

## Optimal integral sliding mode controller design for 2-RLFJ manipulator based on hybrid optimization algorithm

Randa Jalaa Yahya, Nizar Hadi Abbas

Department of Electrical Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq

### Article Info

#### Article history:

Received Apr 24, 2021

Revised Jul 15, 2021

Accepted Aug 3, 2021

#### Keywords:

Flexible joint robot manipulator

Grey wolf optimization

HSSGWOA

Integral sliding mode control

Salp swarm algorithm

### ABSTRACT

A newly hybrid nature-inspired algorithm called HSSGWOA is presented with the combination of the salp swarm algorithm (SSA) and grey wolf optimizer (GWO). The major idea is to combine the salp swarm algorithm's exploitation ability with the grey wolf optimizer's exploration ability to generate both variants' strength. The proposed algorithm uses to tune the parameters of the integral sliding mode controller (ISMC) that design to improve the dynamic performance of the two-link flexible joint manipulator. The efficiency and the capability of the proposed hybrid algorithm are evaluated based on the selected test functions. It is clear that when compared to other algorithms like SSA, GWO, differential evolution (DE), gravitational search algorithm (GSA), particles swarm optimization (PSO), and whale optimization algorithm (WOA). The ISMC parameters were tuned using the SSA, which was then compared to the HSSGWOA algorithm. The simulation results show the capabilities of the proposed algorithm, which gives an enhancement percentage of 57.46% compared to the standard algorithm for one of the links, and 55.86% for the other.

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



### Corresponding Author:

Randa Jalaa Yahya

Department of Electrical Engineering, College of Engineering, University of Baghdad

Al-Jadriya, Baghdad, Iraq

Email: r.zaki1802m@coeng.uobaghdad.edu.iq

## 1. INTRODUCTION

Robotic manipulators can be further categorized as rigid or flexible. The elastic properties of motor shafts, as well as harmonic drive transmission systems like gearboxes, straps, and pulleys, contribute to joint flexibility. A flexible-joint robot manipulator presents serious problems such as nonlinearity, largeness, coupling, uncertainty, and joint flexibility in modeling and control. This has led to much research into developing high-performance control approaches using state-of-the-art control theories [1]. For instance, proportional integral derivative (PID) controller [2]-[4], sliding mode control [5], fractional-order sliding mode controller [6], adaptive sliding mode control [7], fuzzy sliding mode control [8] have been dedicated to the study of flexible-joint robots. An integral sliding mode controller (ISMC) tracks a flexible joint manipulator driven by a direct current (DC) motor. It is an efficient control strategy for resolving many issues with the sliding mode control (SMC) approach, including the high-frequency chattering effect and insensitivity [9].

In this analysis, the ISMC controller parameters were tuned using the standard salp swarm algorithm (SSA) and a new hybrid nature-inspired algorithm named HSSGWOA. The SSA and the grey wolf optimizer (GWO) have been combined to create HSSGWOA GWO. The GWO is selected to be combined with SSA since it can converge to a higher-quality near-optimal solution, it is more suitable to converge in than any other common population-based method, such as genetic algorithm (GA), particles swarm optimization

(PSO), and firefly algorithm (FA), it is easily deployed, implemented, and only has a few optimization parameters.

In this work, the HSSGWOA is presented to increase the exploration and exploitation ability of the SSA to produce new solutions that lead to an increase in both speeds and search space selection compared to [10], that focus on solving the continuous search space problem. At the same time, the HSSGWOA enhances the trajectory of SSA depending on the GWO hunting mechanism features compared to [11], which is limited to the spiral aggregation chain. Simultaneously the HSSGWOA used to increase the robotic manipulator ability to overcome the trajectory problem by tuning the ISMA to present better results such as rising, settling, and maximum peak overshoot for low and high dimensional problem compared to [12]. Which present enhanced SSA for the high dimensional problem of trajectory tracking.

The HSSGWOA performance is tested by benchmark test functions classified as (unimodal, multimodal, and fixed dimension multimodal test functions). The benchmark test functions are used to obtain the average and standard deviation in order for the proposed algorithm to be demonstrated and finally compared to other optimization algorithms: SSA [13], GWO [14], differential evolution (DE) [15], gravitational search algorithm (GSA) [16], particles swarm optimization (PSO) [17], and whale optimization algorithm (WOA) [18] improved the trajectory tracking of fault.

The following are the organization of the rest of the paper: section 2 establishes the modeling for two rigid link/flexible joint manipulators. The theoretical basics for the controlling method and the standard algorithms (SSA and GWO) are presented in section 3. The hybridization of the classic SSA and GWO and pseudocode are explored fully in section 4. Results and discussion are represented in section 5. Finally, section 6 summarizes the conclusions of this research work.

## 2. MODELLING FOR TWO RIGID LINK-FLEXIBLE JOINT MANIPULATOR

In this section, the general form of dynamical equations for a two rigid link-flexible joint manipulator (2-RLFJM) is expressed, and the state-space form is presented. The model's dynamics of robot manipulator can be expressed as [19]:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \quad (1)$$

where  $\theta, \dot{\theta}, \ddot{\theta} \in \mathbb{R}^{n \times n}$  denote link position, velocity and acceleration vectors, respectively,  $M(\theta) \in \mathbb{R}^{n \times n}$  and  $C(\theta, \dot{\theta}) \in \mathbb{R}^{n \times n}$  are manipulator inertia and centrifugal/coriolis forces, respectively and  $g(\theta) \in \mathbb{R}^n$  is a vector of gravitational forces. Electric motors have a mechanical subsystem and electrical subsystem, as illustrated in the following dynamic (2) and (3) respectively [7]:

$$J\ddot{\theta}_m + B\dot{\theta}_m + r\tau = k_m I_a \quad (2)$$

$$L_m \dot{I}_a + R_m I_a + K_b \dot{\theta}_m = v(t) \quad (3)$$

where,  $\theta_m, \dot{\theta}_m, \ddot{\theta}_m \in \mathbb{R}^n$  denote Joint position, velocity and acceleration vectors, respectively,  $J \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times n}$  and  $r \in \mathbb{R}^{n \times n}$  are diagonal matrices to represent coefficients of the motor inertia, motor damping, and reduction gear, respectively.  $k_m \in \mathbb{R}^{n \times n}$  diagonal matrices for the coefficients of torque of the armature.  $L_m \in \mathbb{R}^{n \times n}$ ,  $R_m \in \mathbb{R}^{n \times n}$ , and  $K_b \in \mathbb{R}^{n \times n}$  illustrate the  $n \times n$  diagonal matrices for the coefficients of electrical inductance, armature resistance, and back-EMF constant, respectively.  $I_a \in \mathbb{R}^n$  is a vector of armature current and  $v(t) \in \mathbb{R}^n$  is the control input voltage applied to the joint actuators. Torsional springs connect rigid bodies. The joint torques  $\tau$  will in this situation be transported via n shafts:

$$\tau = K_s(r\theta_m - \theta) \quad (4)$$

where,  $k_s \in \mathbb{R}^{n \times n}$  are diagonal matrices which are lumped flexibility coefficients. Then, (1) and (2) can be written as follows to obtain the state space representation:

$$\ddot{\theta} = M(\theta)^{-1}[K_s(r\theta_m - \theta) - C(\theta, \dot{\theta})\dot{\theta} - g(\theta)] \quad (5)$$

$$\ddot{\theta}_m = J^{-1}(\tau_m - B\dot{\theta}_m - rK_s(r\theta_m - \theta)) \quad (6)$$

a canonical state-space depiction can be described as follows using the state variable vector  $x = [\theta, \dot{\theta}, \theta_m, \dot{\theta}_m]$ :

$$\begin{aligned}
 \dot{x}_1 &= X_2 \\
 \dot{x}_2 &= M(\theta)^{-1} \left[ K_s(r\theta_m - \theta) - C(\theta, \dot{\theta})\dot{\theta} - g(\theta) \right] \\
 \dot{x}_3 &= X_4 \\
 \dot{x}_4 &= J^{-1}(\tau_m - B\dot{\theta}_m - rK_s(r\theta_m - \theta))
 \end{aligned} \tag{7}$$

### 3. THEORETICAL BASICS

#### 3.1. Integral sliding mode control

ISMC's main concept is high-frequency switching gain, which aims to force the state to achieve the integral sliding surface. The states are then guided to the desired equilibrium point by the integral action in the sliding manifold [9]. It reduces chattering and improving the control system's robustness and accuracy while maintaining nominal control efficiency. The sliding manifold  $s = [s_1 \ s_2]^T$  selected for the system is [20], [21]:

$$s = \dot{e} + ce + k_i \int \dot{e} + ce \, dt \tag{8}$$

where,  $c = [c_1 \ c_2]^T$  and  $k_i$  are positive design parameters and  $e = [e_1 \ e_2]^T$  is the error value between the desired and actual angular positions ( $\theta_d$ ) and ( $\theta$ ), respectively,

$$e = \theta_d - \theta; \dot{e} = \dot{\theta}_d - \dot{\theta} \text{ and } \ddot{e} = \ddot{\theta}_d - \ddot{\theta} \tag{9}$$

where  $\theta_d = [\theta_{d1} \ \theta_{d2}]^T$  represents the desired joint positions. The time derivative of (9):

$$\dot{s} = \ddot{\theta}_d - \ddot{\theta} + c\dot{e} + k_i(\dot{e} + ce) \tag{10}$$

The control signal is composed of two terms, which are the equivalent ( $u_{eq}$ ) and the switching ( $u_{sw}$ ) control laws, represented by

$$u = u_{eq} + u_{sw} \tag{11}$$

where,  $u_{eq} = [u_{eq1} \ u_{eq2}]^T$  and  $u_{sw} = [u_{sw1} \ u_{sw2}]^T$ , the equivalent law is given by,

$$u_{eqi} = w_i \int \dot{e} + ce \, dt \tag{12}$$

where,  $w_i$  is a positive design parameter. Also, the design of the switching law is given by,

$$u_{swi} = k_i \operatorname{sgn}(s_i) \tag{13}$$

Then, the control law  $u$  is important for driving the system's initial states to equilibrium in a finite time,

$$u = k_i \operatorname{sgn}(s_i) + w_i \int \dot{e} + ce \, dt \tag{14}$$

The ISMC control that controls the flexible manipulator as shown in Figure 1 as a closed-loop block diagram.

#### 3.2. Standard salp swarm algorithm

The SSA algorithm is a recently proposed meta-heuristic algorithm proposed by Mirjalili *et al.* [22]. SSA is a stochastic algorithm in which the initial population is generated by generating a set of initial random solutions and then improving these solutions over time in two stages, exploring and exploiting, to begin the optimization process. The promising regions are discovered in the first stage by exploring the search space, and better solutions can be found in the second stage by searching the neighborhood of unique solutions. The salp chains are mathematically divided into two classes by random population (salp) division: the first salp in a set of salps is called the leader, and the remaining salps are called followers. The (15) can be used to update the leader's position:

$$x_j^1 = \begin{cases} F_j + c_1 \left( (ub_j - lb_j) * c_2 + lb_j \right) & \text{if } c_3 \leq 0.5 \\
 F_j - c_1 \left( (ub_j - lb_j) * c_2 + lb_j \right) & \text{if } c_3 > 0.5 \end{cases} \tag{15}$$

$$c_1 = 2e^{-\left(\frac{4t}{T}\right)^2} \tag{16}$$

For the updating of followers' position, the following (17) (newton movement law) are used:

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \tag{17}$$

where,  $x_j^1$  and  $x_j^i$  are the position of leader and followers in the  $j^{\text{th}}$  dimension, and  $i \geq 2$ .  $F_j$  represents the position of  $F_j$  in the  $j^{\text{th}}$  dimension. The  $c_2$  and  $c_3$  parameters are randomly allocated in the interval [0, 1]. The upper and lower bounds of search space in the  $j^{\text{th}}$  dimension are denoted by  $ub_j$  and  $lb_j$ . The current iteration is denoted by the letter t. It is assumed that the maximum number of iterations is T [23].

**3.3. Standard grey wolf optimizer**

Mirjalili *et al.* offered the GWO algorithm in 2014 that imitated grey wolves' social conduct [14]. The initial population  $n$  is randomly generated, and each search agent (solution)  $\vec{X}_i$  is evaluated by calculating its fitness function ( $f(\vec{X}_i)$ ). The overall best three solutions are assigned according to their fitness values which are alpha  $\alpha$ , beta  $\beta$ , and the delta  $\delta$  solutions  $\vec{X}_\alpha$ ,  $\vec{X}_\beta$ , and  $\vec{X}_\delta$  respectively. The main loop is iterated over and over until the final criterion is met. In the population every search agent (solution) is updated on the basis of the  $\alpha$ ,  $\beta$ , and  $\delta$  solutions location as shown in (18)-(20) [24].

$$\vec{d}_\alpha = |\vec{c}_1 \vec{X}_\alpha - \vec{X}|, \vec{d}_\beta = |\vec{c}_2 \vec{X}_\beta - \vec{X}|, \vec{d}_\delta = |\vec{c}_3 \vec{X}_\delta - \vec{X}| \tag{18}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1(\vec{d}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2(\vec{d}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3(\vec{d}_\delta) \tag{19}$$

$$\vec{X}^{i+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{20}$$

The parameter  $a$  is gradually decreased from 2 to 0, and the coefficients  $\vec{A}$  and  $\vec{C}$  are updated as shown in (21) and (22).

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{21}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{22}$$

For iterations, the components of  $\vec{a}$  are linearly reduced from 2 to 0, and  $\vec{r}_1, \vec{r}_2$  are random vectors in the range [0, 1].

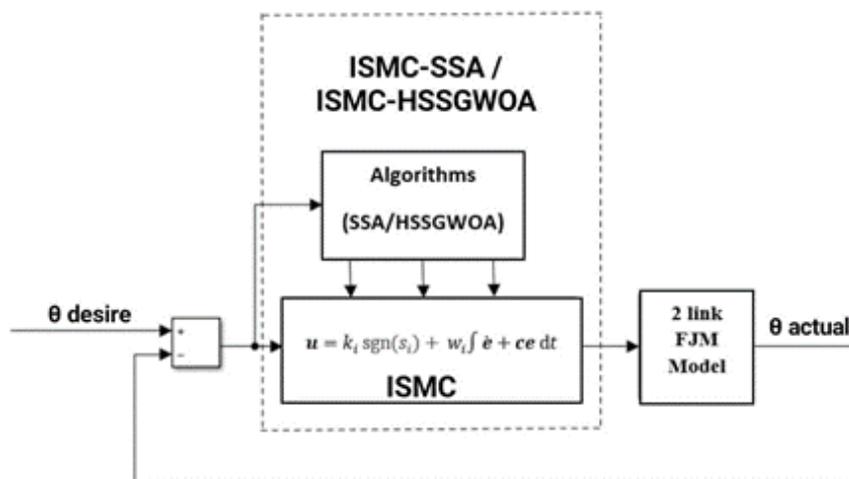


Figure 1. ISMC control of the flexible manipulator in a closed loop

#### 4. PROPOSED HYBRID ALGORITHM

Many researchers have presented several hybrid techniques to improve the balance between exploration and exploitation of the meta-heuristics. Two variants can be hybridized as heterogeneous or homogeneous at the low or high level using relay or co-evolutionary techniques, according to [25]. In the hybrid SSA with GWO (HSSGWOA), the proposed mathematical (24) are used to update the positions of the first three agents in the search space. Rather than using the standard mathematical equations. The inertia constant is used to control the grey wolf's exploration and exploitation in the search space. The major set of equations has been updated to include (23).

$$\vec{d}_\alpha = |\vec{c}_1 \vec{X}_\alpha - w * \vec{X}|, \vec{d}_\beta = |\vec{c}_2 \vec{X}_\beta - w * \vec{X}|, \vec{d}_\delta = |\vec{c}_3 \vec{X}_\delta - w * \vec{X}| \quad (23)$$

To merge SSA and GWO variants, the updated equation of leader and followers are proposed as:

$$x_j^1 = \begin{cases} \vec{X}^{i+1}(j) + c_1((ub_j - lb_j) * c_2 + lb_j) & \text{if } c_3 \leq 0.5 \\ \vec{X}^{i+1}(j) - c_1((ub_j - lb_j) * c_2 + lb_j) & \text{if } c_3 > 0.5 \end{cases} \quad (24)$$

$$x_j^i = \frac{1}{2} * w * (x_j^i + \vec{X}_j^{i+1}) \quad (25)$$

where,  $w = 0.5 + M/2$ ,  $M$ : random number between 0 and 1. (26)

This technique is also useful for fast finding the optimal global solution while ignoring the local optima in the search field. As a result, by accelerating the search, this approach will improve the search capacity and obtain accurate convergences. Pseudocode (1) is a description of the proposed algorithm which is less formal than the programming language.

##### Pseudocode 1. Pseudocode of the proposed variant (HSSGWOA)

```

Initialization
Initialize  $a$ ,  $w$ , and  $c$ 
// $w = 0.5 + rand()/2$ 
Evaluate the fitness of agents by using (23)
while ( $t < \text{max no. of iter}$ )
  for each search agent
    update the salp's position by using (24), (25)
  end for
  update  $a$ ,  $w$ , and  $c$ 
  evaluate the fitness of all search agents
  update position first three agents
   $t = t + 1$ 
end while
return // first best search agent position

```

## 5. NUMERICAL RESULTS AND DISCUSSION

### 5.1. Test functions details

In this section, twelve benchmark problems are used to assess HSSGWOA's ability. The benchmark functions are a set of five unimodal functions have used (F01-F04), while five multimodal functions have been used (F05-F08), and lastly, fixed dimension multimodal functions are used (F09-F12).

### 5.2. Proposed algorithm evaluation

In comparison to the twelve test functions chosen, the proposed algorithm was evaluated with other optimization algorithms, like SSA, GWO, De, GSA, PSO, and WOA. The mean and standard deviation were determined after each test was run 30 times for a maximum of 500 iterations. Table 1 displays the results of the benchmark functions that were chosen, while the performance of HSSGWOA versus other optimization algorithms for selected test functions is shown in Figure 2.

According to the Figure 2, there is only one local minima value for unimodal test functions, but multimodal test functions have multiple local minima values as the number of task dimensions increases [26]. However, based on the above result, it is clear that unimodal test functions (F1, F2, F3, and F4) multimodal test functions (F5, F6, F7, and F8) fixed multimodal test functions (F9, F10, F11, and F12) tend to reach towards more optimal minimum value than that SSA and GWO. At the same time, it nearly converges in fixed dimension multimodal function (F11), but it has a lower minimum amount than SSA and GWO.

Table 1. Comparison of HSSGWOA with SSA, GWO, DE, GSA, PSO, and WOA algorithms

Function		H	SSA	GWO	DE	GSA	PSO	WOA
F1 Sphere	avg	0	1.532465e-07	1.798918e-27	3.785863e-04	2.395230e-16	1.972420e-04	9.721323e-72
	std	0	1.574832e-07	2.735927e-27	1.160100e-04	9.987744e-17	2.414174e-04	5.289479e-71
F2 Rosenbrock	avg	1.563649e-257	1.944535e+00	1.002611e-16	2.265834e-03	1.526447e-01	2.791847e-02	2.050305e-51
	std	0	1.416327e+00	7.940182e-17	3.953432e-04	3.952567e-01	1.958926e-02	4.378904e-51
F3 Schwefel 1.2	avg	0	1.452889e+03	2.542808e-05	3.233881e+04	8.776302e+02	9.162589e+01	4.474669e+04
	std	0	7.522332e+02	8.622467e-05	5.113106e+03	2.957608e+02	3.336285e+01	1.567523e+04
F4 Schwefel 2.21	avg	2.138370e-248	1.153759e+01	1.000470e-06	1.319284e+01	6.536704e+00	1.125534e+00	4.096783e+01
	std	0	3.320347e+00	1.132639e-06	1.217085e+00	1.863213e+00	2.656102e-01	2.783494e+01
F5 Rastrigin	avg	0	5.711053e+01	2.996362e+00	8.815668e+01	2.968292e+01	5.430430e+01	1.894781e-15
	std	0	1.833602e+01	4.192170e+00	6.838932e+00	7.079345e+00	1.208748e+01	1.037814e-14
F6 Ackley	avg	8.881784e-16	2.529196e+00	1.020221e-13	5.797166e-03	3.104350e-02	1.874897e-01	3.730349e-15
	std	0	7.948872e-01	2.143302e-14	1.369535e-03	1.700322e-01	4.935341e-01	2.652598e-15
F7 Griewank	avg	0	1.560242e-02	4.411572e-03	5.004505e-03	2.005808e+01	8.628781e-03	2.595940e-02
	std	0	1.173097e-02	9.078760e-03	9.436476e-03	7.493497e+00	9.752355e-03	7.612745e-02
F8 Penalized 1	avg	9.666283e-06	7.345112e+00	3.993490e-02	5.648215e-05	1.945708e+00	6.919864e-03	2.944774e-02
	std	4.609522e-06	3.039908e+00	1.829342e-02	2.962006e-05	1.148009e+00	2.629960e-02	2.038037e-02
F9 Shekel's Foxholes	avg	9.980038e-01	1.097407e+00	5.368922e+00	3.256576e+00	6.632106e+00	3.460122e+00	2.244510e+00
	std	0	3.033061e-01	4.432979e+00	3.320169e+00	3.828878e+00	2.817040e+00	2.460817e+00
F10 Kowalik	avg	3.082014e-04	2.206847e-03	3.078867e-03	8.038624e-04	4.209761e-03	9.376270e-04	6.793260e-04
	std	7.476199e-07	4.942036e-03	6.897560e-03	3.882968e-04	1.965483e-03	2.543194e-04	4.260917e-04
F11 Hartman 6	avg	-3.321995e+0	-3.222635e+0	-3.281525e+0	-3.318009e+0	-3.279230e+0	3.270475e+0	-3.223428e+0
	std	1.596934e-15	5.646238e-02	6.449528e-02	2.170269e-02	7.377653e-02	5.992296e-02	1.265153e-01
F12 Shekel 5	avg	-9.900690e+0	-7.650628e+0	-9.644675e+0	-5.054906e+0	-6.918486e+0	-7.088751e+0	-8.694577e+0
	std	1.010904e+00	3.411437e+00	1.545891e+00	2.140827e-04	3.762296e+00	3.155255e+00	2.455123e+00

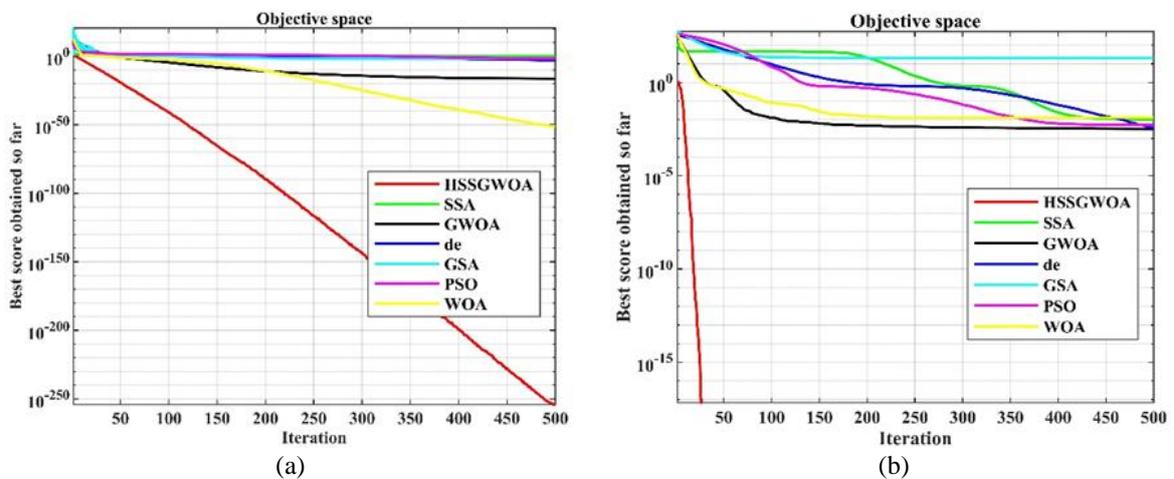


Figure 2. Algorithm's performance comparison based on the selected test functions: (a) F2 (rosen brock), (b) F7 (griewank)

**5.3. Robot manipulator model results and discussion**

The proposed algorithm for tracking control of two-link flexible joint manipulators' performance and effectiveness was validated using results from MATLAB/Simulink simulations. The dynamic equation of the system, and the parameter values, shown in Table 2, are used to evaluate the system's performance: size of population=30, max iteration=200, lower bound=-250, upper bound=250.

$$\text{objective function} = \text{error} = \theta_{desired} - \theta_{actual} \text{ (minimum)} \tag{27}$$

The gains of the controllers are optimized to improve the dynamic response of the system. The standard SSA is used to adjust the ISMC parameters and compare them to HSSGWOA results. Figure 3 illustrates the robot response to unit step by ISMC-HSSGWOA and ISMC-SSA (a: first link position, b: first joint position). Figure 4 shows the robot response to unit step by ISMC-HSSGWOA with ISMC-SSA for (a: second link position, b: second joint position).

Table 2. Parameters of manipulator [8] and DC motor [27]

Parameters	2-RLFJ Robot Manipulator		Parameters	2-RLFJ Robot Manipulator	
	Subsystem 1 Value	Subsystem 2 Value		Subsystem 1 Value	Subsystem 2 Value
B	0.1	0.1	K <sub>S</sub>	9	4
K <sub>b</sub>	0.01	0.01	J	0.01	0.01
K <sub>m</sub>	0.01	0.01	L	0.34	0.34
R <sub>m</sub>	1	1	m	1.51	0.87
L <sub>m</sub>	0.5	0.5	g	9.81	9.81

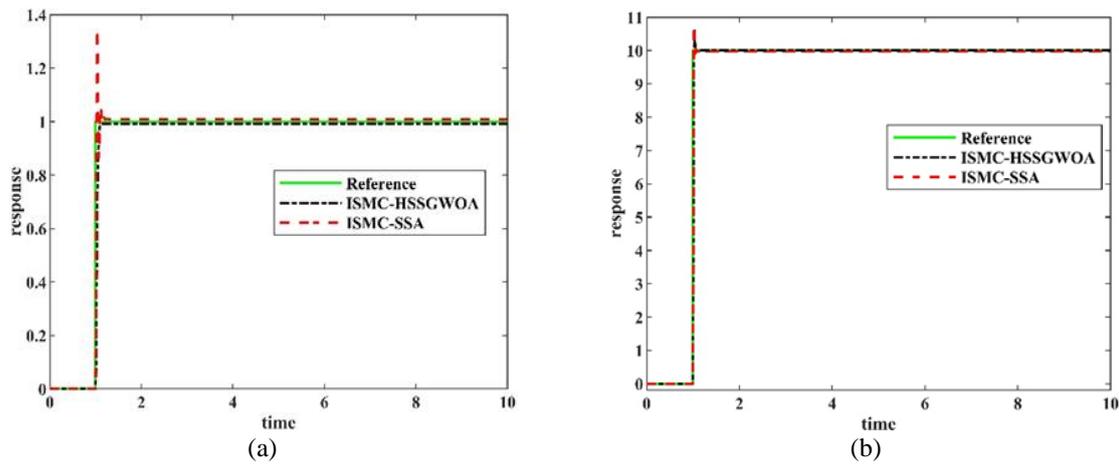


Figure 3. Position tracking results of (a) link1, (b) joint1 controlled by ISMC-HSSGWOA and ISMC-SSA

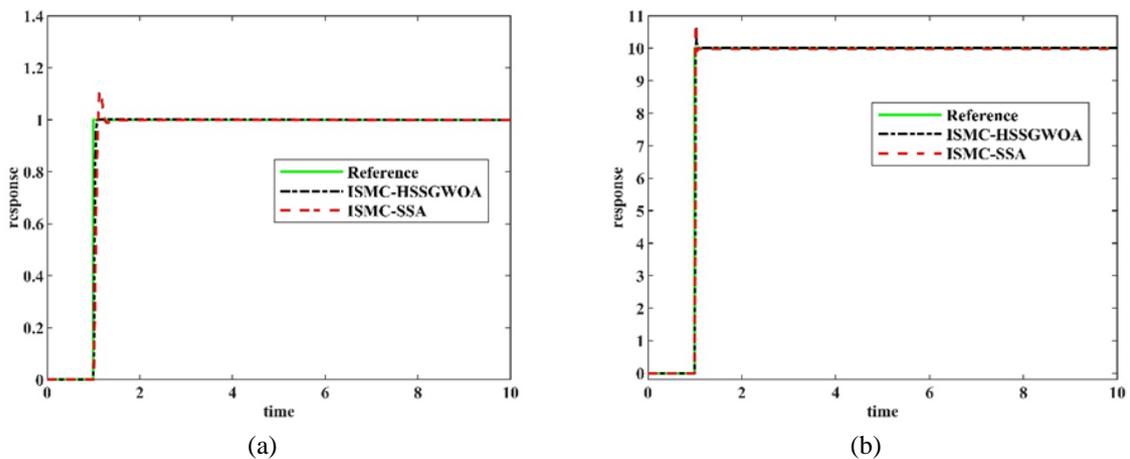


Figure 4. Position tracking results of (a) link2, (b) joint2 controlled by ISMC-HSSGWOA and ISMC-SSA

To thoroughly illustrate the proposed algorithm conduct and the SSA algorithm, Table 3 gives effective values for some of the parameters, such as overshoot and settling time. The proposed algorithm could be concluded to be more effective than SSA based on the results of this table. The results obtained in Table 3 indicate clearly that the HSSGWOA present better results for overshoot and settling time compared to the SSA which present lower performance for 1<sup>st</sup> link position (Theta1) and 1<sup>st</sup> joint position (Thetam1), the same indication observed for 2<sup>nd</sup> link position (Theta2) and 2<sup>nd</sup> joint position (Thetam2). Tables 4 and 5 demonstrate the ISMC controller parameters tuned by HSSGWOA and SSA.

Table 3. A comparison of the HSSGWOA and SSA effective parameters

Definition		HSSGWOA	SSA	Definition		HSSGWOA	SSA
Theta1	Overshoot	0.1544	33.5230	Theta2	Overshoot	0.0859	11.0307
	SettlingTime	1.0864	1.1244		SettlingTime	1.0642	1.2230
Thetam1	Overshoot	6.4566	7.8489	Thetam2	Overshoot	5.8230	5.8350
	SettlingTime	1.0397	1.0363		SettlingTime	1.0435	1.0432

Table 4. The ISMC-HSSGWOA parameters

Parameters	Value	Parameters	Value	Parameters	Value	Parameters	Value
c <sub>1</sub>	43.4995	c <sub>2</sub>	58.8710	c <sub>3</sub>	182.9172	c <sub>4</sub>	150.8086
k <sub>1</sub>	-236.7234	k <sub>2</sub>	-219.6553	k <sub>3</sub>	58.414	k <sub>4</sub>	59.7781
w <sub>1</sub>	81.2286	w <sub>2</sub>	4.4369	w <sub>3</sub>	-100.1642	w <sub>4</sub>	-89.3167
ISE1	0.0142	ISE2	0.0198	ISE3	0.5520	ISE4	0.5812
IAE1	0.0251	IAE2	0.0987	IAE3	0.0888	IAE4	0.1029
Cost-fun	0.0393	Cost-fun	0.1185	Cost-fun	0.6408	Cost-fun	0.6549

Table 5. The ISMC-SSA parameters

Parameters	Value	Parameters	Value	Parameters	Value	Parameters	Value
c <sub>1</sub>	155.3824	c <sub>2</sub>	21.7620	c <sub>3</sub>	275.8593	c <sub>4</sub>	160.005
k <sub>1</sub>	-47.7944	k <sub>2</sub>	-27.3265	k <sub>3</sub>	48.8162	k <sub>4</sub>	61.6350
w <sub>1</sub>	62.7149	w <sub>2</sub>	-4.5492	w <sub>3</sub>	292.5492	w <sub>4</sub>	-106.764
ISE1	0.0333	ISE2	0.1120	ISE3	0.5330	ISE4	0.5809
IAE1	0.0591	IAE2	0.1001	IAE3	0.3442	IAE4	0.2823
Cost-fun	0.0924	Cost-fun	0.2121	Cost-fun	0.8772	Cost-fun	0.8632

The results obtained in Tables 4 and 5 clearly show that the ISMC tuned by HSSGWOA presents a better cost function than the typical SSA. The enhancement percentage can be calculated according to (28),

$$EP]_{\text{algorithm A over algorithm B}} = \left(1 - \frac{OB]_A}{OB]_B}\right) \times 100\% \quad (28)$$

where  $EP$  described the enhancement percentage, while  $OB$  described the objective function value of each algorithm. Then, the enhancement of HSSGWOA, which is the new version of SSA over its original algorithm, is:

$$EP]_{\text{ISMC-HSSGWOA over ISMC-SSA}} = \left(1 - \frac{0.0393}{0.0924}\right) \times 100\% = 57.46\% \text{ (for 1}^{\text{st}} \text{ link position)}$$

$$EP]_{\text{ISMC-HSSGWOA over ISMC-SSA}} = \left(1 - \frac{0.1185}{0.2121}\right) \times 100\% = 55.86\% \text{ (for 2}^{\text{nd}} \text{ link position)}$$

$$EP]_{\text{ISMC-HSSGWOA over ISMC-SSA}} = \left(1 - \frac{0.6408}{0.8772}\right) \times 100\% = 26.95\% \text{ (for 1}^{\text{st}} \text{ joint position)}$$

$$EP]_{\text{ISMC-HSSGWOA over ISMC-SSA}} = \left(1 - \frac{0.6549}{0.8632}\right) \times 100\% = 24.13\% \text{ (for 2}^{\text{st}} \text{ joint position)}$$

## 6. CONCLUSION

In this paper, a newly hybrid nature-inspired algorithm called HSSGWOA tuners is proposed to determine the optimal design parameters of the integral sliding mode controller that is used for tracking control of robot manipulator. In comparison to SSA and other algorithms, experimental results showed the hybrid algorithm is more efficient in delivering high-quality solutions with rational computational iteration (unimodal, multimodal, and fixed dimensions multimodal). From the result, the responses obtained from the ISMC-HSSGWOA are better than the results collected based on ISMC-SSA as previously mentioned.

Finally, the enhancement percentages are as follows: for the first link position (57.46%), for the first joint position (26.95%), for the second link position (55.86%), and the second joint position (24.13%). In the future, the authors plan to the experimental implementation of this work on a real robot.

## REFERENCES

- [1] M. M. Fateh, "Nonlinear control of electrical flexible-joint robots," *Nonlinear Dynamics*, vol. 67, no. 4, pp. 2549-2559, 2012, doi: 10.1007/s11071-011-0167-3.
- [2] A. A. Abdul Razak, A. N. Kasruddin Nasir, M. F. Mat Jusof, S. Mohammad, and N. A. Mhd Rizal, "Opposition based spiral dynamic algorithm with an application to a PID control of a flexible manipulator," *2019 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCCE)*, 2019, pp. 126-130, doi: 10.1109/ICCSCCE47578.2019.9068556.
- [3] G. A. Sultan and M. K. Jarjes, "Optimal PID controller design using artificial bee colony algorithm for robot arm," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 21, no. 1, pp. 84-91, 2021, doi: 10.11591/ijeecs.v21.i1.pp84-91.
- [4] M. A. Ahmad, H. Ishak, A. N. K. Nasir, and N. A. Ghani, "Data-based PID control of flexible joint robot using adaptive safe experimentation dynamics algorithm," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 10, no. 1, pp. 79-85, 2021, doi: 10.11591/eei.v10i1.2472.
- [5] T. T. Nguyen, "Sliding mode control-based system for the two-link robot arm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2771-2778, 2019, doi: 10.11591/ijece.v9i4.pp2771-2778.
- [6] T. T. Nguyen, "Fractional-order sliding mode controller for the two-link robot arm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 5579-5585, 2020, doi: 10.11591/ijece.v10i6.pp5579-5585.
- [7] S. Zaare, M. R. Soltanpour, and M. Moattari, "Adaptive sliding mode control of n flexible-joint robot manipulators in the presence of structured and unstructured uncertainties," *Multibody System Dynamics*, vol. 47, no. 4, pp. 397-434, 2019, doi: 10.1007/s11044-019-09693-1.
- [8] M. R. Soltanpour, S. Zaare, M. Haghgoo, and M. Moattari, "Free-chattering fuzzy sliding mode control of robot manipulators with joints flexibility in the presence of matched and mismatched uncertainties in model dynamic and actuators," *Journal of Intelligent and Robotic Systems*, vol. 100, no. 1, pp. 47-69, 2020, doi: 10.1007/s10846-020-01178-0.
- [9] W. Alam, A. Mehmood, K. Ali, U. Javaid, S. Alharbi, and J. Iqbal, "Nonlinear control of a flexible joint robotic manipulator with experimental validation," *Journal of Mechanical Engineering*, vol. 64, no. 1, pp. 47-55, 2018, doi: 10.5545/sv-jme.2017.4786.
- [10] N. Singh and S. B. Singh, "Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance," *Journal of Applied Mathematics*, vol. 2017, 2017, Art. no. 2030489, doi: 10.1155/2017/2030489.
- [11] Z. Xiang, Y. Zhou, Q. Luo, and C. Wen, "PSSA: Polar coordinate salp swarm algorithm for curve design problems," *Neural Processing Letters*, vol. 52, no. 1, pp. 615-645, 2020, doi: 10.1007/s11063-020-10271-2.
- [12] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Enhanced salp swarm algorithm: Application to variable speed wind generators," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 82-96, 2019, doi: 10.1016/j.engappai.2019.01.011.
- [13] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163-191, 2017, doi: 10.1016/j.advengsoft.2017.07.002.
- [14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [15] V. Gupta, C. C. Chan, and P. T. Sian, "A differential evolution approach to PET image de-noising," *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 4173-4176, doi: 10.1109/IEMBS.2007.4353256.
- [16] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci. (Ny)*, vol. 179, no. 13, pp. 2232-2248, 2009, doi: 10.1016/j.ins.2009.03.004.
- [17] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: An overview," *Swarm Intell.*, vol. 1, no. 1, pp. 33-57, 2007, doi: 10.1007/s11721-007-0002-0.
- [18] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.
- [19] K. M. Lynch and F. C. Park, "Modern robotics: Mechanics, planning, and control," Cambridge, USA: Cambridge University Press, 2017.
- [20] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, "Sliding mode control and observation," in *Control Engineering*, New York, NY, USA: Birkhäuser, 2014.
- [21] A. Q. Al-Dujaili, A. Falah, A. J. Humaidi, D. A. Pereira, and I. K. Ibraheem, "Optimal super-twisting sliding mode control design of robot manipulator: Design and comparison study," *International Journal of Advanced Robotic Systems*, vol. 17, no. 6, 2020, doi: 10.1177/1729881420981524.
- [22] H. Faris, S. Mirjalili, I. Aljarah, M. Mafarja, and A. A. Heidari, "Salp swarm algorithm: Theory, literature review, and application in extreme learning machines," in *Nature-Inspired Optimizers*, 2020, ch. 11, pp. 185-199.
- [23] J. Wu, R. Nan, and L. Chen, "Improved salp swarm algorithm based on weight factor and adaptive mutation," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 31, no. 3, pp. 493-515, 2019, doi: 10.1080/0952813X.2019.1572659.
- [24] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert System with Applications*, vol. 166, 2021, Art. no. 113917, doi: 10.1016/j.eswa.2020.113917.
- [25] E-G. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of Heuristics*, vol. 8, no. 5, pp. 541-564, 2002, doi: 10.1023/A:1016540724870.
- [26] M. W. Hasan and N. H. Abbas, "Controller design for underwater robotic vehicle based on improved whale optimization algorithm," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 10, no. 2, pp. 609-618, 2021, doi: 10.11591/eei.v10i2.2288.
- [27] A. A. Jodah, "Ant colony optimization for PID controller of a DC motor." Mathworks.com 2021. [https://www.mathworks.com/matlabcentral/fileexchange/50873-ant-colony-optimization-for-pid-controller-of-a-dc-motor?s\\_tid=FX\\_rc2\\_behav](https://www.mathworks.com/matlabcentral/fileexchange/50873-ant-colony-optimization-for-pid-controller-of-a-dc-motor?s_tid=FX_rc2_behav) (accessed Apr. 05, 2021).

**BIOGRAPHIES OF AUTHORS**

**Randa Jalaa Yahya**    was born in Baghdad, 1993. She is a M.Sc. research scholar in computer and control specialization with a B.Sc. degree in Electrical Engineering from the University of Baghdad, Baghdad, Iraq in 2015. Her research interests include FJM modeling and control and swarm intelligence optimization techniques. She can be contacted at email: r.zaki1802m@coeng.uobaghdad.edu.iq.



**Nizar Hadi Abbas**    was born in Baghdad, 1975. He received his B.Sc. degree in electrical engineering from University of Baghdad, Baghdad, Iraq in 2000, a M.Sc. degree in control and computer engineering in 2002 from University of Baghdad, and a Ph.D. degree in advanced control in 2011 from Osmania University, Hyderabad, India. He is presently an assistant professor. His research interests include control theory, controllers design, fractional order systems, robot modeling and control and modern optimization techniques. He published around 35 scientific articles. He can be contacted at email: dr.nizar.hadi@coeng.uobaghdad.edu.iq.