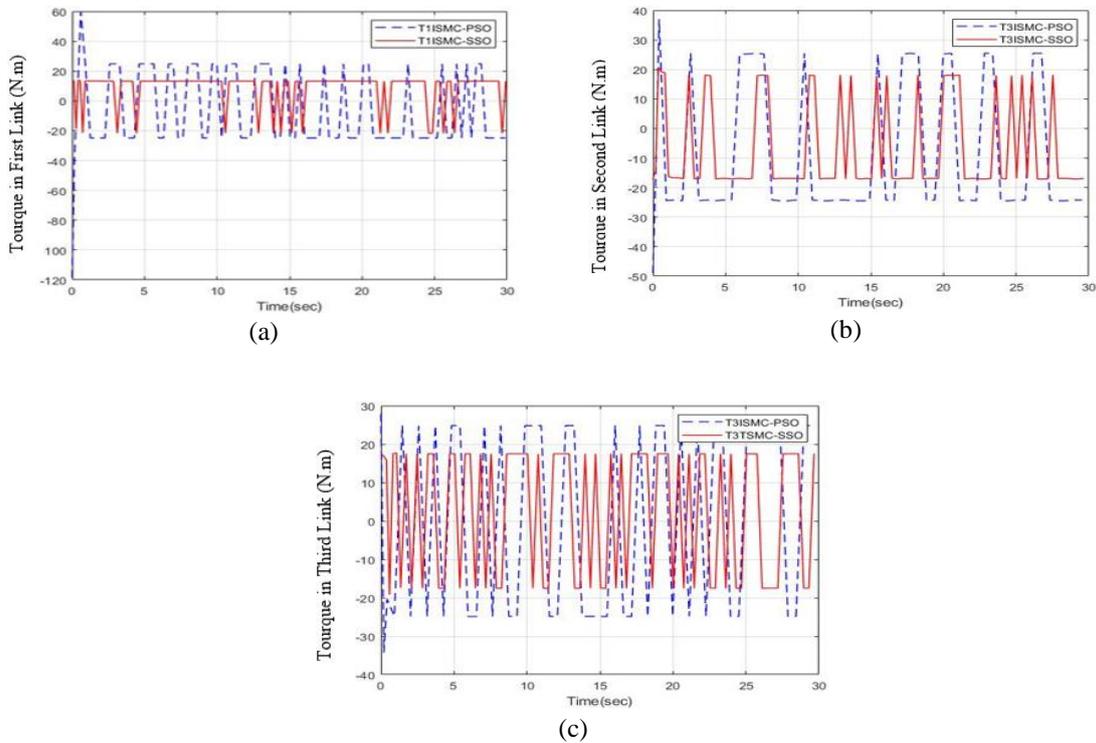


Figure 6. The final torque in the first three links (a, b, c) with ISMC/SSO, and ISMC/PSO techniques

Table 3. The configuration of the basic values for the parameters range of ISMC

The Parameters	Initial Value	Optimization with PSO	Optimization with SSO
K	30	489.2526	70.9456
μ	2	4.4726	9.3382

7.1. Robustness test

In order to test the robustness of the proposed techniques, a white Gaussian noise predefined of 40% power from the input signal was introduced to the system between 5 s to 10 s. Figure 7 shows the effect of the disturbance noise on the response signal for the three links, we can notice that a slight oscillations occur in the trajectory response of the links. Figure 8 shows the iteration of SSO and PSO algorithms with mean square error and objective function, from which it can be seen that SSO successively iterates until it reaches the optimal value for parameters, with best fitness= $4.4876 e^{-6}$. Besides PSO has best fitness= $3.4948 e^{-4}$. In order to evaluate the performance of the proposed techniques, a comparison were made with some existing methods. Table 4 shows a comparison with existing approaches specifically used to operate with robot manipulator in order to verify the validation of the proposed controller techniques in terms of robustness and stability, these techniques are shown in [11], [12], [17].

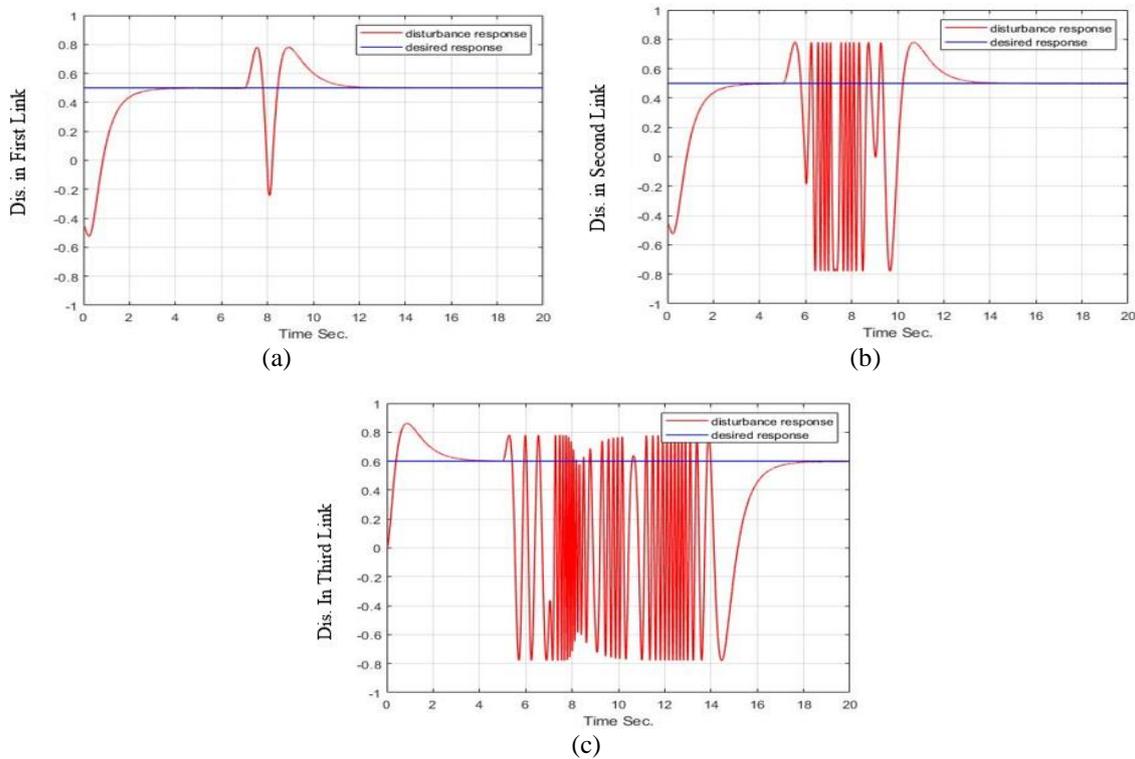


Figure 7. The effect of disturbance of a white Gaussian noise for the first three links (a, b, c)

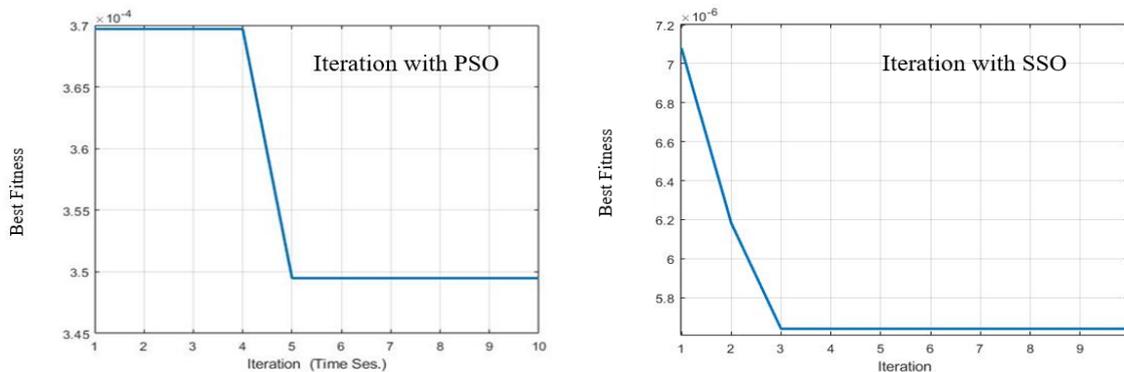


Figure 8. Iteration and objective function for SSO and PSO algorithm

Table 4. Comparison between proposed techniques and existing methods

Category	The Techniques	Mean Square Error
1	Proposed method ISMC /SSO	$4.4876 e^{-6}$
2	Proposed method ISMC /PSO	$3.4948 e^{-4}$
3	Adaptive fuzzy SMC	0.0016
4	Fuzzy SMC	0.0001
5	Estimated SMC	0.00172
6	Super-twisting of sliding mode controller STSMC	0.00085
7	Impedance controller /PSO	0.00208

7.2. Discussion

The results demonstrate significant response with almost zero maximum overshoot and stable steady state with the ISMC and SSO optimization algorithm compared with the ISMC/PSO, the oscillations are dampened faster, besides reduce the chattering phenomenon. The torque reaction becomes more stable. The comparison with the existing techniques shows that ISMC/SSO has the best fitness.

8. CONCLUSION

This work introduces an optimal technique and robust control strategy based on conventional SMC, and integral sliding mode control (ISMC) with two novel swarm intelligent algorithms, SSO and PSO, to control the first three links of PUMA robot manipulators. Therefore, the evolutionary algorithms SSO and PSO are used to determine the optimal values for the parameters of the proposed controllers such that the suggested techniques trajectory response is ensure the stability and robustness for variable conditions that affect the end effector of the robot manipulator. Thus, the developed method effectiveness was observed by simulations, where a comparison with the conventional SMC has been done. The simulation results demonstrated that the proposed ISMC/SSO offering high performance in term of step response, stability, and smoothness of control signal despite disturbance noise, with minimum mean square error and best fitness compared to the other existing techniques.

REFERENCES

- [1] M. Kuang, P. Ouyang, W. Yue, and H. Kang, "Nonlinear PD Terminal Sliding Mode Control for PUMA Robotic Manipulator," *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Xi'an, China, 2019, pp. 250-255, doi: 10.1109/ICIEA.2019.8833984.
- [2] C. S. G. Lee and M. Ziegler, "Geometric approach in solving inverse kinematics of PUMA robots," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 6, pp. 695-706, 1984, doi: 10.1109/TAES.1984.310452.
- [3] M. F. Zohra, B. Mokhtar, and M. Benyounes, "Sliding mode performance control applied to a DFIG system for a wind energy production," *International Journal Electronic Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 6139-6152, 2020, doi: 10.11591/ijece.v10i6.pp6139-6152.
- [4] H. Akkar and S. Haddad, "Design Stable Controller for PUMA 560 Robot with PID and Sliding Mode Controller Based on PSO Algorithm," *International Journal Intelligent Engineering System.*, vol. 13, no. 6, pp. 487-499, 2020, doi: 10.22266/ijies2020.1231.43.
- [5] H. Nurhadi, E. Apriliani, T. Herlambang, and D. Adzkiya, "Sliding mode control design for autonomous surface vehicle motion under the influence of environmental factor," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 4789-4797, 2020, doi: 10.11591/ijece.v10i5.pp4789-4797.
- [6] S. Yadegar and A. binti C. Soh, "Design Stable Robust Intelligent Nonlinear Controller for 6- DOF Serial Links Robot Manipulator," *International Journal of Intelligent Systems & Applications*, vol. 6, no. 8, pp. 19-38, 2014, doi: 10.5815/ijisa.2014.08.03.
- [7] A. Badar, B. S. Umre, and A. S. Junghare, "Study of Artificial Intelligence Optimization Techniques applied to Active Power Loss Minimization," *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, no. 1989, pp. 39-45, 2014.
- [8] E. Cuevas, M. A. Díaz-Cortés, and D. A. Oliva Navarro, "A swarm global optimization algorithm inspired in the behavior of the social-spider," *Advances of Evolutionary Computation: Methods and Operators*, vol. 629, no. 16, pp. 9-33, 2016, doi: 10.1007/978-3-319-28503-0_2.
- [9] J. J. Q. Yu and V. O. K. Li, "A social spider algorithm for global optimization," *Applied Soft Computing Journal*, vol. 30, pp. 614-627, 2015, doi: 10.1016/j.asoc.2015.02.014.
- [10] A. E. Serbencu, A. Serbencu, D. C. Cernega, and V. M., "Particle swarm optimization for the Sliding Mode controller parameters," *Proceedings of the 29th Chinese Control Conference*, Beijing, China, 2010, pp. 1859-1864.
- [11] H. Mehdi and O. Boubaker, "Impedance controller tuned by particle swarm optimization for robotic arms," *International Journal of Advanced Robotic System*, vol. 8, no. 5, pp. 93-103, 2011, doi: 10.5772/45692.
- [12] A. Jalali, F. Piltan, A. Gavahian, M. Jalali, and M. Adibi, "Model-Free Adaptive Fuzzy Sliding Mode Controller Optimized by Particle Swarm for Robot Manipulator," *International Journal Information Engineering Electronic Business*, vol. 5, no. 1, pp. 68-78, 2013, doi: 10.5815/ijieeb.2013.01.08.

- [13] O. Djaneye-Boundjou, X. Xu and R. Ordóñez, "Automated particle swarm optimization based PID tuning for control of robotic arm," *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)*, Dayton, OH, USA, 2016, pp. 164-169, doi: 10.1109/NAECON.2016.7856792.
- [14] C. C. Soon, R. Ghazali, H. I. Jaafar, and S. Y. S. Hussien, "Sliding Mode Controller Design with Optimized PID Sliding Surface Using Particle Swarm Algorithm," *Procedia Computer Science*, vol. 105, pp. 235-239, 2017, doi: 10.1016/j.procs.2017.01.216.
- [15] A. Zidan, S. Tappe, and T. Ortmaier, "A comparative study on the performance of MOPSO and MOCS as auto-tuning methods of PID controllers for robot manipulators," *Proceeding. 15th International Conference on Informatics Control Automation Robot (ICINCO)*, vol. 1, pp. 240-247, 2018, doi: 10.5220/0006899802400247
- [16] M. H. Barhaghtalab, V. Meigoli, M. R. G. Haghghi, S. A. Nayeri, and A. Ebrahimi, "Dynamic analysis, simulation, and control of a 6-DOF IRB-120 robot manipulator using sliding mode control and boundary layer method," *Journal of Central South University*, vol. 25, no. 9, pp. 2219-2244, 2018, doi: 10.1007/s11771-018-3909-2.
- [17] S. Tayebi-Haghghi, F. Piltan, and J. M. Kim, "Robust composite high-order super-twisting sliding mode control of robot manipulators," *Robotics*, vol. 7, no. 1, pp. 13-31, 2018, doi: 10.3390/robotics7010013.
- [18] L. Zhou, J. Zhang, H. She, and H. Jin, "Quadrotor uav flight control via a novel saturation integral backstepping controller," *Automatika*, vol. 60, no. 2, pp. 193-206, 2019, doi: 10.1080/00051144.2019.1610838.
- [19] H. Hassani, A. Mansouri, and A. Ahaitouf, "Robust autonomous flight for quadrotor UAV based on adaptive nonsingular fast terminal sliding mode control," *International Journal of Dynamics and Control*, 2020, pp. 1-17, doi: 10.1007/s40435-020-00666-3.
- [20] T. Dewi, P. Risma, Y. Oktarina, L. Prasetyani, and Z. Mulya, "The Kinematics and Dynamics Motion Analysis of a Spherical Robot," *2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Bandung, Indonesia, 2019, pp. 101-105, doi: 10.23919/EECSI48112.2019.8977011.
- [21] S. A. Mazhari, S. Kumar, "PUMA 560 Optimal Trajectory Control using Genetic Algorithm, Simulated Annealing and Generalized Pattern Search Techniques," *International Journal of Electrical Computer Engineering*, vol. 2, no. 5, pp. 830-839, 2008, doi: 10.1.1.306.7577.
- [22] B. Armstrong, O. Khatib and J. Burdick, "The explicit dynamic model and inertial parameters of the PUMA 560 arm," *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, 1986, pp. 510-518, doi: 10.1109/ROBOT.1986.1087644.
- [23] T. Mei *et al.*, "Simulation Research on Motion Trajectory of PUMA 560 Manipulator Based on MATLAB," *2019 Chinese Control and Decision Conference (CCDC)*, Nanchang, China, 2019, pp. 4857-4862, doi: 10.1109/CCDC.2019.8832476.
- [24] F. Hasan, L. Rashad, and A. Humod, "Integrating Particle Swarm Optimization and Routh-Hurwitz's Theory for Controlling Grid-Connected LCL-Filter Converter," *International Journal Intelligent Engineering and System*, vol. 13, no. 4, pp. 102-113, 2020, doi: 10.22266/ijies2020.0831.10.
- [25] M. A. El-Shorbagy and A. E. Hassanien, "Particle Swarm Optimization from Theory to Applications," *International Journal Rough Sets Data Analysis*, vol. 5, no. 2, pp. 1-24, 2018, doi: 10.4018/IJRSDA.2018040101.
- [26] A. Luque-Chang, E. Cuevas, F. Fausto, D. Zaldívar, and M. Pérez, "Social Spider Optimization Algorithm: Modifications, Applications, and Perspectives," *Mathematical Problems in Engineering*, vol. 2018, no. 2, pp. 1-29, 2018, doi: 10.1155/2018/6843923.
- [27] R. A. Ibrahim, M. A. Elaziz, D. Oliva, E. Cuevas, and S. Lu, "An opposition-based social spider optimization for feature selection," *Soft Computing*, vol. 23, no. 24, pp. 13547-13567, 2019, doi: 10.1007/s00500-019-03891-x.

BIOGRAPHIES OF AUTHORS



Suhad Haddad received her BSc in Electronic and Communication Engineering from Department of Electrical and Electronic Engineering at University of Technology. She received her Higher Diploma in Computer Engineering at University of Technology in 2000. She received her Master's degree from the Electrical and Electronics Engineering Department at the University of Technology in 2015. Ph. D student in Electronic and communication Department at the University of Technology, Baghdad, Iraq. Her research activities in ANN, image processing, electronic circuits, robotic system. Having 15 years of teaching experience.



Hanan A. R. Akkar received her Bachelor's Degree from the Electrical and Electronics Engineering Department at the University of Technology in 1988. She received her Master's degree and Ph.D. degree from the Electrical and Electronics Engineering Department at the University of Technology in 1994 and 1998, respectively. She has been Professor in the Department of Electrical Engineering at the University of Technology in the fields of ANN, FL, GA and swarm intelligent based on FPGA and electronic circuits. Currently she is Head of scientific Upgrade committee in the University of Technology.