



Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator

Fatiha Loucif¹ · Sihem Kechida¹ · Abdennour Sebbagh¹

Received: 10 November 2018 / Accepted: 11 November 2019 / Published online: 22 November 2019
© The Brazilian Society of Mechanical Sciences and Engineering 2019

Abstract

In this work, we are interested to the PID control of nonlinear systems and more specially the control of a robot manipulator. The idea is to determine the optimal parameters (K_p , K_i and K_d) of the controller using a novel algorithm of optimization called whale optimizer algorithm (WOA). To study the effectiveness of WOA-PID controller, its performance is compared with other controllers such as particle swarm optimization-PID (PSO-PID) and grey wolf optimizer-PID (GWO-PID). The model of robot manipulator and all controllers were tested using Simulink/MATLAB. Simulation results obtained clearly indicate the superiority of WOA-PID controller over the other controllers for trajectory tracking, better settling time, and ITAE errors.

Keywords Evolutionary algorithm · PID control · Optimization algorithm · Robot manipulator

1 Introduction

Robot manipulator is multi-input and multi-output (MIMO), highly nonlinear and coupled system, designed to automatically perform tasks emulating or reproducing, in a specific area, human actions. Typical applications include welding, painting, assembly, etc. Industrial robotics must inspect the products quickly and accurately what is required to improve its performance by inserting efficient controllers. Therefore, designing an efficient controller for this system is a challenging task [1]. Despite the success of modern control theory, robot manipulator controllers still commonly employ classical proportional–derivative (PD) or proportional–integral–derivative (PID) algorithm [2–5]. To improve their performances, most of those controllers have

been designed using linear or linearized models. In [6, 7], applications of nonlinear PID control are proposed either by adding a nonlinear proportional and derivative term to PID controller or by gravity compensating robot manipulators. To eliminate the oscillation angle of the end-effectors of the single-link flexible joint robot manipulator, the position trajectory is performed by state feedback control method [8]. A generalized PID-type control scheme with simple tuning for the global regulation of robot was also presented with constrained inputs [9]. Regarding stability and/or control approaches such as predictive control, state feedback control, decentralized control, the readers may be referred to these works [10–17]. In adaptive context, the authors of [18] presented an adaptive PID control with bounded disturbance while those of [19] have designed the controller in respect of the nonlinear uncertainties of the system. The adaptation law is motivated from the sliding mode control and derived to tune the PID gains in order to minimize the sliding conditions. Afterward, other applications based on adaptive neural networks were introduced to control robot manipulator [20]. Recently, some researchers have made an effort to develop a new control strategies based on artificial intelligent. An approach based on partitioning of the space into segments describes to learning feedback control of robot manipulator where the dynamic behavior of the robot is considered as decoupled linear system controlled by conventional PID controllers. The segments are represented as fuzzy sets

Technical Editor: Victor Juliano De Negri, D.Eng.

✉ Fatiha Loucif
loucif.fatiha@univ-guelma.dz
Sihem Kechida
kechida.sihem@univ-guelma.dz
Abdennour Sebbagh
sebbagh.abdennour@univ-guelma.dz

¹ Laboratoire d'Automatique et Informatique de Guelma (LAIG), Université 8 Mai 1945, BP 401, 24000 Guelma, Algeria

ensuring continuity of control variables [21]. Fuzzy logic self-tuning PID controller is introduced in [22–24], and the controller parameters are varied and computed online. In the hybrid context, learning automata is used to adjust the parameters of fuzzy-PID controller for optimal tracking of robot [25]. In order to minimize steady-state error with respect to uncertainties in robot control, PID control needs a big integral gain where a neural compensator is added to the classical PD control with a large derivative gain [26].

An elegant way of enhancing the performance of PID controllers is to find the appropriate settling of parameters values (K_p , K_i and K_d) to achieve optimal performance of the robot. In the recent years, several works have been carried out in optimization approaches. In the literature, some works have proposed methods combining genetic algorithm (GA) and fuzzy logic to tune PID gains for the robot control with unknown internal behavior where in some applications, GA is used as a main gain estimator associated with fuzzy logic as a ranking basement for GA [27], and in others, fuzzy logic is used as a main gain estimator associated with a signal analyzer to extract, from controller error signal, some performance indexes, i.e., overshoot, rise time, and steady-state error [28]. An other application of GA is to find optimal input and output membership functions of a fuzzy self-tuning PID controller [29]. In order to control two-degree-of-freedom (2-DOF) robot manipulator, Elkhateeb and Badr in [30] applied the bee colony algorithm to tune the PID controller. A modified version fruit fly optimization algorithm (MFOA) is also used to find the optimal PID parameters where nonlinear robot’s dynamic is linearized and decoupled using a nonlinear feedback linearization control technique [31]. In the same context, a new modified biogeography-based optimization (BBO) algorithm is developed to tune PID parameters of a five bar robot [32]. Other interesting works have emerged thanks to PSO algorithm [33–39] such as the application of

PSO algorithm in control chain of SCARA robot. In other hand, we noticed an increased attention to investigate the application of GWO such as the optimization of robot path planning with a multi-objective GWO approach [40].

In this work, a novel technique of optimization called whale optimization algorithm (WOA) inspired by Mirjalili in 2016 [41] is applied to tune PID controller for the trajectory tracking of 2-DOF robot manipulator. In order to test the effectiveness of this technique, the WOA-PID controller was compared to those obtained from PSO and GWO algorithms. The rest of this paper is organized in seven sections. The second section exhibits a brief outline of PID control followed by a review of the evolutionary algorithms used in this work (section 3). The next section is devoted to the 2-DOF robot dynamic’s. The fifth part discusses simulation results for all developed controllers. The control of robot manipulator and robustness test appear, respectively, in sections 6 and 7. The last section concludes the paper and suggests some directions for the future works.

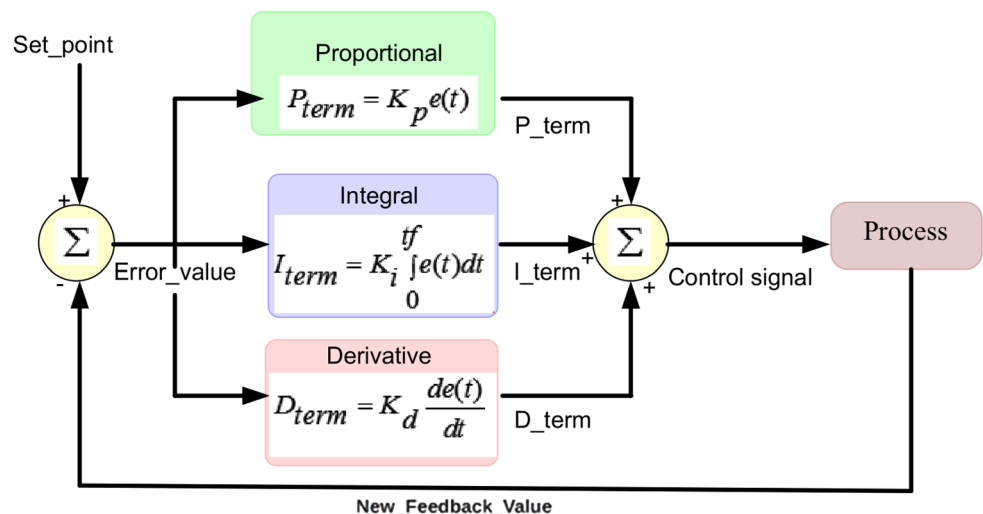
2 Design of PID controller

A PID controller is essentially a generic closed-loop feedback mechanism and monitors the error between measured system output and the desired set point. From this error, a control signal is computed to adjust the process performance. The differential equation of PID controller is:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt} \tag{1}$$

where K_p , K_i and K_d are the proportional, integral, and derivative gains, respectively. The superposition of these three actions constitutes the mechanism for adjustment of process performance as shown in Fig. 1. The continuous transfer

Fig. 1 PID control mechanism



function of the PID controller is obtained through Laplace transform as:

$$C_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \tag{2}$$

3 Evolutionary algorithms

Evolutionary algorithms are exciting new probabilistic research tools inspired by biological models that have great power to solve problems in a variety of domains. Because they ideally do not make any assumption about the underlying fitness landscape. In this section, we are interested to a class of optimization techniques that are whales optimizer algorithm, grey wolf optimizer, and particle swarm optimizer.

3.1 Whale optimizer algorithm (WOA)

The WOA is a new meta-heuristic optimization algorithm mimicking the hunting behavior of humpback whales [41]. The particularity of this approach is in the manner of the simulated hunting behavior which is made randomly, i.e., the best search agent to chase the prey and the use of spiral to simulate bubble-net attacking mechanism of humpback whales. The philosophy of hunting can be described in three steps:

1. *Encircling prey* Once the humpback whales recognize the location of prey, they encircle them. Otherwise, when the position of the optimal design in the search space is not known a priori, the WOA algorithm assumes that the current best candidate solution is the target prey or is close to the optimum that will represent the best search agent. The other agents will hence try to update their positions in the neighborhood of this agent [41]. This behavior, defined by distance \vec{D} and updated position \vec{X} , is represented by the following equations:

$$\vec{D} = |\vec{C}\vec{X}^*(t) - \vec{X}(t)| \tag{3}$$

$$\vec{X}(t + 1) = \vec{X}^*(t) - \vec{A}\vec{D} \tag{4}$$

where t indicates the current iteration; \vec{X}^* is the position vector of the best solution obtained so far, \vec{X}^* is the position vector; $|\cdot|$ is the absolute value. It is worth mentioning here that \vec{X}^* should be updated each iteration if there is a better solution. \vec{A} and \vec{C} are coefficient vectors calculated as follows:

$$\vec{A} = 2\vec{a}\vec{r}_1 - \vec{a} \tag{5}$$

$$\vec{C} = 2\vec{r}_2 \tag{6}$$

where \vec{a} is linearly decreased from 2 to 0 over the course of iterations (in both exploration and exploitation phase) and \vec{r}_1 and \vec{r}_2 are random vectors in [0, 1].

2. *Bubble-net attacking method* This step represents the exploitation phase designed by two approaches:

- Shrinking encircling mechanism: the behavior is achieved by decreasing the value of \vec{a} in (5) from 2 to 0. Consequently, the values of \vec{A} are fluctuated in the interval $[-a, a]$.
- Spiral updating position: the movement of humpback is helix-shaped that can be describe by a spiral equation created between the position of whale and prey:

$$\vec{X}(t + 1) = \vec{D}' \exp^{bl} \cos(2\pi l) + \vec{X}^*(t) \tag{7}$$

with $\vec{D}' = \vec{X}^*(t) - \vec{X}(t)$ indicates the distance of the *i*th whales to the prey (best solution obtained so far), *b* is constant for defining the shape of the logarithmic spiral, *l* is a random number in $[-1, 1]$. Note that humpback whales swim around the prey within a shrinking circle and along a spiral-shaped path simultaneously with a probability of 50% according to the following mathematical model:

$$\vec{X}(t + 1) = \begin{cases} \vec{X}^*(t) - \vec{A}\vec{D} & \text{if } p < 0.5 \\ \vec{D}' \exp^{bl} \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \tag{8}$$

where *p* is a random number in [0, 1].

3. *Exploration phase* The last step illustrates the chase of prey. This approach depends also on the variation of the vector \vec{A} . The whales search randomly according to the position of each other. Indeed, the search agent position is updated according to randomly chosen agent. Note that for the random value of $(|\vec{A}|) > 1$, the search agent moves far away from reference whale. So, this allows to perform a global research and can be modeled as follows.

$$\vec{D} = |\vec{C}\vec{X}_{rand}(t) - \vec{X}(t)| \tag{9}$$

$$\vec{X}(t + 1) = \vec{X}_{rand}(t) - \vec{A}\vec{D} \tag{10}$$

Exploration and exploitation phases are the common features of WOA and GWO which will be presented afterward.

3.2 Grey wolf optimizer (GWO)

GWO is a recent meta-heuristic optimizer inspired by grey wolves and proposed by [42]. It mimics the leadership

hierarchy and the hunting mechanism of grey wolves in nature. As described in the literature, the GWO algorithm includes two mathematical models: encircling prey and hunting prey. The encircling behavior looks like above case and is modeled by:

$$\vec{D} = |\vec{C}\vec{X}_p(t) - \vec{X}(t)| \tag{11}$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A}\vec{D} \tag{12}$$

where \vec{X}_p is the position vector of prey, $\vec{X}(t)$ indicates the position of grey wolf. \vec{A} and \vec{C} have the same expression as (5) and (6), respectively. t indicates the current iteration.

In the hunting model, four types of grey 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 α , β and γ , three best solutions obtained so far, and the other search agents follow them and update their positions according to the best search agent. The proposed formula are the following:

$$\begin{aligned} \vec{D}_\alpha &= |\vec{C}_1\vec{X}_\alpha(t) - \vec{X}(t)| \\ \vec{D}_\beta &= |\vec{C}_2\vec{X}_\beta(t) - \vec{X}(t)| \\ \vec{D}_\gamma &= |\vec{C}_3\vec{X}_\gamma(t) - \vec{X}(t)| \\ \vec{X}_1(t + 1) &= \vec{X}_\alpha(t) - \vec{A}_1\vec{D}_\alpha \\ \vec{X}_2(t + 1) &= \vec{X}_\beta(t) - \vec{A}_2\vec{D}_\beta \\ \vec{X}_3(t + 1) &= \vec{X}_\gamma(t) - \vec{A}_3\vec{D}_\gamma \\ \vec{X}(t + 1) &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \end{aligned} \tag{13}$$

3.3 Partial Swarm Optimization (PSO)

Particle swarm optimization is a stochastic algorithm inspired from natural biotic life of swarms. Based on a population of candidate solutions, its principle is to optimize a problem in search space by iteratively trying to improve a candidate

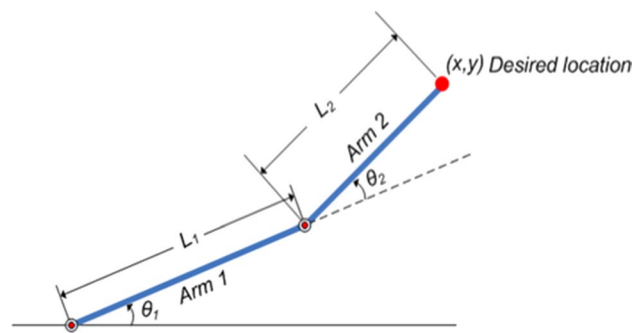


Fig. 2 Model of 2-DOF robot manipulator

solution with regard to a given objective function. Each particle searches for better positions by updating its velocity and position according to simple mathematical formulae [43]. This is expected to move the swarm toward the best solutions using the individual best position (pbest) and global best position (gbest) expressed by:

$$\begin{aligned} V_i &= \lambda(w \cdot V_i(t - 1) + c_1 \text{rand}()(\text{pbest}_i - X_i(t - 1)) \\ &\quad + c_2 \text{rand}() \cdot (\text{gbest}_i - X_i(t - 1))) \\ X_i &= X_i(t - 1) + V_i(t) \end{aligned} \tag{14}$$

where c_1 and c_2 are two positive constants, called cognitive learning rate and social rate, respectively, w is the inertia factor and λ is constriction factor.

4 Dynamic model of 2-DOF robot manipulator

In this paper, a 2-DOF manipulator is regarded as the case study of coupled nonlinear systems and presented in Fig. 2. The manipulator dynamic equation is as follows:

$$\tau = M(q)\ddot{q} + H(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \tag{15}$$

where q_i , \dot{q}_i , and \ddot{q}_i denote the link position, velocity, and acceleration vectors, respectively, $M(q)$ is the matrix inertia; $H(q, \dot{q})$ is the Coriolis centripetal forces matrix, $G(q)$ is the gravity vector, $F(\dot{q})$ is the friction force vector and τ is the vector of the torque control signal.

$$\begin{aligned} M(q) &= \begin{bmatrix} m_1 l_1^2 + m_2(l_1^2 + 2l_1 l_2 \cos(q_2) + l_2^2) & m_2 l_2(l_2 + l_1 \cos(q_2)) \\ m_2 l_2(l_2 + l_1 \cos(q_2)) & m_2 l_2^2 \end{bmatrix} & F(\dot{q}) &= \begin{bmatrix} 2\dot{q}_1 + 0.8 \text{sign}(\dot{q}_1) \\ 4\dot{q}_2 + 0.1 \text{sign}(\dot{q}_2) \end{bmatrix} \\ H(q, \dot{q}) &= \begin{bmatrix} -m_2 l_2 l_1 \sin(q_2) 2\dot{q}_2 & -m_2 l_2 l_1 \sin(q_2) \dot{q}_2 \\ m_2 l_2 l_1 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \\ G(q) &= \begin{bmatrix} (m_1 + m_2)g \cdot l_1 \cos(q_1) + m_2 \cdot g \cdot l_2 \cdot \cos(q_1 + q_2) \\ m_2 \cdot g \cdot l_2 \cdot \cos(q_1 + q_2) \end{bmatrix} \end{aligned}$$

where m_i, l_i and g are the link mass, the link length and the gravity, respectively. In application, $m_1 = 10$ kg, $m_2 = 5$ kg, $l_1 = 1$ m and $l_2 = 0.5$ m.

5 Simulation results

In this section, a PID controller is designed for the trajectory tracking control of robot manipulator using optimization techniques previously described. The idea is to determinate optimal parameters (K_p, K_i and K_d) ensuring best performance indexes (rise time, overshoot, settling time and peak) of system. The main is to apply the novel optimization algorithm (WOA) to tune the PID parameters and to demonstrate thus its efficiency compared to those obtained from the two most known algorithms (GWO and PSO). From randomly initialized parameters, the optimization algorithm minimizes the integral time absolute error criterion (ITAE), mentioned in

Equation (16), for each iteration until obtaining optimal set of K_p, K_i and K_d parameters.

First, we tune the PID controller with three algorithms for different values of agents or population size is 10, 20, 30, 50, 100 and 200; we choose thereafter the best population size for each algorithm.

$$J = ITAE = \int_0^{t_f} e(t)dt \tag{16}$$

$e = q_{ref} - q$ is the error between the reference position and the actual position of the robot. In this study, simulation results are performed using MATLAB/Simulink environment. Figure 3 illustrates the Simulink diagram of the control scheme strategy of two joints robot manipulator. The simulation consists of robot manipulator system, two controllers, trajectory that must followed, and optimization

Fig. 3 PID control mechanism

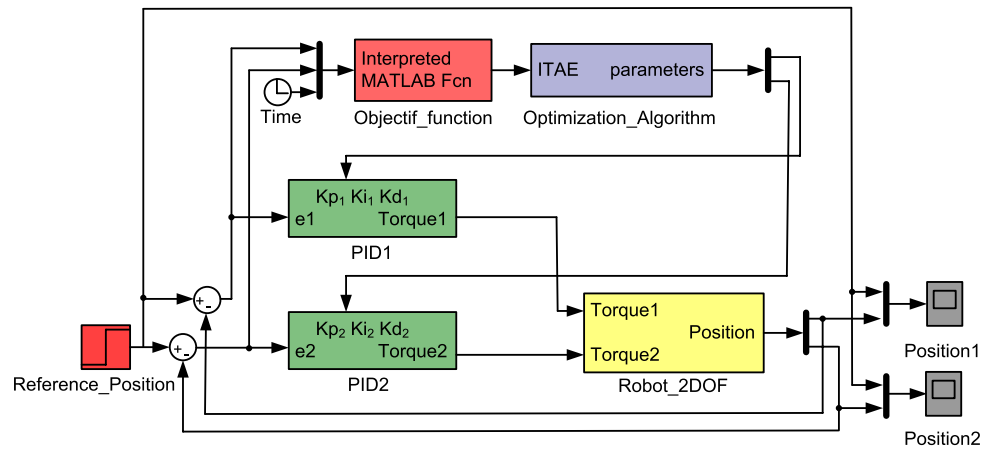


Table 1 Comparison results of the first and second joint response characteristics controlled by PSO-PID

	Nbre agents	Rise time	Settling time	Overshoot	Peak	ITAE
Joint ₁	10	0.2847	1.6044	2.0626	10.261	5.519
	20	0.3706	0.6150	1.8813	10.264	6.694
	30	0.2845	0.4513	1.8375	10.242	5.487
	50	0.379	0.650	0.951	10.18	6.034
	100	0.135	0.445	6.408	10.68	2.958
	200	0.186	1.296	2.595	10.29	3.834
Joint ₂	10	0.428	3.379	2.911	10.34	6.487
	20	0.3150	2.9138	3.2203	10.344	3.813
	30	0.4307	3.545	3.0994	10.36	6.542
	50	0.5014	4.3387	3.8711	10.443	7.86
	100	0.395	3.430	3.177	10.36	6.002
	200	0.3138	2.8678	3.229	10.344	4.106
Joints 1 and 2						
Nbre agents	10	20	30	50	100	200
Objective function	12.006	10.925	12.029	13.894	8.96	7.94

The bold signified the best values

block used for selecting the controlling parameters which based on optimal tracking error constrain. It also has two outputs which are the first joint displacement (position 1) and the second joint displacement (position 2). The desired and actual positions for the first and second joint of robot

manipulator controlled are depicted in Figs. 4, 5 and 6 with the initial conditions $q_1(0) = 0.5$ deg, $q_2(0) = 1$ deg, $\dot{q}_1(0) = 0$ and $\dot{q}_2(0) = 0$. In addition, all controllers were tested under the same conditions with time range of simulation is 10 s, and the reference signal magnitude is given as

Table 2 Comparison results of the first and second joint response characteristics controlled by GWO-PID

	Nbre agents	Rise time	Settling time	Overshoot	Peak	ITAE
Joint ₁	10	0.080	0.468	15.11	11.54	2.243
	20	0.15	0.453	7.8466	10.828	3.157
	30	0.0863	0.3841	13.0525	11.3238	2.097
	50	0.175	0.4795	3.4604	10.3912	3.461
	100	0.090	0.4086	8.7769	10.8973	2.295
	200	0.1936	0.4288	2.1872	10.2687	3.113
Joint ₂	10	0.471	2.191	1.9771	10.277	7.197
	20	0.343	0.585	0.4101	10.09	3.307
	30	0.387	0.631	1.512	10.21	5.579
	50	0.449	3.236	2.658	10.32	6.932
	100	0.444	2.277	2.038	10.27	6.784
	200	0.4293	4.1512	4.5348	10.4862	6.426
Joints 1 and 2						
Nbre agents	10	20	30	50	100	200
Objective function	9.44	6.464	7.676	10.396	9.079	9.539

The bold signified the best values

Table 3 Comparison results of the first and second joint response characteristics controlled by WOA-PID

	Nbre agents	Rise time	Settling time	Overshoot	Peak	ITAE
Joint ₁	10	0.101	0.5104	20.784	12.117	2.033
	20	0.25	0.38	0.434	10.10	3.813
	30	0.1896	0.2728	1.3703	10.19	3.668
	50	0.1224	0.446	17.61	11.78	2.541
	100	0.1824	0.4402	3.058	10.35	3.669
	200	0.079	0.488	19.99	12.03	2.165
Joint ₂	10	0.336	0.577	0.392	10.08	3.207
	20	0.476	3.442	2.774	10.34	7.325
	30	0.342	0.5841	0.4066	10.089	3.301
	50	0.415	0.727	0.498	10.11	4.176
	100	0.364	0.591	1.302	10.18	4.928
	200	0.509	0.789	1.371	10.22	7.111
Joints 1 and 2						
Nbre agents	10	20	30	50	100	200
Objective function	5.24	11.138	6.97	6.717	8.597	9.272

The bold signified the best values

Table 4 PSO-PID, GWO-PID, and WOA-PID parameters

Algorithm/PID parameters	K_p		K_i		K_d	
	1st Joint	2nd Joint	1st Joint	2nd Joint	1st Joint	2nd Joint
PSO	4000	3000	900	900	462.87	450
GWO	4000	3000	639.45	210	346.79	457.03
WOA	4000	3000	350.27	210	453.69	450

Table 5 Convergence time of three algorithms

	PSO	GWO	WOA
Convergence time/Nbre-agents	09min53s/200	2.1117s/20	1.363s/30
	53.392s/20	1.882s/20	0.928s/20

10 degree. Tables 1, 2, and 3 illustrate the temporal characteristics responses of the two joints with different number of agents.

From Table 1, we see that the lower objective function (7.94) is obtained with 200 agents for the PSO optimization

that corresponds at the smallest rise time for both joints and the others characteristics are on average acceptable. We can observe also (Table 2), that the objective function for the GWO optimization is weak with a value of 6.464 for only 20 agents. Besides the overshoot value of the first joint, GWO offers enhancement over related PSO in terms of the settling times for the first joint (0.453/1.296) and the second joint (0.585/2.8678) and gives improvements in the cost function by decreasing the value from 3.834 to 3.157 for the first joint and from 4.106 to 3.307 for the second joint.

While in Table 3, the WOA algorithm has the smallest cost function 5.24, with only 10 agents. But due to the

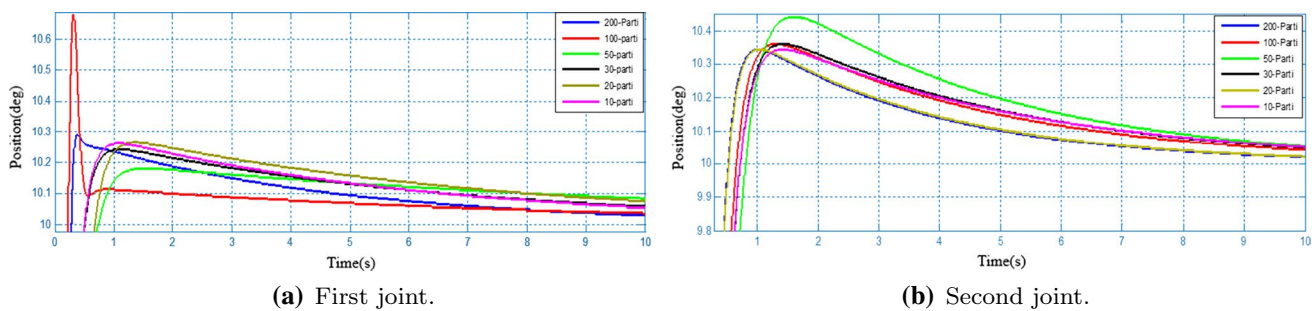


Fig. 4 Actual position for the robot controlled by PSO-PID controller

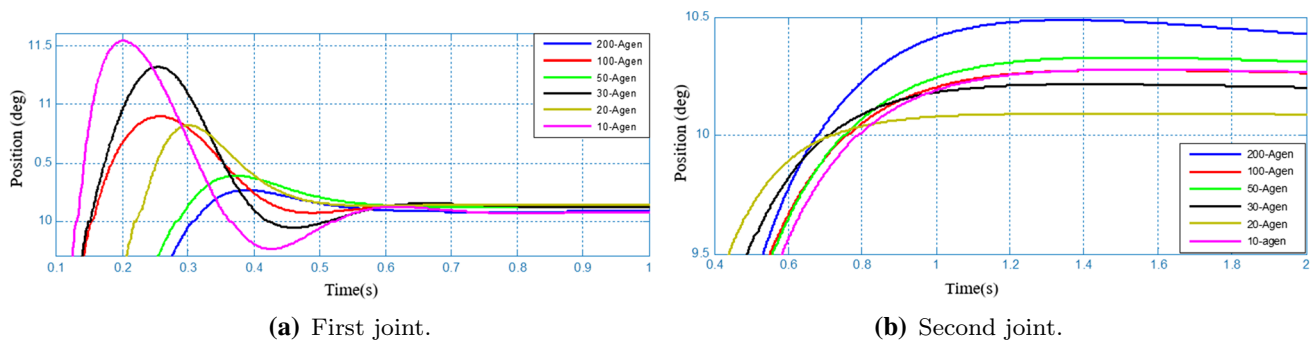


Fig. 5 Actual position for the robot controlled by GWO-PID controller

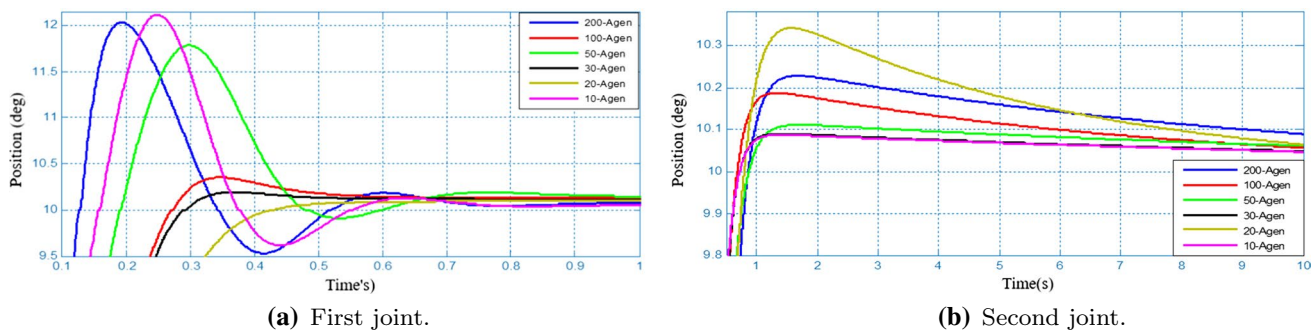


Fig. 6 Actual position for the robot controlled by WOA-PID controller

remarkable overshoot for the first joint, it seems that the results obtained with 30 agents are the best, with a small cost function 6.97, compared with the other values. In general, the WOA presents best performances for the controlled system. The different values of controller parameters deduced from the optimization process with objective function (16) are listed in Table 4. In order to prove the efficiency of WOA algorithm in term of computing time, we have tested the three algorithms for different and same number of agents; the numerical results show that the WOA algorithm spends a short time to compute the set of PID optimal parameters compared to GWO and PSO algorithms, i.e., it converges faster to the optimum. This constitutes another choice

criterion of the WOA algorithm (Table 5) that shows the superiority of WOA algorithm over the other algorithms.

From Figs. 4, 5 and 6, we can see that the first articulation has an oscillatory form for the three optimization techniques, and different population sizes, with a remarkable relative overshoot for the GWO-PID and WOA-PID between [2.18–15.11%] and [0.43–20.78%], respectively, where the maximum overshoot does not precede 6.4% for the PID control optimized by PSO algorithm. For the second articulation, the response is almost aperiodic with a small overshoot of 3.87%, 4.53% and 2.774% maximum for the PSO-PID, GWO-PID, and WOA-PID commands, respectively. We see also that the response of the first joint converges more rapidly toward the final value for the PID control optimized

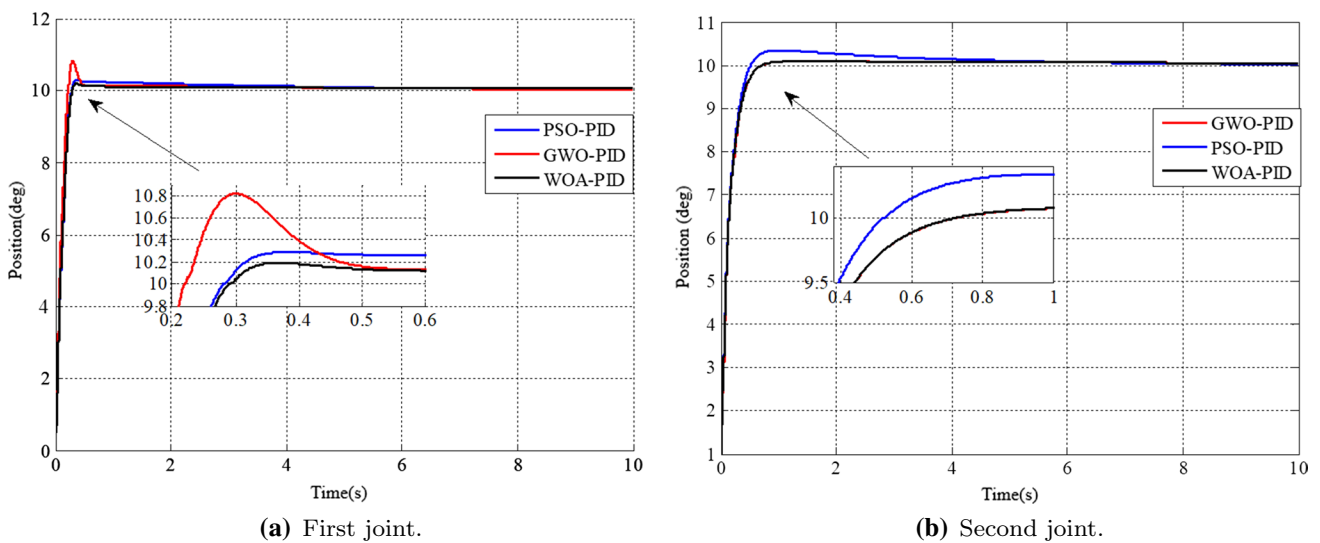


Fig. 7 Actual position for joints controlled by three algorithm controllers

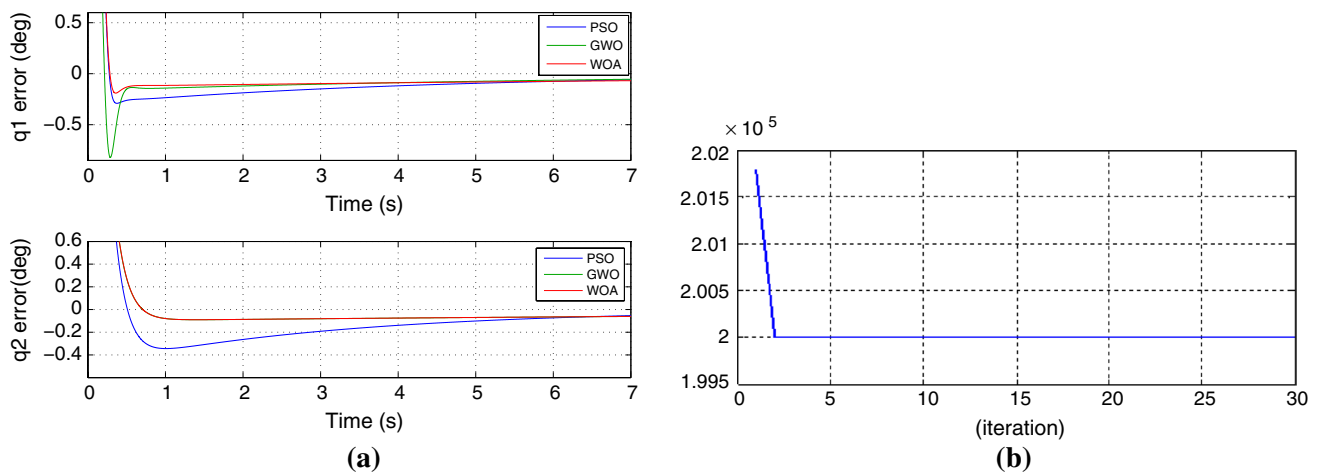


Fig. 8 a Error signal for the first and second articulations from step consign, b converge curve of the WOA algorithm

by GWO and WOA algorithms with an average response time between [0.38–0.48 s] and [0.27–0.51 s], respectively, and also with an interval of [0.58–4.15 s] and [0.57–3.44 s] for the second joint, and the WOA optimization gives good values in this sense. On the contrary, the response of the robot controlled by a PSO-PID takes more response time between [0.44–1.6 s] for the first articulation and an interval of [2.86–4.33 s] for the second articulation.

6 Control of Robot manipulator

To better see the contribution of WOA for a fast and precise trajectory tracking of the robot manipulator. We have applied the PID control tuned by the three algorithms, used the best performance of the controllers obtained by 200 agents for the PSO, 20 agents for GWO and 30 agents

for WOA, respectively. Figure 7 shows the superiority of the WOA-PID and GWO-PID controllers over the PSO-PID controller. The second joint have almost the same behavior with WOA-PID and GWO-PID, and we can see the enhancement in the performance of the first joint of the robot. The WOA algorithm offers the optimization of the settling time for the first joint compared with those obtained with GWO tuning (0.2728 s/0.453 s) and reduce the overshoot from (7.8466% to 1.3703%). It can be said that the system performance is satisfied in terms of settling time and the system responses are faster with the WOA-PID controller. Figure 8a shows the convergence speed of error resulting from WOA algorithm which confirm the competitive of this methodology. As can be seen from Fig. 8b, WOA algorithm converge to the optimum with a smaller number of iterations.

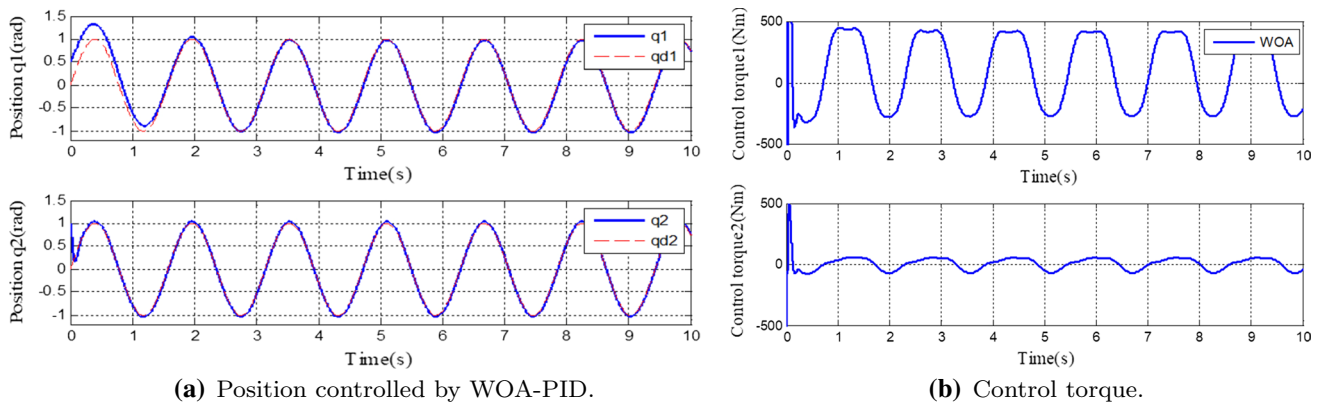


Fig. 9 Position and control torque of the first and second articulations without disturbances

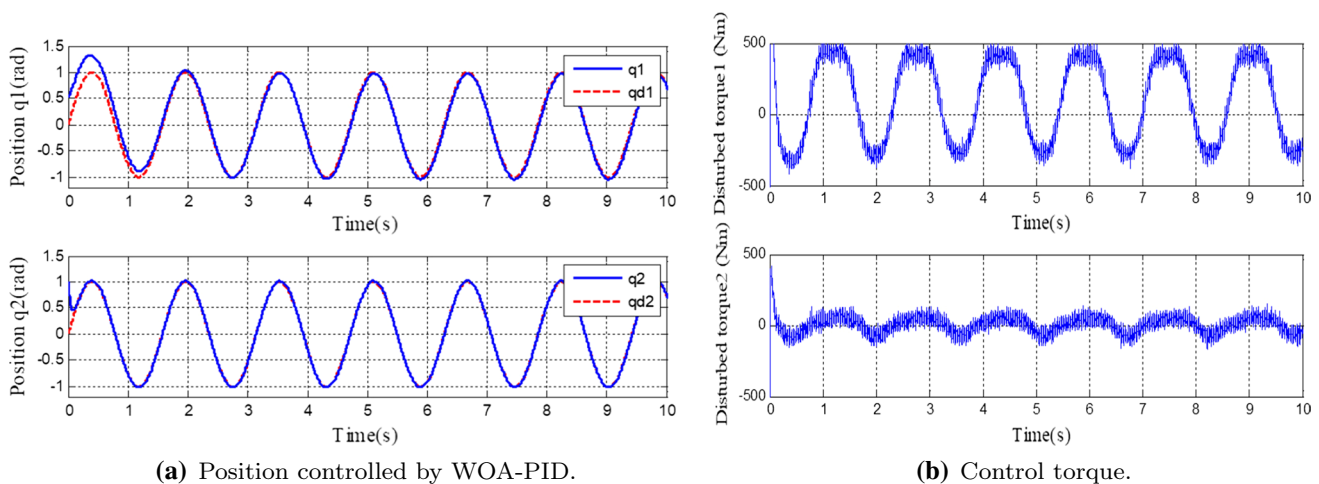


Fig. 10 Position and control torque of both articulations with disturbance (variance = 0.8)

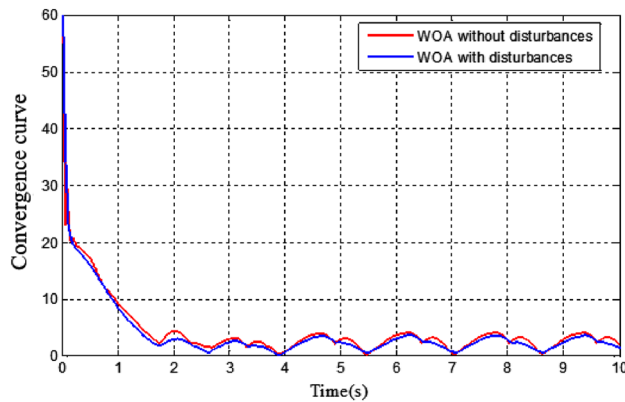


Fig. 11 Cost function of the WOA algorithm

7 Robustness test

To examine the robustness of the optimization algorithm, the obtained controller was tested for the first time by changing the input to the sinus signal and for the second time by introducing a torque disturbance for two articulations.

- (A) Tracking sinus input The optimization was done for a sinus input signal, the obtained gains of the PID controller allows pursuit of the trajectory as it is shown in Fig. 9a and the generated control torque in Fig. 9b.
- (B) Disturbance torque The objective is to test the robustness of the WOA-PID controller when the robot is subjected to disturbances. A white noise torque is added to the PID control signal with variance of (0.8) as depicted in Fig. 10a. It can be seen from Fig. 10b that the efficiency of the controlled robot by the tuned PID to track the desired trajectory despite the existence of external disturbances. Figure 11 reveals the cost function (ITAE) for both cases.

8 Conclusion

The aim of this paper is to introduce new application of the whale optimizer algorithm to adjust the PID parameters for the tracking control of the 2-DOF robot manipulator. The results obtained witness the effectiveness of the proposed WOA-PID in terms settling time, errors, and convergence time as well as its robustness to tune parameters PID controller for the robot tracking control with or without disturbances.

As future work, we propose to change the objective function and to test this controller for robots with more degrees of freedom. We plan also to compare the results of this

study with others control techniques. From an optimization standpoint, we attempt to test the WOA algorithm in others applications such as determining optimal durations of traffic lights.

Acknowledgements Funding was provided by universit  8 Mai 1945.

References

1. Kumar JS, Amuth EK (2014) Control and tracking of robot manipulator using PID controller and hardware in loop simulation. In: International conference on communication and network technologies (ICCNT), pp 1–3
2. Su Y, Muller PC, Zheng C (2010) Global asymptotic saturated PID control for robot manipulator. *IEEE Trans Control Syst Technol* 8(6):1280–1288
3. Abdelhedi F, Bouteraa Y, Chemori A, Derbel N (2014) Nonlinear PID and feedforward control of robotic manipulators. In: 15th International conference on sciences and techniques of automatic control and computer engineering (STA), pp 349–354
4. Bascetta L, Rocco P (2007) Revising the robust control design for rigid robot manipulator. In: International conference on robotics and automation. IEEE, pp 4478–4483
5. Kasac J, Novakovic B, Majetic D, Brezak D (2006) Global positioning of robot manipulator with mixed revolute and prismatic joints. *IEEE Trans Autom Control* 51:1035–1040
6. Su YX, Zheng CH (2017) PID control for global finite-time regulation of robotic manipulators. *Int J Syst Sci* 48:547–558
7. Mendoza M, Zavala-r o A, Santibanez V, Reyes F (2015) A generalized PID-type control scheme with simple tuning for the global regulation of robot manipulators with constrained inputs. *Int J Constr* 88:1995–2012
8. Akyuz IH, Yolacan E, Ertunc HM, Bingul Z (2011) PID and state feedback control of single-link flexible joint robot. In: International conference on mechatronics. IEEE, pp 409–414
9. Zavala-Rio A, Santibanez V (2006) Simple extensions of the PD with desired gravity compensation control law for robot manipulators with bounded inputs. *IEEE Trans Control Syst Technol* 14:958–965
10. Bouhajar S, Maherzi E, Khraief N, Besbes M, Belghith S (2015) Trajectory generation using predictive PID control for stable walking humanoid robot. In: International conference on advanced wireless information and communication technology (AWICT2015) *procedia computer science*, vol 73, pp 86–93
11. Leen G, Ray G (2012) A set of decentralized PID controllers for an n-link robot manipulator. *Indian Acad Sci Sadhana* 37:405–423
12. Su Y, Muller PC, Zheng CH (2007) A global asymptotic stable output feedback PID regulator for robot manipulator. In: International conference on robotics and automation. IEEE, pp 44489–44844
13. Su YX, Sun D, Ren L, Mills JK (2006) Integration of saturated PI synchronous control and PD feedback for control of parallel manipulators. *IEEE Trans Robot* 22:202–207
14. Kasac J, Novakovic B, Majetic D, Brezak D (2006) Global positioning of robot manipulator with mixed revolute and prismatic joints. *IEEE Trans Autom Control* 51:1035–1040
15. Parra-Vega V, Arimoto S, Liu YH, Hirzinger G, Akella P (2003) Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. *IEEE Trans Robot Autom* 19(6):967–976
16. Gorez R (1999) Globally stable PID-like control of mechanical systems. *Syst Control* 38(1):61–72

17. Kelly R (1998) Global positioning of robot manipulators via PD control plus a class of nonlinear integral actions. *IEEE Trans Autom Control* 43(7):934–938
18. Xu J, Qiao L (2013) Robust Adaptive PID control of robot manipulator with bounded disturbances. *Math Probl Eng* 2013:1–13
19. Wang H, Zhu S, Liu S (2009) Adaptive PID control of robot manipulators with Hinf tracking performance. In: International conference on advanced intelligence and mechatronics. *IEEE/ASME*, pp 1515–1520
20. Perez J, Perez JP, Soto R, Flores A, Rodriguez F, Meza JL (2012) Trajectory tracking error using PID control law for two link robot manipulator via adaptive neural networks. *Procedia Technol* 3:139–146
21. Cvejn J, Tvrdik J (2017) Learning control of robot manipulator based on a decentralized position dependent PID controller. In: 1st International conference on process control (PC), pp 167–172
22. Dehghani A, Khodadadi H (2015) Fuzzy logic self-tuning PID control for a single-link flexible joint robot manipulator in the presence of uncertainty. In: 15th international conference on control, automation and systems (ICCAS), pp 186–191
23. Du M, Fang J, Wang L (2011) A parameter self-tuning fuzzy-PID control system for pneumatic manipulator of library robot. In: International conference on electronics, communications and control (ICECC), pp 4111–4115
24. Meza JL, Santibanez V, Soto R, Llama MA (2009) Stable fuzzy self-tuning PID control of robot manipulators. In: International Conference on Systems, Man, and Cybernetics. *IEEE*, pp 2624–2629
25. Norouzzadeh Ravari AR, Taghirad HD (2009) A novel hybrid fuzzy-PID controller for tracking control of robot manipulators. In: International conference in robotics and Biomimetic (ROBIO). *IEEE*, pp 16251–630
26. Yu W, Rosen J (2013) Neural PID control of robot manipulator with application to an upper limb exoskeleton. *IEEE Trans Cybern* 43(2):673–684
27. Nahapetian N, Jahed Motlagh MR, Analoui M (2009) PID gain tuning using Genetic algorithms and fuzzy logic for robot manipulator control. In: International conference on advanced computer control, pp 346–350
28. Nahapetian N, Jahed Motlagh MR, Analoui M (2009) Adaptive PID gain tuning using fuzzy logic and additional external performance index reference for controlling robot manipulator. In: International conference on advanced computer control, pp 448–452
29. Meza JL, Soto R, Arriaga J (2009) An optimal fuzzy self-tuning PID controller for robot manipulators via genetic algorithm. In: 8th Mescican international conference on artificial intelligence (MICAI), pp 21–26
30. Elkhateeb NA, Badr RI (2017) Novel PID tracking controller for 2DOF robotic manipulator system based on artificial bee colony algorithm. In: *Electrical control and communication engineering*, vol 13. Rig Technical University, pp 55–62
31. Mehdiyar F, Gholami HS, Kharrati H, Menhaj MB (2017) A modified fruit fly optimization algorithm and its application to control of robot manipulator. In: 5th International conference on control, instrumentation and automation (ICCIA), pp 120–125
32. Kankashvar MR, Kharrati H, Asl RM, Sadeghiani AB (2015) Designing PID controllers for a five-bar linkage robot manipulator using BBO algorithm. In: 6th International conference on modeling, simulation and applied optimization (ICMSAO), pp 1–6
33. Das MT, Dulger LC (2010) Control of SCARA robot: PSO–PID approach. *Control Intell Syst* 38(1):24–31
34. Al-Saed F, Mohammed AH (2012) Design and implementation of PSO–PID controller for MA2000 robotic manipulator. *Int J Comp Sci Eng Technol* 2(9):1427–1433
35. Djaneye-Boundjou O, Xu X, Ordenez R (2016) Automated particle swarm optimization based PID tuning for control of robotic arm. In: *IEEE national Aerospace and electronics conference (NECON) and Ohio innovation summit (OIS)*, pp 164–169
36. Thunyajareern S, Seeboonruang U, Kaitwanidvilai S (2017) PSO based adaptive force controller for 6DOF robot manipulators. In: *Proceeding of the world congress on engineering and computer sciences (WCECS)*, San Francisco, USA, pp 691–695
37. Pano V, Ouyang PR (2014) Gain tuning of position domain PID control using particle swarm optimization. *Robotica* 34(6):1351–1366
38. Kapoor N, Ohr J (2015) Improved PSO tuned classical controllers (PID and SMC) for robotic manipulator. *Int J Mod Educ Comp Sci* 1:47–54
39. Elkaranshway HA, Bayoumi EHE, Soliman HM (2011) PSO based robust PID control for flexible manipulator systems. *Int J Model Ident Constr* 14(1–2):1–12
40. Tsai PW, Nguyen TT, Dao TK (2016) Robot path planning optimization based on multi-objective grey wolf optimizer. In: *International conference on genetic and evolutionary computation*. Springer, Berlin, pp 166–73
41. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
42. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
43. Kennedy J, Eberhart R (1995) Particles swarm optimization. In: *International conference on neural networks*. *IEEE*, pp 1942–1948

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.