Turk J Elec Eng & Comp Sci, Vol.20, No.Sup.1, 2012, © TÜBİTAK
doi:10.3906/elk-1102-1011

Fractional PID controllers tuned by evolutionary algorithms for robot trajectory control

Zafer BİNGÜL*, Oğuzhan KARAHAN
Department of Mechatronics Engineering, Faculty of Engineering, Kocaeli University, Umuttepe 41380, Kocaeli-TURKEY
e-mails: zaferb@kocaeli.edu.tr, sebnemkn@hotmail.com

Received: 02.02.2011

Abstract

The aim of this paper is to compare the performances of a fractional order proportional integral derivative (FOPID) controller tuned with evolutionary algorithms for robot trajectory control. In order to make this comparison, a 2-degrees-of-freedom planar robot was controlled by a FOPID controller tuned with particle swarm optimization (PSO) and a real coded genetic algorithm (GA). In order to see the effects of the cost functions on the optimum parameters of the FOPID controller, 3 different cost functions were used: the root mean squared error (MRSE), mean absolute error (MAE), and mean minimum fuel and absolute error (MMFAE). In order to compare the performance of PSO and the GA under different conditions and to test the robustness of the FOPID controller tuned with these algorithms, the parameters of the system model and the given trajectory were changed and white noise was added to the system. All of the simulation results for the robot trajectory experiment show that the FOPID controller tuned by PSO has better performance than the FOPID controller tuned by the GA. Furthermore, the results obtained for the FOPID tuned by both PSO and the GA indicate the superiority of the proposed tuning approach for robot trajectory control.

Key Words: *Fractional order controller, PSO, GA, robot trajectory control*

1. Introduction

For linear systems, the proportional integral derivative (PID) controller has been widely used in industrial control processes because it has a simple structure and robust performance, and it is easily tuned in a wide range of operating conditions [1]. In spite of the fact that control theory has been developed significantly, PID controllers are still used for many industrial applications such as process controls, motor drivers, flight control, and instrumentation.

Fractional order dynamic systems and controllers have been increasing in interest in many areas of science and engineering in the last few years. Fractional order controllers are described by fractional order differential equations. Expanding derivatives and integrals to fractional orders can adjust the control system's frequency

*Corresponding author: Department of Mechatronics Engineering, Faculty of Engineering, Kocaeli University, Umuttepe 41380, Kocaeli-TURKEY

response directly and continuously. Controllers consisting of fractional order derivatives and integrals have been used in industrial applications [2] and various fields such as power electronics [3], system identification [4], robotic manipulators [5], irrigation canal control [6], mechatronics systems [7,8], and heat diffusion systems [9]. It should be noted that there are a growing number of physical systems whose behavior can be compactly determined using the fractional order system theory and can be controlled with fractional order proportional-integral-derivative (FOPID) controllers [10], even if the system has unstable or time delay behaviors [11]. There are many aspects that should be taken into account when designing these controllers. In the FOPID controller, the 5 parameters $(K_p, K_d, K_i, \lambda, \mu)$ need to be tuned based on some design specifications. The desired specifications for the controllers are usually to achieve robust to load disturbances, high frequency noise, and uncertainties of the plant model. Taking into account all of the constraints in the tuning method of the FOPID controller, the optimal set of values for K_p, K_d, K_i, λ and μ can be found. The ultimate goal of this paper is to develop a method to find the optimum parameters of the FOPID controller under given constraints. However, the method developed should be simple and reliable. Evolutionary algorithms seem to be a good potential tuning method for these requirements.

It is quite difficult to optimize the parameters of the FOPID controller in linear and nonlinear systems. There is a need for an effective and efficient global approach to optimize these parameters automatically. Several evolutionary optimization algorithms such as the genetic algorithm (GA) [12-14], differential evolution algorithm (DE) [15,16], and particle swarm optimization (PSO) [17-20] have been proposed to optimize the parameters of the several controllers.

PSO is a relatively new evolutionary algorithm that may be used to find optimal or near optimal solutions in a large search space. The PSO algorithm is especially useful for parameter optimization in continuous, multidimensional search spaces. The PSO method can generate a high quality solution within a shorter calculation time and it tends to converge very fast compared to other stochastic methods. Moreover, it is implemented easily in most of the programming languages.

There have been many studies related with the evolutionary algorithm-based design of the FOPID and PID controllers in the literature. Chang et al. [12] proposed a novel adaptive GA for the multiobjective optimization design of a FOPID controller and applied it to the control of an active magnetic bearing system. They found that the fractional PID controllers have remarkably reduced the overshoot and settling time compared with the optimized conventional PID controller. Aldair and Wang [13] designed an optimal FOPID controller for a full vehicle nonlinear active suspension system. Their results illustrate clearly the effectiveness and robustness of the proposed controller. Lee et al. [14] developed a design method of a FOPID controller via a hybrid of an electromagnetism-like algorithm and the GA for a second order system with time delay. Simulation results showed that the hybrid system has better ability of global optimization and faster convergence. Bingul [15] employed the differential evolution (DE) algorithm to tune a PID controller for unstable and integrating processes with time delay. The results showed that a faster settling time, less or no overshoot, and higher robustness were obtained with the PID tuned DE. Biswas et al. [16] designed FOPID controllers using the DE algorithm and compared the DE with PSO and the GA in the speed control of a DC motor, second order plant, and fractional order plant. The DE-based method could find better solutions consuming a lesser amount of computational time. Cao and Cao [17] demonstrated the parameter optimization of a fractional order controller based on a modified PSO. In their paper, the improved PSO could achieve faster search speed and better solution compared to the GA. Maiti et al. [18] employed PSO for designing fractional order PID controllers. They reduced significantly the percentage of overshoot, rise, and settling times using FOPID controllers compared to a PID

controller. Majid et al. [19] employed the PSO algorithm to determine 5 parameters of the FOPID controller for an automatic voltage regulator. The results of the paper showed that PSO could perform an efficient search for optimal FOPID parameters. Moreover, the proposed FOPID controller has more robust and better performance characteristics in comparison with the PID controller. Alfi and Modares [20] found optimal system parameters for an unstable nonlinear system and optimal parameters of the PID controller using a novel adaptive PSO (APSO). They compared the APSO with a linearly decreasing inertia weight PSO (LDW-PSO) and the GA. The APSO has a faster convergence speed than the GA and LDW-PSO.

The rest of this article is organized as follows: the FOPID controller used in the robot trajectory control is presented in Section 2, the dynamic model of the robot arm is described in Section 3, the PSO-tuning method for the FOPID controller is described in Section 4, and the experimental results and conclusion are given in Sections 5 and 6, respectively.

2. Fractional order PID controller

One of the possibilities for improvements in the quality and robustness of PID controllers is to use fractional order controllers with noninteger derivation and integration parts. The $PI^{\lambda}D^{\mu}$ controller generalizes the PID controller involving an integrator of order λ and a differentiator of order μ.

The differential equation of the $PI^{\lambda}D^{\mu}$ controller is given as follows:

$$u(t) = k_p e(t) + k_i D_t^{-\lambda} e(t) + k_d D_t^{\mu} e(t). \tag{1}$$

The continuous transfer function of the FOPID controller is obtained by means of the Laplace transformation, as given by:

$$G_c(s) = K_P + K_I s^{-\lambda} + K_D s^{\mu}. \tag{2}$$

For designing a FOPID controller, 3 parameters (K_P, K_I, K_D) and 2 orders (λ, μ) with nonintegers should optimally determined for a given system.

3. Dynamic model of the planar robot

To test the performance of the proposed controllers, the planar robot with 2 degrees of freedom (DOF), shown in Figure 1, is selected as an example problem.

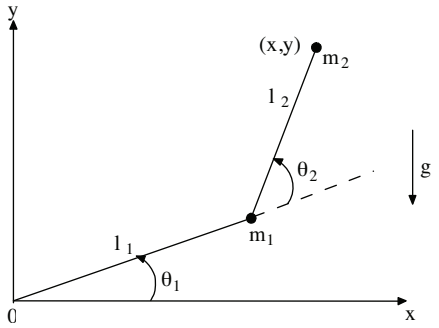

Figure 1. Model of a 2-DOF planar robot.

The dynamic equations of the serial robot are usually represented by the following coupled nonlinear differential equations:

$$\tau = D(q)\ddot{q} + C(q,\dot{q}) + G(q). \tag{3}$$

The joint variable q is an n-vector containing the joint angles for the revolute joints. The dynamic equation of the 2-DOF planar robot was computed in [21] as follows:

$$\begin{aligned}\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{pmatrix} (m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos\theta_2 & m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2 \\ m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2 & m_2 l_2^2 \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} \\ &+ \begin{pmatrix} -m_2 l_1 l_2 (2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2)\sin\theta_2 \\ m_2 l_1 l_2 \dot{\theta}_1^2 \sin\theta_2 \end{pmatrix} + \begin{pmatrix} (m_1+m_2) g l_1 \cos\theta_1 + m_2 g l_2 \cos(\theta_1+\theta_2) \\ m_2 g l_2 \cos(\theta_1+\theta_2) \end{pmatrix}.\end{aligned} \tag{4}$$

4. Particle swarm optimization

In this section, the PSO algorithm is used for the FOPID design in a 2-DOF robot arm.

4.1. PSO algorithm

PSO is an evolutionary computational technique based on the movement and intelligence of swarms looking for the most fertile feeding location. A "swarm" is an apparently disorganized collection (population) of moving individuals that tend to cluster together, while each individual seems to be moving in a random direction. PSO uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution [22,23].

Each particle is treated as a point in an n-dimensional space and adjusts its "flying" according to its own flying experience, as well as the flying experience of other particles. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) that has been achieved so far. This value is called *pbest*. Another best value called *gbest* is that obtained so far by any particle in the neighbors of the particle.

The PSO concept consists of changing the velocity (or acceleration) of each particle toward its *pbest* and the *gbest* position at each time step. Each particle tries to modify its current position and velocity according to the distance between its current position and *pbest*, and the distance between its current position and the *gbest*. At each step n, by using the individual best position, *pbest*, and global best position, *gbest*, a new velocity for the ith particle is updated by:

$$V_i(n) = \chi\left(V_i(n-1) + \varphi_1 r_1 \left(pbest_i - P_i(n-1)\right) + \varphi_2 r_2 \left(gbest - P_i(n-1)\right)\right). \tag{5}$$

χ is defined below:

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \varphi = \varphi_1 + \varphi_2, \varphi > 4 \tag{6}$$

The velocity is confined within the range of $[-v_{\max}, +v_{\max}]$. If the velocity violates these limits, it is forced to its proper values. Changing velocity in this way enables the ith particle to search around its local best position, *pbest*, and global best position, *gbest*. Based on the updated velocity, each particle changes its position as follows:

$$P_i(n) = P_i(n-1) + V_i(n). \tag{7}$$

4.2. Optimization of the FOPID controller with PSO

Since each FOPID controller has 5 parameters, there are a total of 10 parameters to be optimized with PSO. The PSO algorithm searches all of the controller parameters in the 10 dimensional spaces. The particle includes 10 elements assigned real values.

The order of a particle is shown as follows:

$$P_i = [K_{p_1}, K_{d_1}, \mu_1, K_{i_1}, \lambda_1, K_{p_2}, K_{d_2}, \mu_2, K_{i_2}, \lambda_2]\,. \tag{8}$$

In the above, the parameters $K_{p_1}..., \lambda_1$ and $K_{p_2}..., \lambda_2$ correspond to the elements of the 1st controller and the 2nd controller, respectively.

4.3. Cost functions

The most crucial step in applying PSO is to choose the best cost function, which is used to evaluate the fitness of each particle. During the tuning process with PSO, 1 of the 3 different cost functions [root mean squared error (MRSE), mean absolute error (MAE), and mean minimum fuel and absolute error (MMFAE)] is employed. These cost functions for the ith particle are given below.

$$E_{MRSE}(k) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(e_1^2(i) + e_2^2(i))} \tag{9}$$

$$E_{MAE}(k) = \frac{1}{N}\sum_{i=1}^{N}|e_1(i)| + |e_2(i)| \tag{10}$$

$$E_{MMFAE}(k) = \frac{1}{N}\sum_{i=1}^{N}|e_1(i)| + |e_2(i)| + |u_1(i)| + |u_2(i)| \tag{11}$$

4.4. Tuning of the FOPID using PSO

A block diagram of the robot system controlled with the FOPID controllers is shown in Figure 2. All of the simulations are performed here in MATLAB 7.0.1. Figure 3 illustrates the block structure of the FOPID controller optimizing process with PSO. All of the parameters of the FOPID controllers are updated at every simulation time (t_f).

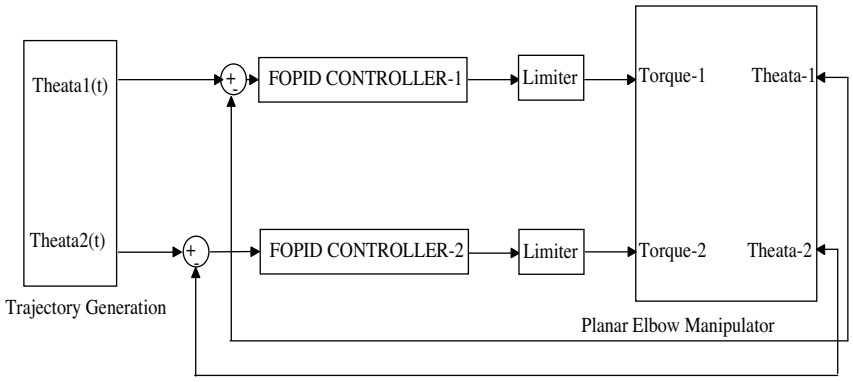

Figure 2. Simulink model of the robot and FOPID controllers.

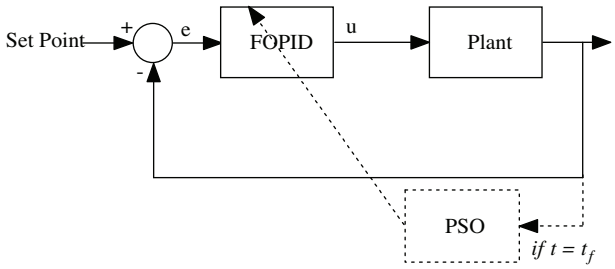

Figure 3. Tuning process of the FOPID controller parameters with PSO.

In this study, the population size is set to 10 particles for considering the excessive computational load. As each particle P_i has 10 elements, the swarm size is 10×10. It should be noted that the values of an element in the particle may exceed its reasonable range. In this case, inspired from practical requirements and from the papers focusing on tuning the parameters of the FOPID in application of the different systems, the lower bound of the FOPID parameters is 0 and their upper bounds are set to $K_{p_{\max}} = K_{i_{\max}} = 2000, K_{d_{\max}} = 100$, and $\lambda_{\max} = \mu_{\max} = 2$. The initial values of the particles are randomly generated based on the maximum values in the first generation.

In the PSO block (Figure 3), the cost function is calculated for each particle, and *pbest* and *gbest* are computed for every final time (t_f). A particle velocity [Eq. (5)] is calculated for each particle and the particle position [Eq. (7)] is updated. This algorithm is run until maximum iteration or minimum cost criteria are attainable. In the PSO algorithm, the parameters χ, φ_1, and φ_2 are set to 0.76, 2.05, and 2.05 [22,23]. For the given trajectory, the PSO algorithm with 1 of the 3 different cost functions is processed for 60 generations and repeated for 10 runs. The training trajectories for the robot joints are given below.

$$\theta_1(t) = 0.1524 + 0.24384\cos(\frac{2\pi t}{5} - \frac{\pi}{2}) \tag{12}$$

$$\theta_2(t) = 0.39624 + 0.24384\sin(\frac{2\pi t}{5} - \frac{\pi}{2}) \tag{13}$$

Figure 4 shows the movement of 1 particle (*Particle 1*) in the swarm from the 1st generation to the last generation along the given trajectory.

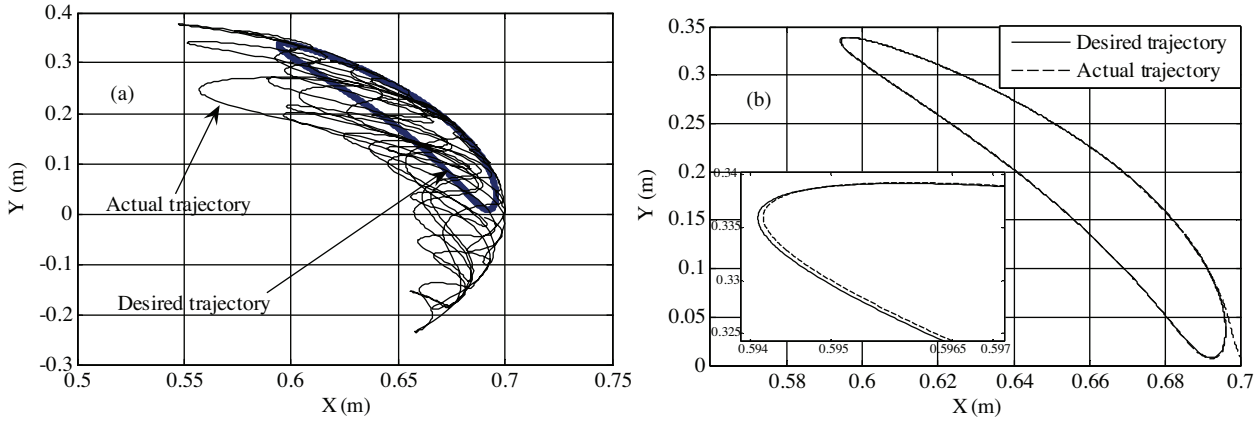

Figure 4. One particle movement from the initial generation (a) to the last generation (b) during the tuning process.

5. Tuning of the FOPID controller using the GA

In order to emphasize the advantages of the proposed design method, the FOPID-GA controller, based on the real-value GA, is implemented. The parameters of the FOPID controller are also tuned with the GA using the 3 different cost functions defined above. In order to have an accurate comparison, the optimization conditions for the FOPID-PSO and FOPID-GA should be the same. Therefore, the range of the possible values of the control signal in the optimization process is restricted to the upper bound of 30 Nm and 20 Nm and the lower bound of –14 Nm and –16 Nm for the 1st joint and the 2nd joint, respectively. In order to have a better comparison, the initial range of the parameters for the FOPID controller is selected to be the same as that of PSO. Parameters of the GA algorithm are chosen as below.

- Selection: normalized geometric select ('*normGeomSelect*')
- Fitness function: MRSE, MAE, and MMFAE
- Crossover: arithmetic crossover ('*arithXover*')
- Mutation: uniform method ('*unifMtuation*')
- Number of FOPID parameters : 5
- Population number: 10
- Generation number: 100

In order to compare the search performance of the different evolutionary techniques, PSO algorithm and the GA are applied to the FOPID controller optimization with 3 different cost functions. Figure 5 shows the fitness values of both algorithms with the MRSE cost function, and as can be seen, the fitness value of the FOPID-PSO is decreased to 0.0027 after 55 generations. On the other hand, the fitness value of the FOPID-GA is decreased to 0.0031 after 82 generations. It is clear from Figure 5 that PSO converges fast initially, but requires more generations later to reach the optimal point. As can be seen, through about 60 generations, the PSO algorithm provides better convergence. Furthermore, the results obtained here show that the PSO algorithm can search optimal FOPID controller parameters more quickly and efficiently than the GA algorithm.

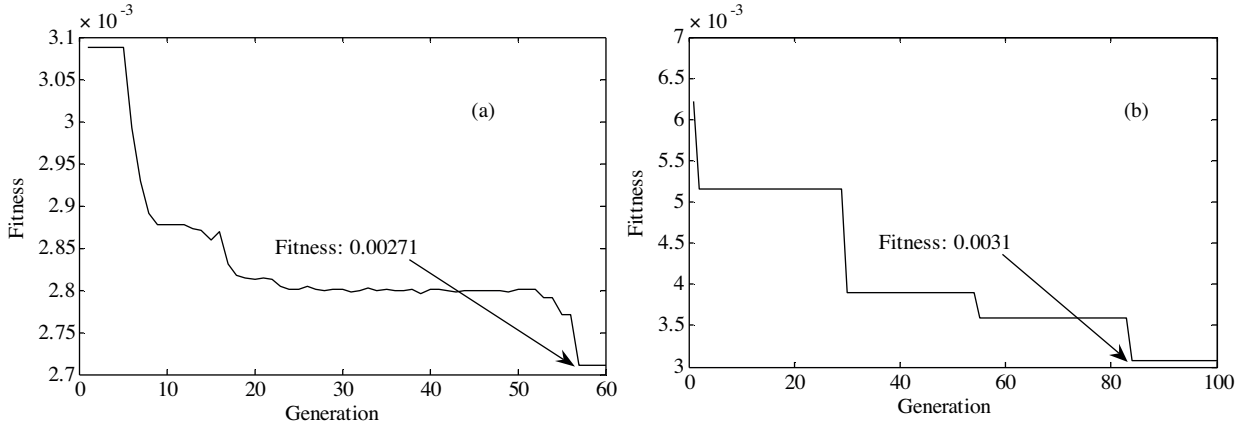

Figure 5. The convergence behaviours of the FOPID-PSO (a) and FOPID-GA (b) controllers with the MRSE cost function during the optimization process.

6. Simulations and discussion

For 3 different cost functions, the parameters of the FOPID controller tuned with 2 different algorithms are summarized in Tables 1 and 2, respectively.

Table 1. Tuned parameters of the designed $FOPI^{\lambda}D^{\mu}$ controller with PSO for the 3 different cost functions.

FOPID parameters	PSO-MRSE		PSO-MAE		PSO-MMFAE	
	Joint 1	Joint 2	Joint 1	Joint 2	Joint 1	Joint 2
K_p	994	1086	677	1650	241	273
K_i	603	25	582	117	1550	147
K_d	31	8	23	17	27	11
λ	1.014	1.044	0.988	1.018	1.0329	1.1350
μ	1.011	1.1711	0.985	1.009	0.9906	0.8991

Table 2. Tuned parameters of the designed $FOPI^{\lambda}D^{\mu}$ controller with the GA for the 3 different cost functions.

FOPID parameters parameters	GA-MRSE		GA-MAE		GA-MMFAE	
	Joint 1	Joint 2	Joint 1	Joint 2	Joint 1	Joint 2
K_p	1074	329	980	1339	157	317
K_i	628	984	660	75	1030	1157
K_d	21	18	30	19	31	21
λ	1.1423	0.9080	1.15	0.98	0.5427	0.4256
μ	1.0274	0.8333	0.946	0.89	1.0883	0.8739

From Tables 1 and 2, the parameters of the FOPID-PSO controller for the cost function MRSE are approximately close to that of the FOPID-GA controller for joint 1. However, all of the parameters obtained from both algorithms for joint 2 are different from each other. The reason is that joint 2 is very sensitive to parameter changes in the controller. As expected, each cost function produces different optimum controller parameters based on its priority. Specifically, the MMFAE cost function yields controller parameters that are very different from those of the other cost functions. K_p, K_i, and K_d in the FOPID-PSO controller change with the different cost functions. However, there is very little change in λ and μ for the different cost functions. For the MMFAE, λ in the FOPID-GA controller alters dramatically from the values obtained from the other cost functions for both joints. A comparison in terms of the cost functions is given in Table 3. In the case of no disturbance, PSO has better fitness values.

Table 3. Cost function values of the 3 different controllers without any disturbance.

Cost functions	FOPID-PSO	FOPID-GA
MRSE	0.0027	0.0031
MAE	0.0030	0.0032
MMFAE	7.2380	7.2465

The objective of this section is to test the robustness of the FOPID-PSO and FOPID-GA controllers and to show their performances. In order to study the robustness of the controllers, the mass of the robot arm for each joint is increased 3-fold. For this case, the controllers' parameters tuned by PSO and the GA are kept unchanged. The performances of the controllers are compared to each other for different trajectories. Figure 6 shows the position and control signal of the controllers with the MRSE cost function along the trajectory

tracking. The simulations are performed by using the MAE and MMFAE as well, and it can be observed that the same PSO advantages arrived at using the MRSE are still valid. Considering all of the simulation results, the FOPID-GA controller has a slower response time compared to the FOPID-PSO controller. The FOPID-PSO controller successfully tracks the trajectory for both arms. It can also be seen that the trajectory tracking performance obtained with PSO is more robust to the variation of the mass compared to the GA.

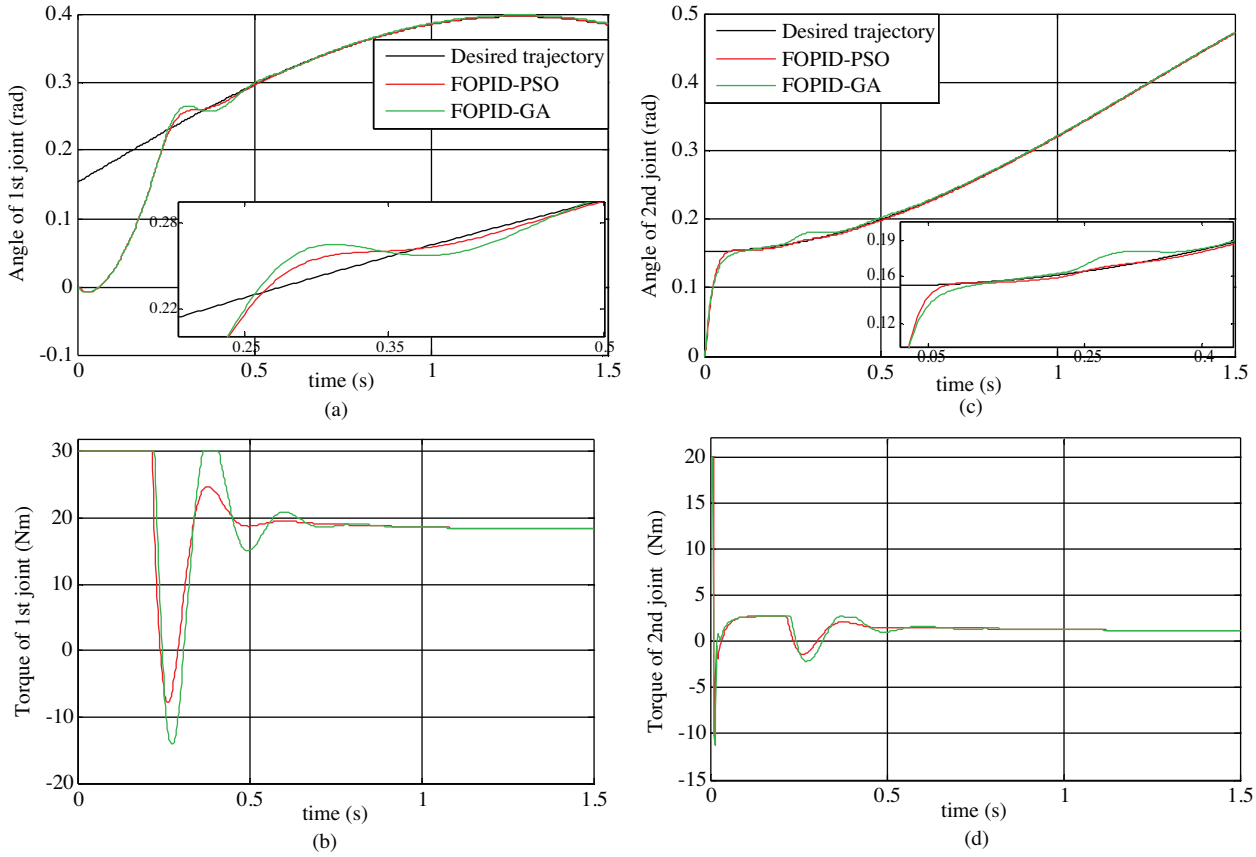

Figure 6. Positions (a and c) and control signal (b and d) of the FOPID-PSO and FOPID-GA controllers with the MRSE cost function under mass changes.

In the 2nd case for the robustness evaluation of the controllers, external disturbances such as the white noise with different variances are added to the robot system. The noise is applied to the 1st joint, the 2nd joint, and then both joints, respectively. For the experiments related to different noise variances, the values of the MRSE cost function are given in Table 4. Similar results are observed when the MAE and MMFAE cost functions are used. As can be seen from the results given in Table 4, the FOPID-PSO controller produces better results than the FOPID-GA controller and provides more robust control performance, ensuring good disturbance rejection. When the noise variance is increased, it can be seen that the deviations from the given trajectory occur.

In the 3rd robustness test, the values of the amplitude, A, and phase, ϕ, in the given trajectory are changed for joint 1, joint 2, and then both joints, respectively, and the frequency of the given trajectories is also varied. For these experiments, the FOPID-PSO controller is robust to changes in the given trajectory. Among these experiments, one is selected to illustrate as an example. Figure 7 shows the desired and actual

positions and control signals of the joints for controllers with the MRSE under changes in both the amplitude and the phase. The FOPID-PSO controller has more robust stability and performance characteristics than the FOPID-GA controller for both joints.

Table 4. MRSE cost function values in the case of adding noise with different variances.

	Joint 1		Joint 2		Joints 1 and 2	
Noise power	FOPID-PSO	FOPID-GA	FOPID-PSO	FOPID-GA	FOPID-PSO	FOPID-GA
0.1	0.0082	0.0088	0.0085	0.0099	0.0086	0.0103
0.5	0.0090	0.0110	0.0115	0.0140	0.0115	0.0156
0.7	0.0094	0.0122	0.0138	0.0165	0.0136	0.0187
0.8	0.0096	0.0129	0.0151	0.0179	0.0149	0.0203

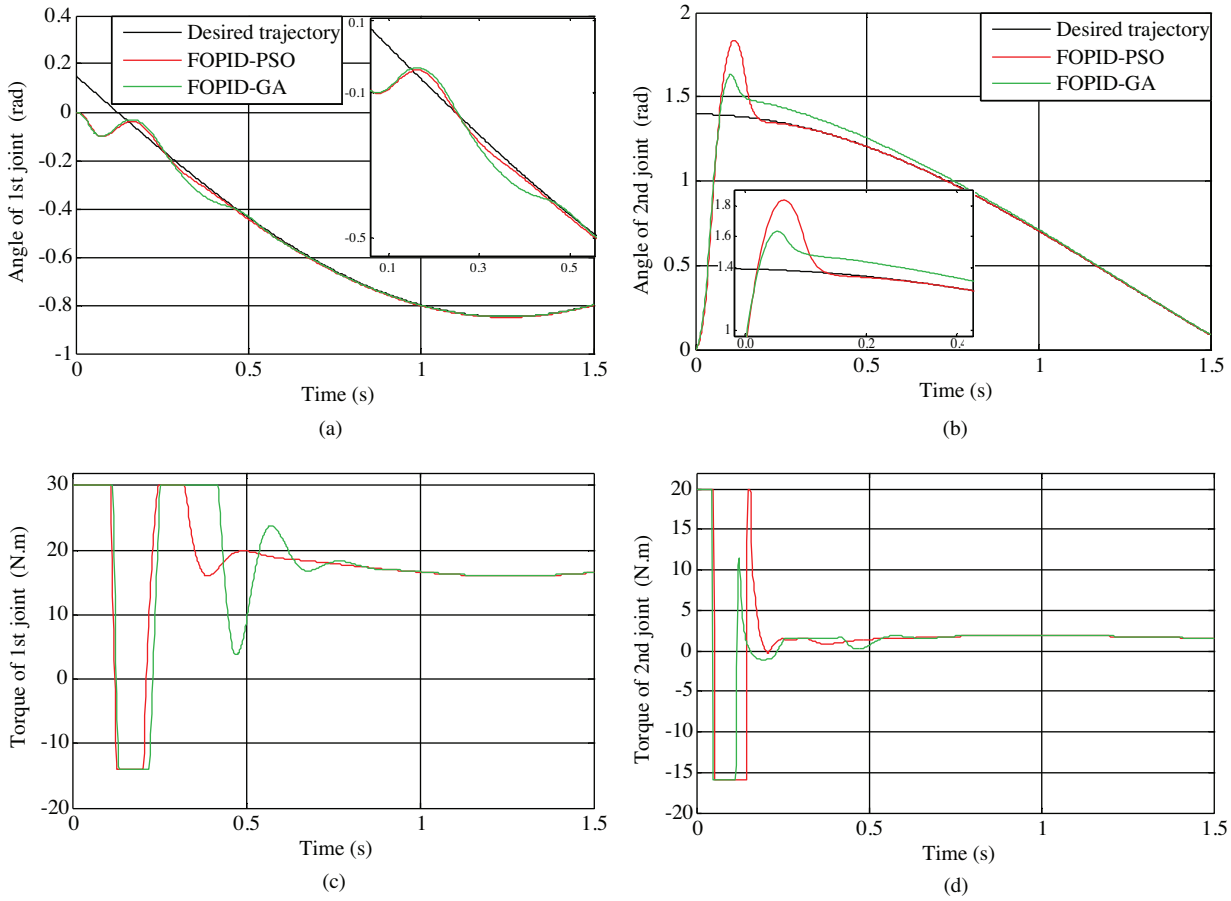

Figure 7. Position (a and b) and control signal (c and d) of the FOPID-PSO and FOPID-GA controllers with the MRSE under changes in both the amplitude and the phase.

In order to see the effects of frequency changes in the given trajectory on tracking performance, different frequency values (2π and 3π) are applied to the controlled robot system. In this case, the responses of the controlled system with the FOPID-PSO and FOPID-GA are shown in Figures 8 and 9.

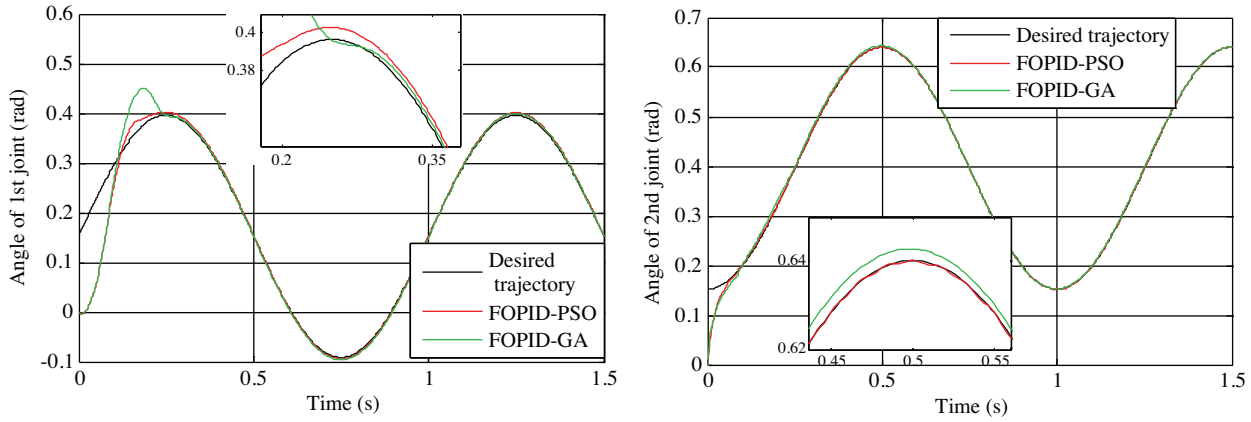

Figure 8. Frequency responses of the FOPID-PSO and FOPID-GA controllers with the MRSE at a frequency of 2π Hz.

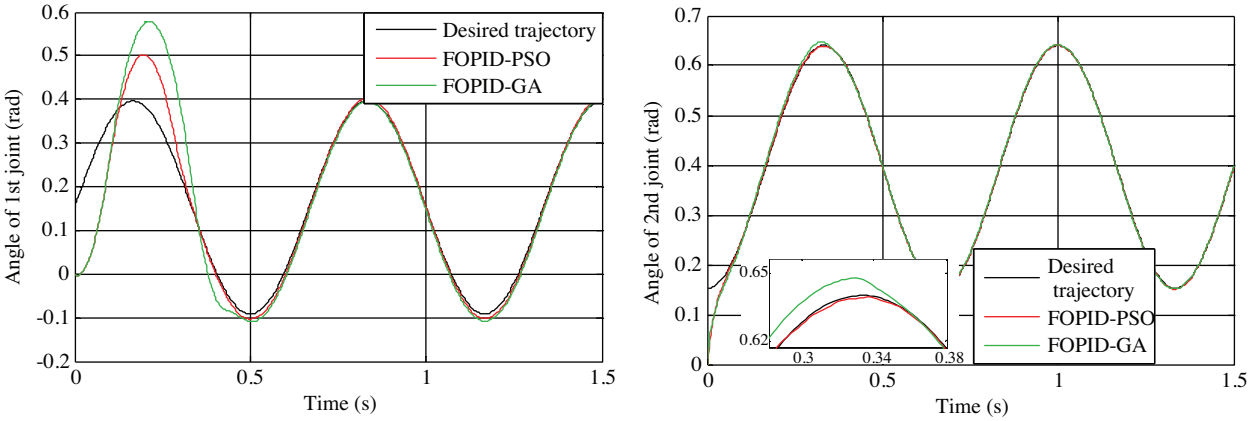

Figure 9. Frequency responses of the FOPID-PSO and FOPID-GA controllers with the MRSE at a frequency of 3π Hz.

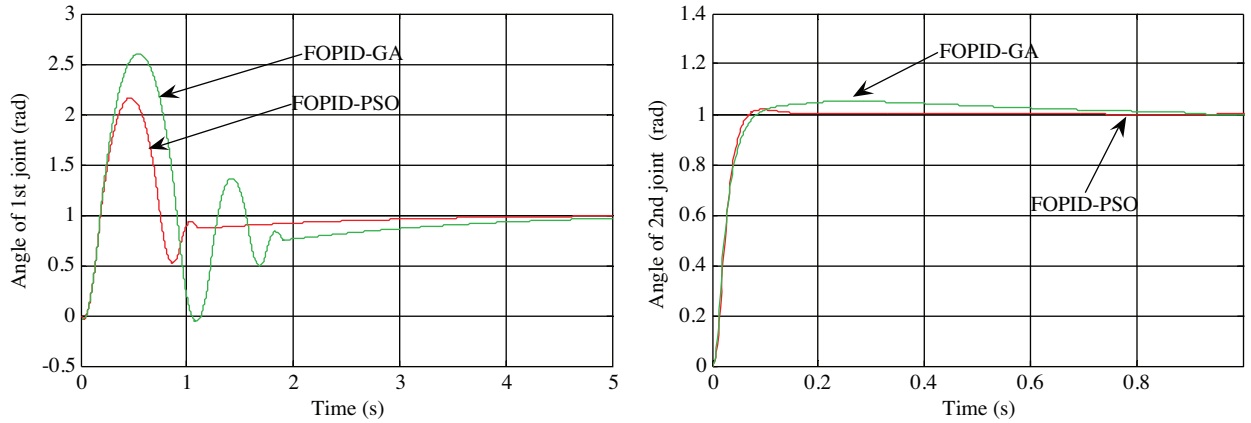

Figure 10. Step responses of the FOPID-PSO and FOPID-GA controllers with the MRSE.

As can be seen from Figures 8 and 9, the FOPID-PSO controller with the MRSE cost function is more robust and has better trajectory tracking than the FOPID-GA.

In order to see the position control performance of the controllers, the unit step is applied to the robot arm. Figure 10 shows the responses of the FOPID-PSO and FOPID-GA controllers with the MRSE cost function. Furthermore, the simulations are performed using the MAE and MMFAE, and it is observed that the same PSO advantages arrived at using the MRSE are still valid.

As can be seen in Figure 10, the dynamic properties (overshoot and settling time) of the controlled system response obtained from the FOPID-PSO controller are much better than those of the FOPID-GA controller.

7. Conclusion

This paper introduced an intelligent optimization method for FOPID controllers tuned with evolutionary algorithms. In order to evaluate the performance of the controller, trajectory tracking of a 2-DOF planar robot was done. The robust design of the FOPID controller is difficult to compare to the PID controller, since the FOPID controller includes more parameters. All of the parameters related to the FOPID controller were determined using PSO and the GA. In order to examine the effects of the cost functions on the controller parameter optimization, 3 different cost functions were used in the algorithms. The performance of PSO and the GA was compared with several simulation experiments. Moreover, the robustness of the FOPID-PSO and FOPID-GA controllers was tested in the case of mass changes, the presence of noise with different variances, and different trajectories (amplitude, phase, and frequency change). Position control performance of the controllers was studied by applying step input.

Considering all of the results from the simulation experiments, the FOPID-PSO controller can achieve good performance and robustness, superior to those obtained with the FOPID-GA controller. Moreover, PSO can achieve faster search speed and better solutions compared to the GA. The FOPID-PSO controller has good tracking performance in comparison with the FOPID-GA controller. In addition, the FOPID-PSO controller enhanced the flexibility and stability of the PID controller. Furthermore, the implementation of the controller tuning with PSO is much easier than with the traditional methods because there is no need for derivative knowledge or complex mathematical equations. In future studies, fuzzy-FOPID will be developed using these optimized FOPID controllers.

Nomenclature

K_p	Proportional gain
K_d	Derivative gain
K_i	Integral gain
λ	Integrator order
μ	Differentiator order
$D(q)$	Inertia matrix
$C(q,\dot{q})$	Coriolis/centripetal matrix
$G(q)$	Gravity vector
τ	Torque
q	Joint variable
m_i	ith link mass
l_i	ith link length
θ_i	ith joint position
$\dot{\theta}_i$	ith joint velocity
$\ddot{\theta}_i$	ith joint acceleration
P_i	ith particle position vector
r_1 and r_2	Random numbers between 0 and 1
φ_1 and φ_2	Positive constant learning rates
χ	Constriction factor
$e_1(i)$	Trajectory error of the 1st joint for the ith step
$e_2(i)$	Trajectory error of the 2nd joint for the ith step
N	A number of sample
k	Iteration number
$u_1(i)$	Control signal of the 1st joint
$u_2(i)$	Control signal of the 2nd joint
t_f	Simulation time

References

[1] H.A. Varol, Z. Bingul, "A new PID tuning technique using ant algorithm", Proceedings of the American Control Conference, pp. 2154- 2159, 2004.

[2] B.M. Vinagre, C.A. Monje, A.J. Calderón, J.I. Suárez, "Fractional PID controllers for industry application: a brief introduction", Journal of Vibration and Control, Vol. 13, pp. 1419-1429, 2007.

[3] A.J. Calderón, B.M. Vinagre, V. Feliu, "Fractional order control strategies for power electronic buck converters", Signal Processing, Vol. 86, pp. 2803-2819, 2006.

[4] M. Schlegel, M. ech, "Fractal system identification for robust control - the moment approach", Proceedings of the 5th International Carpathian Control Conference, pp. 1-6, 2004.

[5] N.M. Fonseca Ferreira, J.A. Tenreiro Machado, "Fractional-order hybrid control of robotic manipulators", 11th International Conference on Advanced Robotics, pp. 393-398, 2003.

[6] V. Feliu-Batlle, R. Rivas Pérez, L. Sánchez Rodríguez, "Fractional robust control of main irrigation canals with variable dynamic parameters", Control Engineering Practice, Vol. 15, pp. 673-686, 2007.

[7] H. Fan, Y. Sun, X. Zhang, "Research on fractional order controller in servo press control system", Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, pp. 934-2938, 2007.

[8] C. Ma, Y. Hori, "The application of fractional order PID controller for robust two-inertia speed control", Proceedings of the 4th International Power Electronics and Motion Control Conference, 2004.

[9] I.S. Jesus, J.A. Tenreiro Machado, "Fractional control of heat diffusion systems", Nonlinear Dynamics, Vol. 54, pp. 263-282, 2008.

[10] I. Podlubny, "Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers", IEEE Transactions on Automatic Control, Vol. 44, pp. 208-214, 1999.

[11] S.E. Hamamci, M. Koksal, "Calculation of all stabilizing fractional-order PD controllers for integrating time delay systems", Computers and Mathematics with Applications, Vol. 59, pp. 1621-1629, 2010.

[12] L.Y. Chang, H.C. Chen, "Tuning of fractional PID controllers using adaptive genetic algorithm for active magnetic bearing system", WSEAS Transactions on Systems, Vol. 8, pp. 226-236, 2009.

[13] A.A. Aldair, W.J. Wang, "Design of fractional order controller based on evolutionary algorithm for a full vehicle nonlinear active suspension systems", International Journal of Control and Automation, Vol. 3, pp. 33-46, 2010.

[14] C.H. Lee, F.K. Chang, "Fractional-order PID controller optimization via improved electromagnetism-like algorithm", Expert Systems with Applications, Vol. 37, pp. 8871-8878, 2010.

[15] Z. Bingul, "A new PID tuning technique using differential evolution for unstable and integrating processes with time delay", Proceedings of the 11th International Conference on Neural Information Processing, pp. 254-260, 2004.

[16] A. Biswas, S. Das, A. Abraham, S. Dasgupta, "Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution", Engineering Applications of Artificial Intelligence, Vol. 22, pp. 343-350, 2009.

[17] J.Y. Cao, B.G. Cao, "Design of fractional order controller based on particle swarm optimization", International Journal of Control, Automation and Systems, Vol. 4, pp. 775-781, 2006.

[18] D. Maiti, S. Biswas, A. Konar, "Design of a fractional order PID controller using particle swarm optimization technique", Second National Conference on Recent Trends in Information Systems, Vol. 30, 2008.

[19] M. Zamania, M. Karimi-Ghartemanib, N. Sadatib, M. Parnianib,, "Design of a fractional order PID controller for an AVR using particle swarm optimization", Control Engineering Practice, Vol. 17, pp. 1380-1387, 2009.

[20] A. Alfi, H. Modares, "System identification and control using adaptive particle swarm optimization", Applied Mathematical Modelling, Vol. 35, pp. 1210-1221, 2011.

[21] F.L. Lewis, D.M. Dawson, C.T. Abdallah, Control of Robot Manipulators, New York, Macmillan, 1993.

[22] J. Kennedy, R.C. Eberhart, Y. Shi, Swarm Intelligence, New York, Morgan Kaufmann, 2001.

[23] R.C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39-43, 1995.