

# Sliding Mode Control with PID Surface for Robot Manipulator Optimized by Evolutionary Algorithms



Fatiha Loucif and Sihem Kechida

**Abstract** In this study, sliding mode controller (SMC) with PID surface is designed for the trajectory tracking control of robot manipulator using antlion optimization algorithm (ALO) compared with another technique called gray wolf optimizer (GWO). The idea is to determine optimal parameters ( $K_p$ ,  $K_i$ ,  $K_d$ , and  $\lambda$ ) ensuring best performance of robot manipulator system minimizing the integral time absolute error (ITAE) criterion or the integral time square error (ISTE) criterion; the modeling and the control of the robot manipulator were realized in MATLAB environment. The simulation results prove the superiority of ALO in comparison with GWO algorithm.

**Keywords** Sliding mode control · PID sliding surface · Nonlinear control · Robot manipulator

## 1 Introduction

The use of robotic arms in industrial applications has significantly been increased. The robot motion tracking control which required high accuracy, stability, and safety is one of the challenging problems due to highly coupled and nonlinear dynamic. In the presence of model uncertainties such as dynamic parameters (e.g., inertia and payload conditions), dynamic effects (e.g., complex nonlinear frictions), and unmodeled dynamics, conventional controllers have many difficulties in treating these uncertainties. Sliding mode control (SMC) is one of the most robust approaches to overcome this problem. The most distinguished property of the sliding

---

F. Loucif (✉)  
Université 8 Mai 1945, Guelma, Algeria  
e-mail: [loucif.fatiha@univ-guelma.com](mailto:loucif.fatiha@univ-guelma.com)

S. Kechida  
Laboratoire d'Automatique et Informatique de Guelma LAIG, Guelma, Algeria  
e-mail: [kechida.sihem@univ-guelma.com](mailto:kechida.sihem@univ-guelma.com)

mode control lies in its insensitivity to dynamic uncertainties and external disturbances.

However, this approach exhibits high-frequency oscillations called chattering when the system state reaches the sliding surface, which has negative effects on the actuator control and excite the undesirable unmodeled dynamics.

Recently, sliding mode, integral sliding mode controllers (ISMC), and proportional integral sliding mode controllers (PI-SMC) were examined by many researchers as a powerful nonlinear controller in [1–7]. An adaptive sliding mode control is designed in several papers [8–12]. The proportional integral derivative sliding mode controller (PID-SMC) was designed to control the robot manipulator in several works. The investigation of fuzzy logic and neuro-fuzzy logic control to design an adaptive sliding mode control are found in [13–16].

Currently, evolutionary algorithms have appeared as an alternative design method for robotic manipulator. M. Vijay and Debaschisha design a PSO-based backstepping sliding mode controller and observer for robot manipulators [17]. This authors used PSO to tune the sliding surface parameters of SMC coupled with artificial neuro-fuzzy inference system (ANFIS) [18].

The optimization of the PID-SMC parameters using ALO is outperformed in comparison with GA and PSO algorithms by Mokeddem and Draidi to control a nonlinear system [19].

This paper presents the use of novel optimization algorithms to tune SMC with PID surface for the trajectory tracking control of robot manipulator. These algorithms are described in Sect. 3. Section 2 designates the mathematical model of robot manipulator. The principle of SMC and its application on the robot manipulator are titled in Sect. 4. The simulation results are presented in Sect. 5.

## 2 Dynamic Model of Robot Manipulator

By applying Lagrange's principle, the dynamic model of two-degree-of-freedom (2DOF) robot manipulator is given by

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \quad (1)$$

where  $q_i$ ,  $\dot{q}_i$ , and  $\ddot{q}_i$  present the link position, velocity, and acceleration vectors, respectively.  $M(q)$  is the matrix inertia given by

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

where

$$\begin{aligned}
M_{11} &= m_1 \cdot l_1^2 + m_2 \cdot (l_1^2 + 2l_1 \cdot l_2 \cdot \cos(q_2)) + l_2^2 \\
M_{12} &= M_{21} = m_2 \cdot l_2(l_2 + l_1 \cdot \cos(q_2)) \\
M_{22} &= m_2 \cdot l_2^2
\end{aligned}$$

$C(q, \dot{q})$  is the Coriolis centripetal force matrix given by

$$C(q) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

where

$$\begin{aligned}
C_{11} &= -m_2 \cdot l_1 \cdot l_2 \cdot \sin(q_2) \cdot 2\dot{q}_2 \\
C_{12} &= -m_2 \cdot l_1 \cdot l_2 \cdot \sin(q_2) \cdot \dot{q}_2 \\
C_{21} &= m_2 \cdot l_1 \cdot l_2 \cdot \sin(q_2) \cdot \dot{q}_1 \\
C_{22} &= 0
\end{aligned}$$

$$\text{The gravity vector } G(q) = [G_{11} \ G_{12}]^T$$

is given by

$$\begin{aligned}
G_{11} &= (m_1 + m_2) \cdot g \cdot l_1 \cdot \cos(q_1) + m_2 \cdot g \cdot l_2 \cdot \cos(q_1 + q_2) \\
G_{12} &= m_2 \cdot g \cdot l_2 \cdot \cos(q_1 + q_2)
\end{aligned}$$

Finally,  $F(\dot{q}) = [F_{11} F_{21}]^T$  is the friction force vector given by

$$\begin{aligned}
F_{11} &= 2 \cdot \dot{q}_1 + 0.8 \text{ sign}(\dot{q}_1) \\
F_{21} &= 4 \cdot \dot{q}_2 + 0.1 \text{ sign}(\dot{q}_2)
\end{aligned}$$

and  $\tau$  is the vector of the torque control signal, where  $m_i$  and  $l_i$  are the link mass and length, respectively.

### 3 Evolutionary Algorithms

#### 3.1 Gray Wolf Optimizer (GWO)

GWO is a recent meta-heuristic optimizer inspired by gray wolves and proposed by [20]. It mimics the leadership hierarchy and the hunting mechanism of gray wolves in nature.

As described in literature, the GWO algorithm includes two mathematical models: encircling prey and hunting prey.

**The encircling behavior:** During the hunting, the gray wolves encircle prey. The mathematical model is presented in the following equations:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (2)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\vec{D} \quad (3)$$

where  $D$  is the distance,  $\vec{X}_p(t)$  is the position vector of prey,  $\vec{X}(t)$  indicates the position of the gray wolf,  $t$  indicates the current iteration, and  $\vec{A}$  and  $\vec{C}$  are coefficient vectors calculated as follows:

$$\vec{A} = 2a\vec{r}_1 - \vec{a} \quad (4)$$

$$\vec{C} = 2r_2 \quad (5)$$

where components of  $a$  are linearly decreased from 2 to 0 over the course of iterations and  $r_1$  and  $r_2$  are random vectors in  $[0, 1]$ .

**The hunting model:** Four types of gray wolves participate in chasing prey; alpha, beta, delta, and omega denote the wolf group and are employed as solutions (fittest, best, and candidate) for simulating the leadership hierarchy.

The optimization algorithm is guided by  $\alpha$ ,  $\beta$ , and  $\delta$ ; with three best solutions obtained so far, the other search agents follow them and update their positions according to the best search agent as follows:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}(t)| \quad (6)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}(t)| \quad (7)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}(t)| \quad (8)$$

$$\vec{X}_1 = \vec{X}_\alpha(t) - \vec{A}_1 \cdot \vec{D}_\alpha \quad (9)$$

$$\vec{X}_2 = \vec{X}_\beta(t) - \vec{A}_2 \cdot \vec{D}_\beta \quad (10)$$

$$\vec{X}_3 = \vec{X}_\delta(t) - \vec{A}_3 \cdot \vec{D}_\delta \quad (11)$$

$$\text{and } \vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (12)$$

### 3.2 The Ant Lion Optimizer

Another novel nature-inspired algorithm called ant lion optimizer (ALO) mimics the hunting mechanism of antlions in nature [21]. Five main steps of hunting prey such

as the random walk of ants, building traps, entrapment of ants in traps, catching preys, and rebuilding traps are implemented in this algorithm.

Since ants move stochastically in nature when searching for food, a random walk is chosen for modeling ants' movement as follows:

$$X(t) = [0, \text{cumsum}(2r(t_1 - 1)), \text{cumsum}(2r(t_2 - 1)), \dots, \text{cumsum}(2r(t_n - 1))] \quad (13)$$

where  $\text{cumsum}$  calculates the cumulative sum,  $n$  is the maximum number of iteration,  $t$  shows the step of random walk (iteration in this study), and  $r(t)$  is a stochastic function defined as follows:

$$r(t) = \begin{cases} 1 & \text{if rand} > 0.5 \\ 0 & \text{if rand} \leq 0.5 \end{cases} \quad (14)$$

and  $\text{rand}$  is a random number generated with uniform distribution in the interval of  $[0,1]$ . The position of ants is saved and utilized during optimization in the following matrix:

$$M_{\text{Ant}} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,d} \\ A_{2,1} & A_{2,2} & \dots & A_{2,d} \\ \vdots & & & \\ A_{n,1} & A_{n,2} & \dots & A_{n,d} \end{bmatrix} \quad (15)$$

where  $A_{i,j}$  shows the value of the  $j^{\text{th}}$  variable (dimension) of  $i^{\text{th}}$  ant,  $n$  is the number of ants, and  $d$  is the number of variables. The position of an ant refers to the parameters for a particular solution. A fitness (objective) function is utilized during optimization, and the following matrix stores the fitness value of all ants:

$$M_{\text{OA}} = \begin{bmatrix} f([A_{1,1}, A_{1,2}, \dots, A_{1,d}]) \\ f([A_{2,1}, A_{2,2}, \dots, A_{2,d}]) \\ \vdots \\ f([A_{n,1}, A_{n,2}, \dots, A_{n,d}]) \end{bmatrix} \quad (16)$$

where  $f$  is the objective function. In addition to ants, we assume the antlions are also hiding somewhere in the search space. In order to save their positions and fitness values, the following matrices are utilized:

$$\begin{aligned}
 M_{\text{Antlion}} &= \begin{bmatrix} AL_{1,1} & AL_{1,2} & \dots & AL_{1,d} \\ AL_{2,1} & AL_{2,2} & \dots & AL_{2,d} \\ & & \vdots & \\ AL_{n,1} & AL_{n,2} & \dots & AL_{n,d} \end{bmatrix} \\
 M_{\text{OAL}} &= \begin{bmatrix} f([AL_{1,1}, AL_{1,2}, \dots, AL_{1,d}]) \\ f([AL_{2,1}, AL_{2,2}, \dots, AL_{2,d}]) \\ & \vdots \\ f([AL_{n,1}, AL_{n,2}, \dots, AL_{n,d}]) \end{bmatrix} \quad (17)
 \end{aligned}$$

$AL_{i,j}$  shows the  $j$ th dimension value of  $i$ th antlion,  $n$  is the number of antlions, and  $d$  is the number of variables (dimension), where  $M_{\text{OAL}}$  is the matrix for saving the fitness of each antlion.

**Random walks of ants:** Random walks are all based on the Eq. (1). Ants update their positions with random walk at every step of optimization, which is normalized using the following equation (min–max normalization) in order to keep it inside the search space:

$$X_i^t = \frac{(X_i^t - a_i) * (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad (18)$$

where  $a_i$  is the minimum of random walk,  $d_i$  is the maximum of random walk,  $c_i^t$  is the minimum, and  $d_i^t$  indicates the maximum of  $i$ th variable at  $t$ th iteration.

**Trapping in antlion's pits:** The ants walk in a hypersphere defined by the vectors  $c$  and  $d$  around a selected antlion are affected by antlions' traps. In order to mathematically model this supposition, the following equations are proposed:

$$c_i^t = \text{Antlion}_j^t + c^t \quad (19)$$

$$d_i^t = \text{Antlion}_j^t + d^t \quad (20)$$

where  $c^t$  is the minimum of all variables at  $t$ th iteration,  $d^t$  indicates the vector including the maximum of all variables at  $t$ th iteration,  $c_i^t$  is the minimum of all variables for  $i$ th ant,  $d_i^t$  is the maximum of all variables for  $i$ th ant, and  $\text{Antlion}_j^t$  shows the position of the selected  $j$ th antlion at  $t$ th iteration.

**Building trap:** In order to model the antlions's hunting capability, a roulette wheel is employed for selecting antlions based of their fitness during optimization. Ants are assumed to be trapped in only one selected antlion. This mechanism gives high chances to the fitter antlions for catching ants.

**Sliding ants towards antlion:** Once antlions realize that an ant is in the trap they shoot sands outwards the center of the pit. This behavior slides down the trapped ant that is trying to escape. For mathematically modelling this behavior, the radius of

ants's random walks hyper-sphere is decreased adaptively. The following equations are proposed in this regard:

$$c^t = \frac{c^t}{I} \quad (21)$$

$$d^t = \frac{d^t}{I} \quad (22)$$

where  $I$  is a ratio,  $c^t$  is the minimum of all variables at  $t$ th iteration, and  $d^t$  indicates the vector including the maximum of all variables at  $t$ th iteration. In Eqs. (21) and (22),  $I = 10^{w \frac{t}{T}}$  where  $T$  is the maximum number of iterations and  $w$  is a constant defined based on the current iteration. Basically, the constant  $w$  can adjust the accuracy level of exploitation.

**Catching prey and rebuilding the pit:** For mimicking the final stage of hunt this process, it is assumed that catching prey occur when ants becomes fitter (goes insides and) than its corresponding antlion, which is required to update its position to the latest position of the hunted ant. The following equation is proposed in this regard:

$$\text{Antlion}_j^t = \text{Ant}_i^t \text{ if } f(\text{Ant}_j^t) > f(\text{Antlion}_j^t) \quad (23)$$

where  $\text{Antlion}_j^t$  shows the position of selected  $j$ th antlion at  $t$ th iteration and  $\text{Ant}_i^t$  indicates the position of  $i$ th ant at  $t$ th iteration.

**Elitism:** In this algorithm the best antlion obtained so far in each iteration is saved and considered as an elite.

Since the elite is the fittest antlion, it should be able to affect the movements of all the ants during iterations. Therefore, it is assumed that every ant randomly walks around a selected antlion by the roulette wheel and the elite simultaneously as follows:

$$\text{Ant}_i^t = \frac{R_A^t + R_E^t}{2} \quad (24)$$

where  $R_A^t$  is the random walk around the antlion selected by the roulette wheel at  $t$ th iteration,  $R_E^t$  is the random walk around the elite at  $t$ th iteration, and  $\text{Ant}_i^t$  indicates the position of  $i$ th ant at  $t$ th iteration.

## 4 Sliding Mode Control

The principle of this type of control consists in bringing, whatever the initial conditions, the representative point of the evolution of the system on a hypersurface of the phase space representing a set of static relationships between the state variables.

The sliding mode control generally includes two terms:

$$U = U_{\text{eq}} + U_n \quad (25)$$

$U_{\text{eq}}$ : A continuous term, called equivalent command.  $U_n$ : a discontinuous term, called switching command.

#### 4.1 Equivalent Command

The method, proposed by Utkin, consists to admit that in sliding mode, everything happens as if the system was driven by a so-called equivalent command. The latter corresponds to the ideal sliding regime, for which not only the operating point remains on the surface but also for which the derivative of the surface function remains zero  $\dot{S}(t) = 0$  (that mean, invariant surface over time).

#### 4.2 Switching Control

The switching command requires the operating point to remain at the neighborhood of the surface. The main purpose of this command is to check the attractiveness conditions:

$$U_n = \lambda \text{sign}(S) \quad (26)$$

The gain  $\lambda$  is chosen to guarantee the stability and the rapidity and to overcome the disturbances which can act on the system. The function sign ( $S(x, t)$ ) is defined as

$$\text{Sign}(S(x, t)) = \begin{cases} 1 & \text{si } S > 0 \\ -1 & \text{si } S < 0 \end{cases} \quad (27)$$

The PID sliding surface for the sliding mode control can be indicated using the following equation:

$$S(t) = k_d \dot{e}(t) + k_p e(t) + k_i \int_0^t e(t) dt \quad (28)$$

with  $k_p$ ,  $k_i$ , and  $k_d$  mentioned as PID parameters.  $e(t) = q_d - q$ ,  $\dot{e}(t) = \dot{q}_d - \dot{q}$ .  $q_d$  and  $q$  are the desired and actual position of the robot articulations.  $\dot{q}_d$  and  $\dot{q}$  are the desired and actual speed of the robot articulations.



According to Eq. (26)

$$U_n = \lambda \text{sign} \left( k_p e(t) + k_i \int e(t) dt + k_d \dot{e}(t) \right)$$

To calculate  $U_{\text{eq}}$ , it is necessary that  $\dot{S}(t) = 0$

$$\begin{aligned} k_p \dot{e}(t) + k_i e(t) + k_d \ddot{e}(t) &= 0. \\ \ddot{e}(t) &= k_d^{-1} (k_p \dot{e}(t) + k_i e(t)) \end{aligned}$$

with

$$\begin{aligned} \ddot{e}(t) &= \ddot{q}_d - \ddot{q} \\ \ddot{q} &= M(q)^{-1} (U_{\text{eq}} - H(q, \dot{q})\dot{q} - G(q) - F(\dot{q})) = \ddot{q}_d + k_d^{-1} (k_p \dot{e}(t) + k_i e(t)) \\ U_{\text{eq}} &= M(q) (\ddot{q}_d + k_d^{-1} (k_p \dot{e}(t) + k_i e(t))) + H(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \end{aligned}$$

Finally, the PID-SMC torque presented as in [18], with the demonstration of the Lyapunov stability condition, becomes

$$\begin{aligned} U &= M(q) (\ddot{q}_d + k_d^{-1} (k_p \dot{e}(t) + k_i e(t))) + H(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \\ &+ M(q) k_d^{-1} \lambda \text{sign} \left( k_p e(t) + k_i \int_0^{t_f} e(t) dt + k_d \dot{e}(t) \right) \end{aligned} \quad (29)$$

## 5 Simulation and Results

The main goal of this work is to optimize the parameters of SMC with PID surface for the trajectory control of 2DOF robot manipulator by the minimization of ITAE and ISTE objective functions mentioned as

$$J_1 = \text{ITAE} = \int_{t_1}^{t_f} |e(t)| dt \quad (30)$$

$$J_2 = \text{ISTE} = \int_{t_1}^{t_f} e(t)^2 dt \quad (31)$$

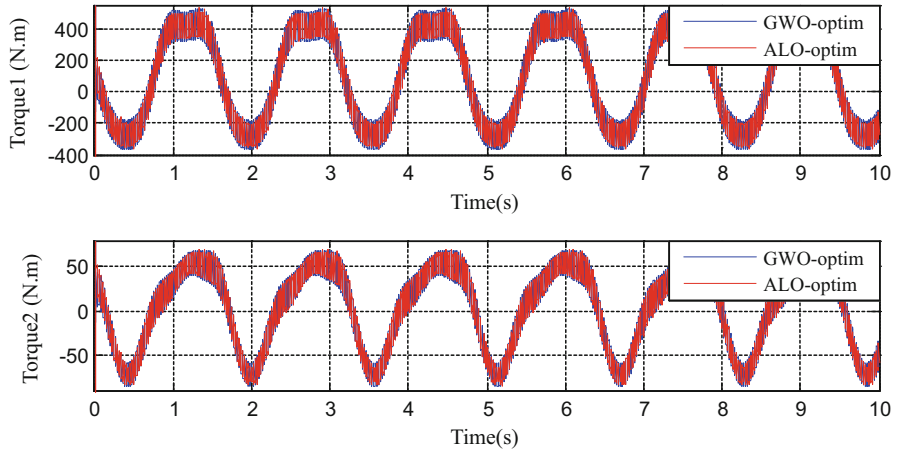
The parameters of the robot that have been taken in application are  $m_1 = 10$  kg,  $m_2 = 5$  kg,  $l_1 = 1$  m,  $l_2 = 0.5$  m, and the gravity  $g = 9.8$  m/s<sup>2</sup>. First, we apply the

**Table 1** Cost function for the first and second articulation without disturbance

Cost functions	ITAE		ISTE	
	First	Second	First	Second
SMC_PID_ALO	$9.669 \times 10^{-5}$	$3.896 \times 10^{-5}$	$5.519 \times 10^{-11}$	$3.951 \times 10^{-7}$
SMC_PID_GWO	$82.2 \times 10^{-5}$	$300.9 \times 10^{-5}$	$2.095 \times 10^{-8}$	$8.118 \times 10^{-8}$

**Table 2** SMCPID parameters for the first and second articulation

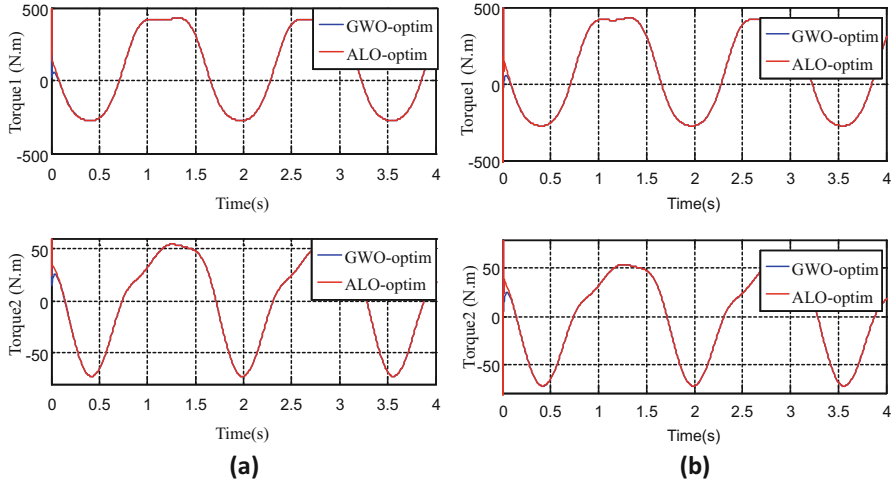
ISTE	$K_{p1}$	$K_{i1}$	$K_{d1}$	$\lambda_1$	$K_{p2}$	$K_{i2}$	$K_{d2}$	$\lambda_2$
SMC_PID_ALO	500	45.22	1.18	58.5	500	16.5	17.135	34.83
SMC_PID_GWO	500	49.89	15.75	43.01	603	37	46	48
ITAE	$K_{p1}$	$K_{i1}$	$K_{d1}$	$\lambda_1$	$K_{p2}$	$K_{i2}$	$K_{d2}$	$\lambda_2$
SMC_PID_ALO	500	10.97	1	2.86	500	7.33	1	1
SMC_PID_GWO	500	65.91	12.72	59.8	507	63.79	79	40

**Fig. 1** Control torque of link 1 and link 2 with (sign) function ITAE criteria

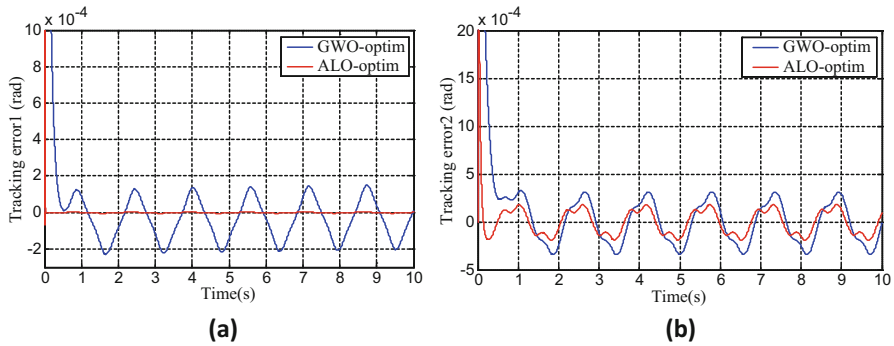
algorithms described above to tune SMC controller. The objective function values for different optimization algorithms obtained with ITAE and ISTE criteria, defined in Eqs. (30) and (31), respectively, have been shown in Table 1.

It can be seen from the Table 1 that ALO algorithm gives minimum to the objective function compared with those of GWO, which means that ALO algorithm gives the best optimum that has minimum objective function better than GWO algorithm. The corresponding optimum parameters of PIDSMC were recapitulated in Table 2.

Figure 1 shows the control input applied to the first and second articulations obtained so far by both optimization algorithms. In order to avoid the chattering



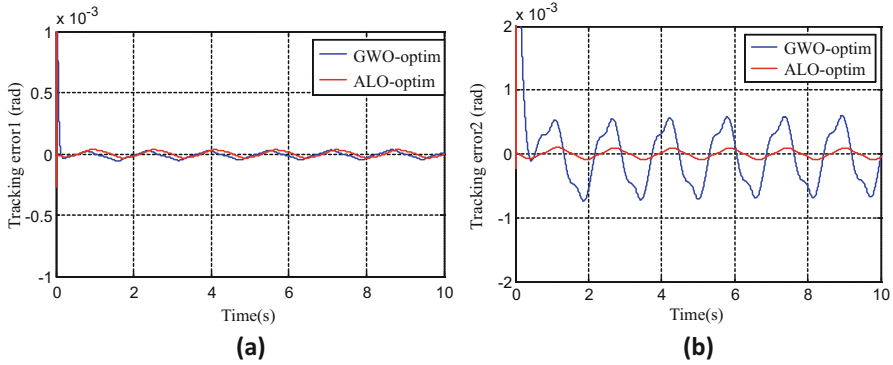
**Fig. 2** Control torque signal of SMCPID controller with  $(\tanh)$  function: (a) ISTE criteria and (b) ITAE criteria



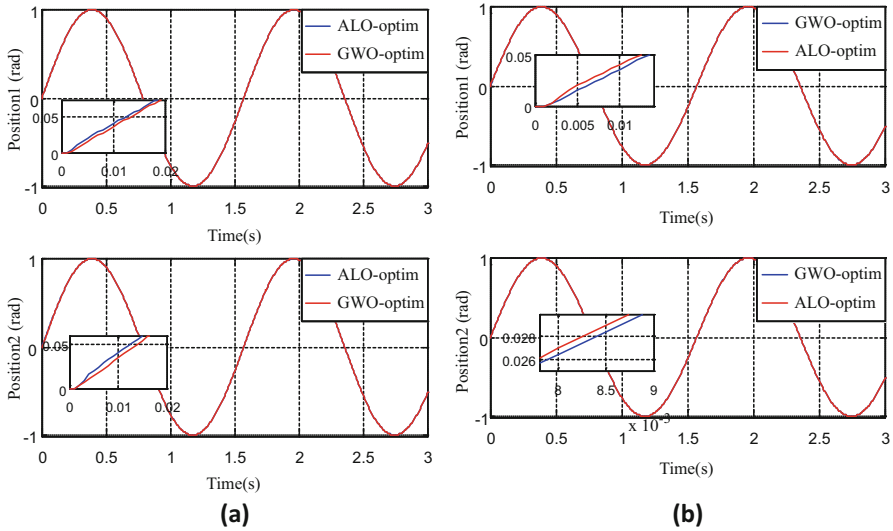
**Fig. 3** Tracking error of both articulations with ISTE criteria and  $(\tanh)$  function

effect of the “sign” function, used in Eq. (29), the latter is replaced by the “ $\tanh$ ” (hyperbolic tangent) function. It can be seen from Fig. 2 that the resulting torque was almost close for both chosen criteria and the two optimization algorithms.

Figures 3 and 4 show the error of the robot manipulator to track the desired trajectory by the minimization of ISTE and ITAE criteria, respectively. We can see from the figures that ALO algorithm, which has smaller cost function, outperforms GWO algorithm even if we change the objective function. The convergence curve of the used functions was represented in Fig. 5. The corresponding position of robot manipulator controlled by SMCPID controller optimized by the two algorithms was shown in Fig. 6.



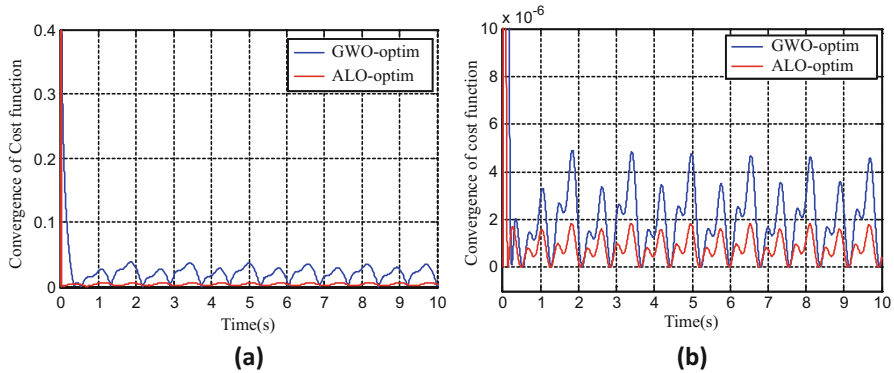
**Fig. 4** Tracking error of both articulations with ITAE criteria and  $(\tan/h)$  function



**Fig. 5** Convergence curve of cost function: (a) ITAE criteria and (b) ISTE criteria

## 6 Conclusion

In this paper the optimization of the SMC with PID surface was realized with new techniques of optimization called ALO and GWO algorithms; the ALO presents more robustness in trajectory tracking control of 2DOF robot manipulator, regarding the convergence curve of the cost function, even if we change the objective function. From the observations of the simulations, we can realize the benefits of using evolutionary algorithms to tune the controller parameters than the traditional methods, especially when the system is highly nonlinear or in presence of disturbances where an online optimization is recommended.



**Fig. 6** Position of two articulations controlled by SMCPID controller with  $(\tanh)$  function: (a) ITAE criteria and (b) ISTE criteria

## References

- Adhikary, N., & Mahanta, C. (2018). Sliding mode control of position commanded robot manipulators. *Control Engineering Practice*, 81, 183–198.
- Jung, S. (2018). Improvement of tracking control of a sliding mode controller for robot manipulators by a neural network. *International Journal of Control, Automation and Systems*, 16(2), 937–943.
- Yoo, D. (2009). A comparison of sliding mode and integral sliding mode controls for robot manipulators. *Transactions of the Korean Institute of Electrical Engineers*, 58(1), 168–172.
- Liu, R., & Li, S. (2014). Optimal integral sliding mode control scheme based on pseudo spectral method for robotic manipulators. *International Journal of Control*, 87(6), 1131–1140.
- Zhao, Y., Sheng, Y., & Liu, X. (2014). A novel finite time sliding mode control for robotic manipulators. *IFAC Proceedings*, 47, 7336–7341.
- Azlan, N. Z., & Osman, J. H. S. (2006). Proportional integral sliding mode control of hydraulic robot manipulators with chattering elimination. In: *First International Conference on Industrial and Information Systems*, Sri Lanka.
- Das, M., & Mahanta, C. (2014). Optimal second order sliding mode control for nonlinear uncertain systems. *ISA Transactions*, 53, 1191–1198.
- Adelhedi, F., Jribi, A., Bouteraa, Y., & Derbe, N. (2015). Adaptive sliding mode control design of a SCARA robot manipulator system under parametric variations. *Journal of Engineering Science and Technology Review*, 8(5), 117–123.
- Jiang, S., Zhao, J., Xie, F., Fu, J., Wang, X., & Li, Z. (2008). A novel adaptive sliding mode control for manipulator with external disturbance. In: *37th Chinese Control Conference (CCC)*.
- Yi, S., & Zhai, J. (2019). Adaptive second-order fast nonsingular terminal sliding mode control for robotic manipulators. *ISA Transactions*, 90, 41–51.
- Jing, C., Xu, H., & Niu, X. (2019). Adaptive sliding mode disturbance rejection control with prescribed performance for robotic manipulators. *ISA Transactions*, 91, 41–51.
- Mirshekaran, M., Piltan, F., Esmaeili, Z., Khajepour, T., & Kazeminasab, M. (2013). Design sliding mode modified fuzzy linear controller with application to flexible robot manipulator. *International Journal of Modern Education and Computer Science*, 10, 53–63.
- Tran, M.D., & Kang, H.J. (2015). Adaptive fuzzy PID sliding mode controller of uncertain robotic manipulator. In: *International Conference on Intelligent Computing: Intelligent Computing Theories and Methodologies, ICIC*, pp. 92–103.

14. Ataei, M., & Shafiei, S. E. (2008). Sliding mode PID-controller design for robot manipulators by using fuzzy tuning approach. In: 27th Chinese Control Conference.
15. Nejatbakhsh Esfahani, H., & Azimirad, V. (2013). A new fuzzy sliding mode controller with PID sliding surface for underwater manipulators. *International Journal of Mechatronics, Electrical and Computer Technology*, 3(9), 224–249.
16. Kharabian, B., Bolandi, H., Ehyaei, A. F., Mousavi Mashhadi, S. K., & Smailzadeh, S. M. (2017). Adaptive tuning of sliding mode-PID control in free floating space manipulator by sliding cloud theory. *American Journal of Mechanical and Industrial Engineering*, 2(2), 64–71.
17. Vijay, M., & Jena, D. (2018). PSO based backstepping sliding mode controller and observer for robot manipulators. In: International Conference on Power, Instrumentation, Control and Computing (PICC).
18. Vijay, M., & Jena, D. (2017). PSO based neuro fuzzy sliding mode control for a robot manipulator. *Journal of Electrical Systems and Information Technology*, 4(1), 243–256.
19. Mokeddem, D., & Draïdi, H. (2018). Optimization of PID sliding surface using Antlion optimizer. International Symposium on Modelling and Implementation of Complex Systems MISC, pp. 133–145.
20. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
21. Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, 83, 80–98.