Hindawi
Journal of Robotics
Volume 2019, Article ID 5052185, 10 pages
https://doi.org/10.1155/2019/5052185

Research Article

Inverse Dynamics Based Optimal Fuzzy Controller for a Robot Manipulator via Particle Swarm Optimization

M. J. Mahmoodabadi and A. Ziaei

Department of Mechanical Engineering, Sirjan University of Technology, Sirjan, Iran

Correspondence should be addressed to M. J. Mahmoodabadi; mahmoodabadi@guilan.ac.ir

Received 23 August 2018; Revised 4 December 2018; Accepted 23 December 2018; Published 1 January 2019

Academic Editor: Yangmin Li

This paper endeavors to contribute to the field of optimal control via presenting an optimal fuzzy Proportional Derivative (PD) controller for a RPP (Revolute-Prismatic-Prismatic) robot manipulator based on particle swarm optimization and inverse dynamics. The Denavit-Hartenberg approach and the Jacobi method for each of the arms of the robot are employed in order to gain the kinematic equations of the manipulator. Furthermore, the Lagrange method is utilized to obtain the dynamic equations of motion. Hence, in order to control the dynamics of the robot manipulator, inverse dynamics and a fuzzy PD controller optimized via particle swarm optimization are used in this research study. The obtained results of the optimal fuzzy PD controller based on the inverse dynamics are compared to the outcomes of the PD controller, and it is illustrated that the optimal fuzzy PD controller shows better controlling performance in comparison with other controllers.

1. Introduction

In the recent years, the robotic arms have been used in the industrial applications, to name but a few, assembly lines [1], painting operation via spray [2, 3], and welding [4] for the wide use of 24/7. Indeed, many robots and mechanical arms have been inspired from humans' arms and characteristics. The structure of this kind of robots involves rigid bodies which are connected through sets of different joints. As a matter of fact, the RPP robot is a kind of the robots being called series robots. While series robots benefit from appropriate characteristics such as large workspace [5], the end effector is attached to the base, which is opposite to the characteristics of the parallel robots [6]. As mentioned, while this gives a vast workspace, it causes that the load is not distributed on all the kinematic chains and the stiffness of the robot is also diminished.

One of the most crucial problems raised in the study of the direct kinematics of the robot is how the robot's framework is changed when moving [7–9]. The Denavit-Hartenberg approach was utilized to obtain the kinematic equations of the robot. Moreover, the inverse kinematics were employed to determine the amount of changes in the joints' variables with regard to the position and the orientation of the end effector [10–14]. The approach used to gain the dynamics of the robot is the Lagrange method [15–18]. As these robots have been introduced as a controllable system, various approaches including classic, intelligent, and combinatory control methods were proposed and examined. The examples of the classic control methods are Proportional Derivative (PD) and sliding mode control [19–23]. Moreover, the instances of the intelligent control approaches involve fuzzy control and neural control methods [24, 25].

In order to control the challenging dynamics of the robots, the nonlinearity in the dynamics must be turned into linearity via using techniques such as inverse dynamics. Afterwards, by choosing an appropriate controller, such as PD control, it is feasible to control the challenging dynamics with high precision [19, 20, 26–30]. A proper controller not only provides a superior performance in tracking the desired trajectory but also diminishes the negative effects of disturbances, the clearance of joints, and the elasticity of the links or joints. In order to augment the efficacy of the control approach, researchers used the benefits of the fuzzy controllers in association with other controllers. A number of research studies have been conducted on the design of the fuzzy logic controllers (FLC). For instance, the parameters of FLC were optimized via particle swarm optimization to control a robot following a trajectory [31–33]; the Lyapunov synthesis was employed to design FLC in order to address the adjustment of the output of an actuator with nonlinear backlash [34–36].

Particle swarm optimization (PSO) is a smart swarm-based technique introduced by Kennedy and Eberhart [37], where the algorithm based upon swarm intelligence uses random population and adaptive optimization inspired from the social behavior of the bird flocks and swarm fish. Indeed, PSO presents high quality solutions in a vast searching area and in a short time with a fast convergence compared to other random optimization techniques [38–42, 42–55]. As some recent notable applications of PSO on a wide range of controllers, the following research studies can be reviewed. Cheng et al. [56] designed the iterative learning control and iterative feedback tuning via formulating into one constrained optimization problem based upon particle swarm optimization and the stabilizing projection. The effectiveness of the method was provided via simulations on a four-room building control test-bed system. Vijay and Jena [42] proposed the control method for two-degrees-of-freedom rigid robot manipulator based upon the coupling of artificial neurofuzzy inference system as well as sliding mode control. Particle swarm optimization was utilized to tune the parameters of the sliding surface via minimizing quadratic performance indices. Ye et al. [41] utilized an enhanced particle swarm optimization algorithm to search for the optimal proportional-integral-derivative controller gains of the nonlinear hydraulic system. Furthermore, selection and crossover operators were proposed for the standard PSO algorithm. By comparing enhanced PSO, standard PSO, and Phase Margin tuning approaches, it was shown that the enhanced PSO algorithm provides a superior performance in PID control for positioning of nonlinear hydraulic system. Vinodh Kumar et al. [57] introduced an adaptive particle swarm optimization approach to gain the elements of Q and R matrices in optimal linear quadratic regulator control. Moreover, to augment the convergence speed and precision of the PSO, an adaptive inertia weight coefficient was proposed in the velocity equation of PSO. Sadeghpour et al. [58] proposed PSO-based multivariable control, where two or more control parameters were tuned concurrently either in a single or in multiple control inputs in order to stabilize the 1-cycle fixed points of the Logistic map, the Hénon map, and the chaotic Duffing system. Sedghizadeh and Beheshti [59] employed PSO based fuzzy gain scheduling approach to optimally update the subspace predictive control gains directly with no need to apply persistently excitation signals. Marinaki et al. [49] used PSO with a combination of continuous and discrete variables for the optimal design of the fuzzy controller, which is an appropriate tool for the systematic development of active control methods and can be properly tuned if no experience exists. Soon et al. [60] found the control parameters of the sliding mode control and PID control using PSO and compared the results with the outcomes of the Ziegler-Nichols algorithm which showed the superiority of the PSO algorithm. Wang et el. [61] proposed a novel fuzzy neural network sliding mode control approach for an overhead crane using PSO. The simulation results proved the proper performance of the controller. Jeyalakshmi and Subburaj [62] proposed the particle swarm optimization-based fuzzy logic controller design for the load frequency control in a two-area interconnected hydrothermal power system. In fact, effective control method was gained through using a combination of PSO and fuzzy logic technique. Al-Mamun and Zhu [63] introduced a fuzzy logic controller for steering control of a single wheel robot with the aid of fuzzy membership functions optimized using the PSO algorithm. Martinez et al. [64] utilized the PSO approach to gain the parameters of the membership functions of a type-2 fuzzy logic controller in order to minimize the state error for linear systems. Indeed, PSO was employed to gain the optimal Type-2 FLC to achieve regulation of the output and stability of the closed-loop system. Simulation results showed the feasibility of the proposed method. In the performance assessment on the designed PID sliding surface, the controller parameter is first obtained through conventional tuning method known as Ziegler-Nichols (ZN), which is then compared with the particle swarm optimization (PSO) computational tuning algorithm.

This paper introduced an inverse dynamics based optimal fuzzy Proportional Derivative (PD) controller for a RPP (Revolute-Prismatic-Prismatic) robot manipulator optimized via particle swarm optimization. The Denavit-Hartenberg and Lagrange approaches were utilized for deriving the kinematic and dynamic equations of the manipulator. In order to stabilize the links of the robot, the inverse dynamics approach and proportional-derivative controller were used in this research study. For the gain tuning of the designed controllers, the particle swarm optimization was successfully implemented. The obtained results of the optimal fuzzy PD controller based on the inverse dynamics were compared to the outcomes of the PD controller, and it was illustrated that the optimal fuzzy PD controller shows better controlling performance in comparison with other controllers.

The structure of the paper is as follows. The kinematics and dynamics of the RPP robot manipulator are presented in Section 2. Section 3 provides the control of the RPP robot manipulator. The optimization of the parameters using particle swarm optimization is discussed in Section 4. Section 5 presents results and discussion. Finally, Section 6 provides the conclusions and future work.

2. The Kinematics and Dynamics of the RPP Robot Manipulator

The RPP robot has three degrees of freedom, which involves one degree of the revolute joint and two degrees of prismatic joint (RPP), where the degree of freedom of the end effector is not considered. The most common method to address the direct kinematic problems is the Denavit-Hartenberg approach. In this method, each member is assigned a number from 0 to n, where the link 0 is a fixed base (ground) and

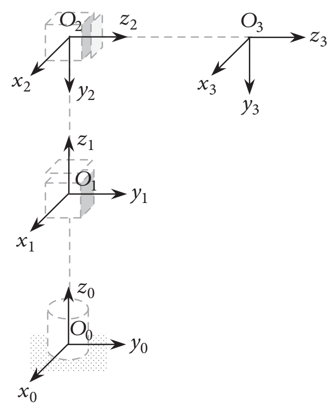

FIGURE 1: The position of the coordinate systems on each of the joints.

TABLE 1: The Denavit-Hartenberg parameters with regard to each joint.

Link	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	d_1	θ_1^*
2	0	-90	d_2^*	0
3	0	0	d_3^*	0

the final link is the end effector. It is assumed for this kind of robots that there exist $n-1$ links and n joints. For each joint, a separate coordinate system is defined with regard to the reference coordinate system (ground). The position of the i^{th} joint is shown with q_i. The position is considered in radian for the revolute joints and in meter for prismatic joints. Figure 1 illustrates how the coordinate system is placed on each joint. Moreover, the Denavit-Hartenberg parameters with regard to each joint are provided in Table 1. According to Figure 1, the coordinate system of $X_0Y_0Z_0$ is considered as the reference system.

The homogenous transformation matrix is gained through the required transformations and using (1)-(2), as illustrated in (3).

$$T_i^{i-1} = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \tag{1}$$

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$T = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & -d_3\sin\theta_1 \\ \sin\theta_1 & 0 & \cos\theta_1 & d_3\cos\theta_1 \\ 0 & -1 & 0 & d_1+d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Due to avoiding the computational complexity and providing simplification for nonlinear dynamic equations, the Lagrange method is employed in this study. In the Lagrange approach, the scalar quantities of the kinematic energy and potential energy are calculated and expressed with respect to the general coordinates. Finally, by utilizing the Lagrange approach according to (4), the dynamic equations of the system are obtained through (5).

$$L(q,\dot{q}) = K(q,\dot{q}) + U(q) \tag{4}$$

$$\tau = \frac{d}{dt}\left(\frac{\partial L(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial L(q,\dot{q})}{\partial q} \tag{5}$$

where L represents the Lagrange function, K denotes the kinematic energy, and U illustrates the potential energy. Moreover, q is the generalized vector of the coordinate of joints, $\dot{q}$ represents the generalized vector of the velocity of the joints, and τ shows the generalized vector of the torques and forces. Therefore, the dynamic equations of a robot with n links where there is no flexibility in interacting with other objects are obtained according to

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{6}$$

where $q \in R^n$ represents the position vector of joints, $D(q) \in R^{n\times n}$ is the inertia matrix of the robot, $C(q, \dot{q})$ is the Coriolis force vector, $G(q) \in R^n$ dentoes the vector of the gravitational torque, and $\tau \in R^n$ illustrates the vector of torques applied to the manipulator's joints.

By substituting each of the parameters into (6), the governing dynamic equations for each of the links are obtained as follows.

$$\begin{gathered} \left(d_3^2(m_1+m_2+m_3) + I_{yy} + 2\times I_{zz}\right)\ddot{\theta}_1 + 2 \\ \times \left(d_3(m_1+m_2+m_3)\right)\dot{\theta}_1\dot{d}_3 + \emptyset_1 = \tau_1 \end{gathered} \tag{7}$$

$$(m_2+m_3)\ddot{d}_2 + \emptyset_2 = f_2 \tag{8}$$

$$m_3\ddot{d}_3 - d_3(m_1+m_2+m_3)\dot{\theta}_1^2 + \emptyset_3 = f_3 \tag{9}$$

3. The Control of the RPP Robot Manipulator

The dynamics of the manipulator's arm is extremely nonlinear, and, hence, designing an efficient controller is a complicated task and of great importance. One of the appropriate approaches to enhance the tracking efficiency of the manipulator's arm is the control method of the computed torque. By using the computed torque control, the linear closed-loop equations are gained, and if the coefficients of the control approach are chosen properly, the stability of the controller would be guaranteed.

The required control torque, which is u considered as the calculated force produced by an actuator, is computed according to

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u \tag{10}$$

where $M(q)$ represents the inertia matrix of the robot and the motor.

The inverse dynamic control approach is a crucial method for the control of the mechanical arms. The inverse dynamic control rule is defined as follows.

$$u = M(q)a_q + C(q,\dot{q})\dot{q} + g(q) \tag{11}$$

where a_q is a function of q and its derivative. Furthermore, a_q is independent of the movement of other links and is only affected by the q of its own link.

$$a_q = -K_0 q - K_1\dot{q} + \mathrm{r} \tag{12}$$

The selected reference input is as follows:

$$r(t) = \ddot{q}^d(t) + K_0 q^d(t) + K_1\dot{q}^d(t) \tag{13}$$

The tracking error is defined as $e(t) = q - q^d$ which guarantees that the steady-state error equals zero.

$$\ddot{e}(t) + K_1\dot{e}(t) + K_0 e(t) = 0 \tag{14}$$

In order to address this problem straightforwardly, the coefficients of K_0 and K_1 are considered as the proportional coefficient K_p and the derivative coefficient K_D, respectively.

$$a_q = -K_P q - K_D\dot{q} + \mathrm{r} \tag{15}$$

The adjustment and determination of these design parameters are a key issue to design PD controllers. Hence, the fuzzy logic approach is applied to calculate the gains adaptively as follows.

$$\widehat{K}_P = K_P^b + K_P^r \Delta W_1 \tag{16}$$

$$\widehat{K}_D = K_D^b + K_D^r \Delta W_2 \tag{17}$$

where ΔW_1 and ΔW_2 are the fuzzy variables. K_P^b and K_D^b are base variables and K_P^r and K_D^r are regulation variables. The base and regulation variables can be obtained by the try and error process. However, one of the best solutions to find these to have an optimal controller is the use of the optimization approaches such as particle swarm optimization algorithm.

The fuzzy system selected for ΔW_1 and ΔW_2 has the following characteristics:

(I) The inference product engine

(II) The fuzzifier and the trapezoidal membership function in the beginning and end of the range and a triangle in the middle of the range

(III) The defuzzifier of the average of the centers

Therefore, the input membership function is selected as Figure 2 and Table 2.

The fuzzy rules of the system are adjusted via setting up the above-mentioned parameters in the MATLAB and using Table 2.

Results and discussions are conducted through the outcomes of the fuzzy PD controller and substituting the outcomes into the inverse dynamic equations of (11). The summation of the control gains of the PD controller illustrated in Table 3 and the control gains of the fuzzy system are considered as the control gains of the fuzzy PD controller.

Table 2: The fuzzy rules of the system.

X	Y_L
NL	1
Z	0.5
PL	1

Table 3: The assumed parameters of the PD controller.

link	K_{P_1}	K_{D_1}
1	50	**14**
2	30	**10**
3	40	**12**

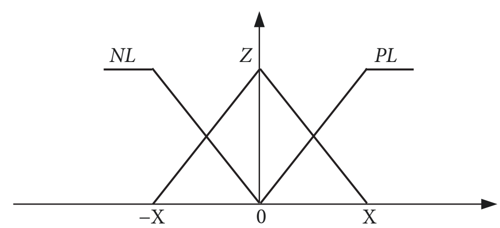

Figure 2: The input membership function.

Table 4 provides the information of the initial conditions of each of the variables and the desired values of each of the links.

In order to elaborate on the performance of the control approach, a smart optimization method is presented in the following section to optimize the control gains of the fuzzy PD controller.

4. The Optimization of the Parameters of the Proposed Controller

For the algorithm of the particle swarm optimization, each particle is evaluated among the whole population and if the termination condition is satisfied, the algorithm operation is stopped. However, if the termination condition is not satisfied, the position of each particle is evaluated with respect to its previous position and the position of the best particle among population. In fact, the position and velocity of the particle are being iteratively calculated until the termination condition of the algorithm is satisfied. Consider the general mathematical equations of the PSO as

$$\begin{aligned}\vec{v}_i(t+1) = {} & w\vec{v}_i(t) + C_1\vec{r}_1\left(\vec{x}_{pbest_i}(t) - \vec{x}_i(t)\right) \\ & + C_2\vec{r}_2\left(\vec{x}_{gbest}(t) - \vec{x}_i(t)\right)\end{aligned} \tag{18}$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{19}$$

where $\vec{x}_i(t)$ and $\vec{v}_i(t)$ describe the position and velocity of the particle i at certain iteration t, respectively. Moreover, C_1 and C_2 are coefficients that express the tendency of the particle to the personal and social successes, respectively, $\vec{r}_1$ and $\vec{r}_2$ are random vectors in interval [0,1], and w illustrates

Table 4: The initial conditions of the location and velocity and the desired values (q^d) of each of the links.

link	*Position*	*Velocity*	q^d
1	30°	**0**	$\mathbf{10^\circ}$
2	0.1 m	**0**	**0.5** $\boldsymbol{m}$
3	0.5 m	**0**	**1** $\boldsymbol{m}$

the effect of the particle velocity at the previous iteration on the next one. Furthermore, $\vec{x}_{pbest_i}(t)$ and $\vec{x}_{gbest}$ are personal and global best positions of the particle i and the swarm, respectively.

Two stages can be regarded in order to analyze the performance of this optimization algorithm on the control method of the robot manipulator. In the first stage, the gains of the PD controller are optimized. In the second stage, the gains of the fuzzy PD controller are optimized. The initial assumptions of this problem are as follows.

(1) The number of particles in the initial population is NP=10;

(2) The maximum number of iterations is MI=100;

(3) The impact rate of a particle from its local position is $C_1 = 2$;

(4) The impact rate of a particle from the general position of all particles is $C_2 = 2$;

(5) The inertia weight is $w = 1$;

(6) The objective function (OF) is considered as the integral of the summation of the absolute of the joint variables as follows.

$$OF = \int \left(\left|\theta_1\right| + \left|d_2\right| + \left|d_3\right|\right) dt \tag{20}$$

By substituting the conditions assumed for each of the links and defining the objective function, the results and analyses will be provided in the next section.

5. The Results and Analyses

In the fuzzy control, the heuristic fuzzy parameters (K^b_{Pi}, $K^r_{Pi}, K^b_{Di}, K^r_{Di}$, $i = 1, 2, 3$) are required to be chosen properly. Therefore, the particle swarm optimization is used to determine the proper parameters and to eliminate the tedious and repetitive trial-and-error process. Furthermore, the performance of a controlled closed loop system is usually evaluated by variety of goals. In this paper, the integral of the summation of the absolute of the joint variables is considered as the objective functions which have to be minimized. This means that, by selecting various values for the selective parameters, we can make changes in the objective function. Figure 3 depicts the convergence diagram of the PSO algorithm over the iteration number for the best particle and mean value of all particles. Moreover, the obtained design variables and objective function via this optimization process are stated in Table 5.

By regarding and comparing the position, force, and velocity diagrams of each of the links, the obtained results illustrate the superiority of the fuzzy PD controller over the PD controller. It is crucial to note that the input parameters of the controllers and the input coefficients are assumed in an ideal status in this study, and if there exist differences between

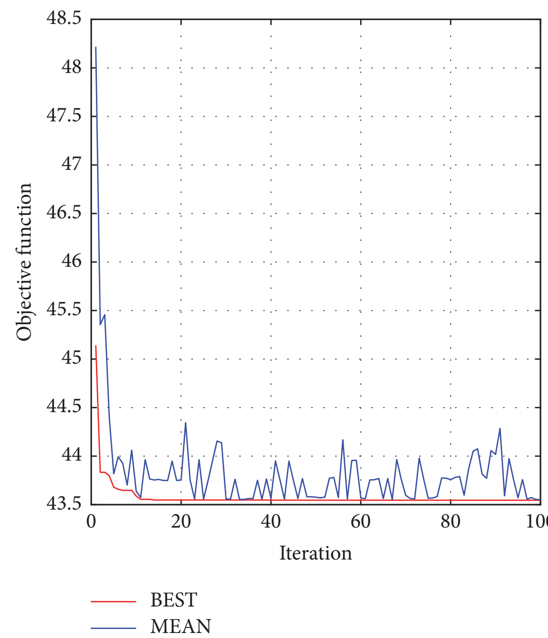

Figure 3: Convergence diagram of the PSO algorithm over the iteration number.

Table 5: Design variables and objective functions found by the particle swarm optimization algorithm.

Design variables	K^b_{P1}	**50.00**
	K^b_{P2}	**27.92**
	K^b_{P3}	**35.69**
	K^r_{P1}	**50.00**
	K^r_{P2}	**27.79**
	K^r_{P3}	**35.03**
	K^b_{D1}	**10.99**
	K^b_{D2}	**10.61**
	K^b_{D3}	**12.00**
	K^r_{D1}	**15.98**
	K^r_{D2}	**10.00**
	K^r_{D3}	**12.00**
Objective functions	$\int \left\|\theta_1 - \theta_{1d}\right\| dt$	**43.89**
	$\int \left\|d_2 - d_{2d}\right\| dt$	**0.11**
	$\int \left\|d_3 - d_{3d}\right\| dt$	**1.25**

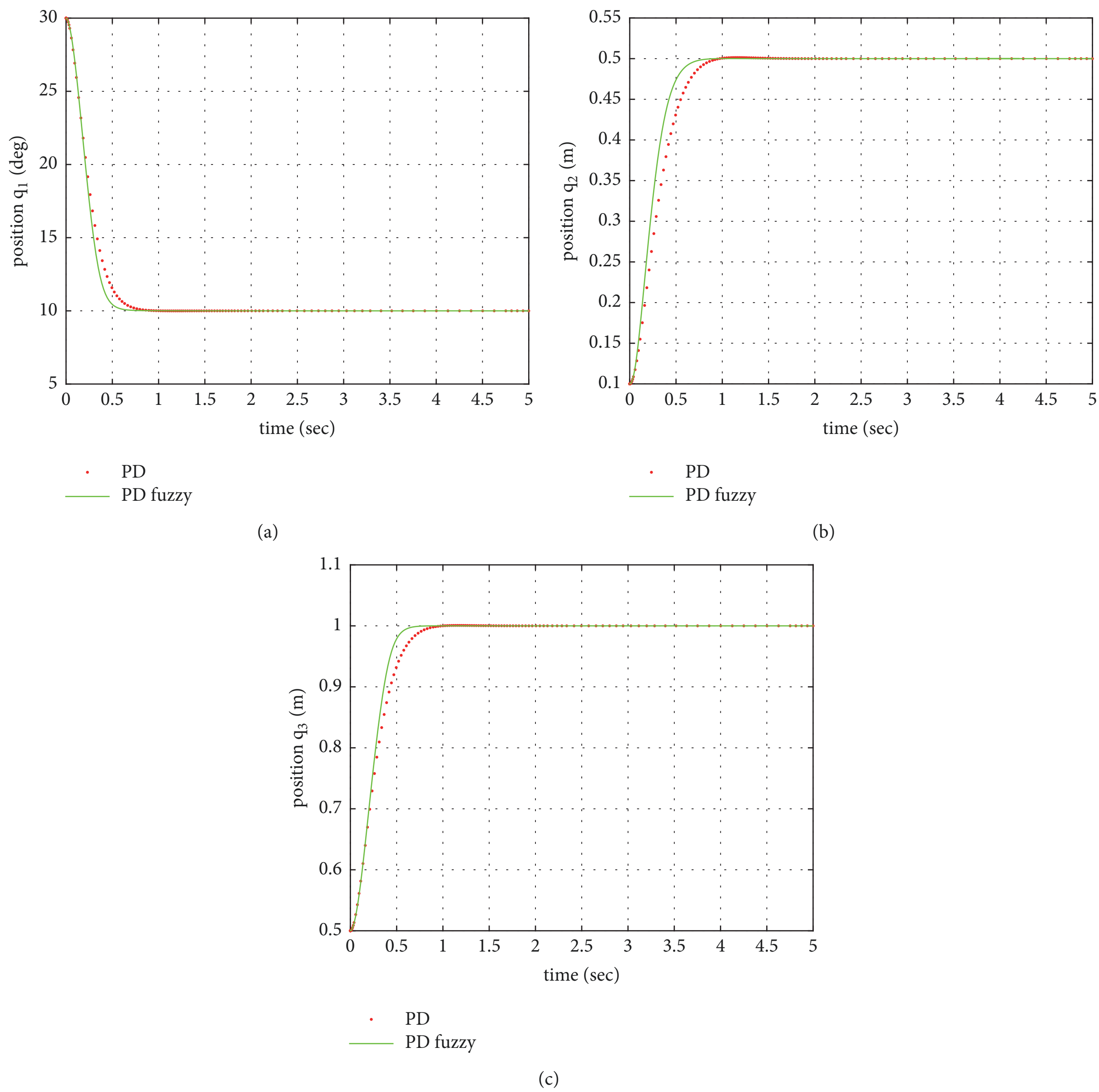

Figure 4: The position diagram of each of the links controlled via optimal fuzzy PD and optimal PD controllers, (a) the position of the first link (revolute), (b) the position of the second link (prismatic), and (c) the position of the third link (prismatic).

the real model of the manipulator and its assumed model, it is probable that these controllers show error or instability. First, the position diagrams of each of these controllers are put under analysis.

By comparing the graphs of Figure 4, it can be concluded that the optimal fuzzy PD controller provides a better performance compared to the optimal PD controller by presenting a lower time for achieving the desired position. In this respect, in order to provide a better understanding of the performance of the controllers, the graphs of the controlling force of each of the actuators are compared for the optimal fuzzy PD controller and the optimal PD controller.

As it can be found from Figure 5, the optimal fuzzy PD controller requires nearly the same controlling force as the optimal PD controller; however, the optimal fuzzy PD controller provides a lower time for reaching the desired position compared to the optimal PD controller.

By comparing the graphs of the controlling force for each of the links, it can be found that the forces of the actuators in both cases have some limits, which illustrates the appropriate performance of the fuzzy PD controller. Further, by using the fuzzy PID controller, the desired state is reached in less time compared to the PD controller.

By comparing the velocity diagrams of each of the links (Figure 6), it can be understood that the velocities of the links

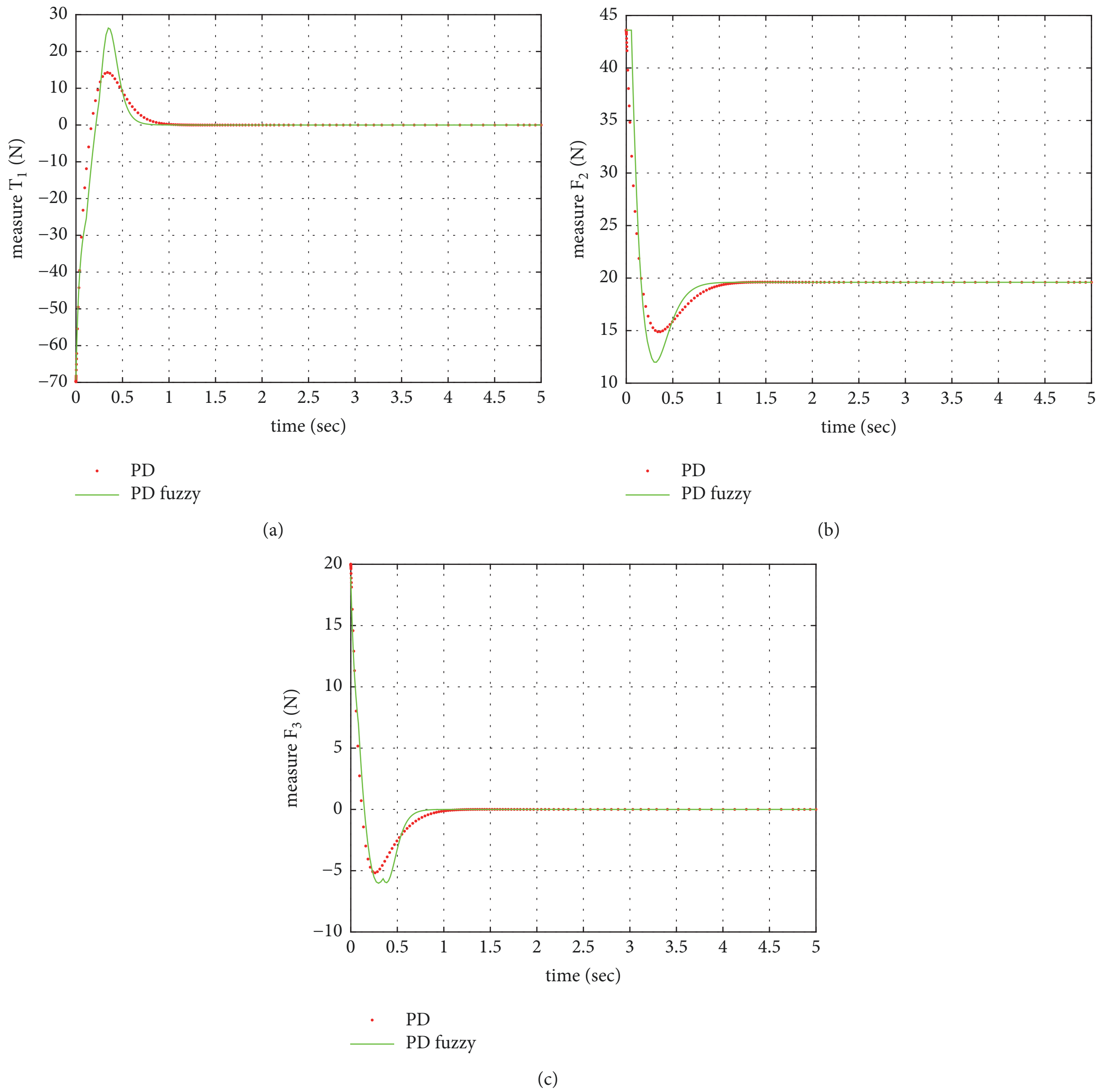

FIGURE 5: The comparison of the controlling force for optimal fuzzy PD and optimal PD controllers, (a) the controlling force of the first link (revolute), (b) the controlling force of the second link (prismatic), and (c) the controlling force of the third link (prismatic).

are higher when using the fuzzy PD controller in contrast to the PD controller. Therefore, the proper performance of the fuzzy PD controller is guaranteed by reaching the desired state in less time than the PD controller.

6. Conclusions and Future Work

This research study presented an optimal fuzzy PD controller for a RPP robot manipulator based on particle swarm optimization and inverse dynamics. Indeed, the Denavit-Hartenberg method and the Jacobi approach for each of the arms of the robot were utilized to obtain its kinematic equations. Moreover, the Lagrange approach was used to gain the dynamic equations of the manipulator. By comparing the results, it was found that the fuzzy PD controller provides a lower settling time and a near-zero steady state error, which resulted in the superiority of the fuzzy PD controller over the PD controller.

The use of classic controllers for the control of the smart autonomous vehicles requires complete knowledge of all the forces and torque. These forces and torque are required to obtain the differential reciprocal and rotational movements. In fact, to obtain these equations, it is needed to address complex equations in a high amount of time or conduct experiments to gain the coefficients.

Future Works. (1) By combining the robust control approach and computational torque, it is feasible to eliminate the weakness of unrobustness in this type of controllers. (2)

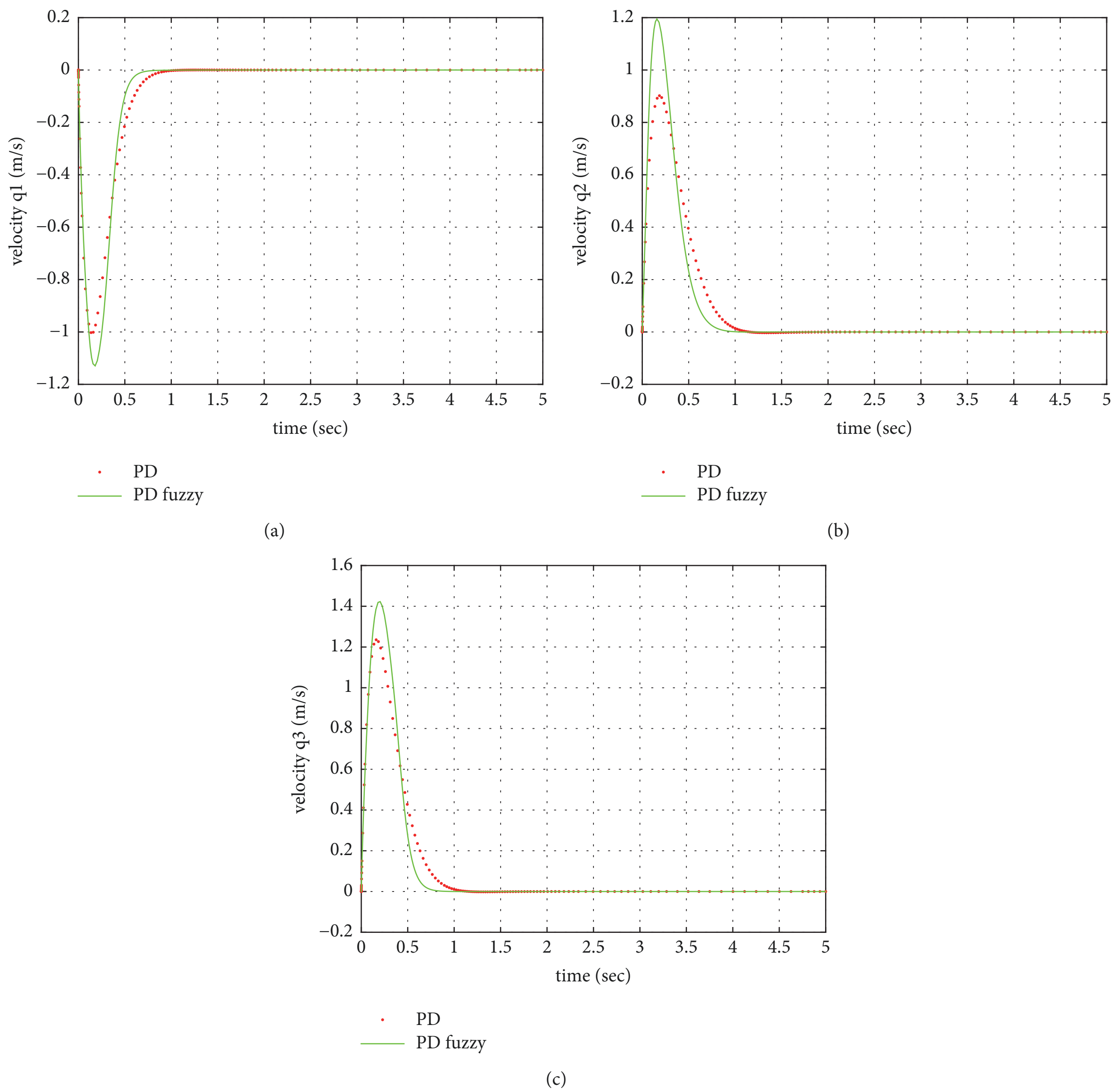

Figure 6: The comparison of the velocity diagrams, (a) the velocity of the first link (revolute), (b) the velocity of the second link (prismatic), and (c) the velocity of the third link (prismatic).

By utilizing a PID controller, it is possible to diminish the disturbances within the system and make them fairly zero. (3) By using a fuzzy controller which is independent of the inverse dynamic, it is feasible to receive the output feedback in every instant and so adjust the inputs of the system based upon that.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Ahmad and P. Plapper, "Safe and Automated assembly process using vision assisted robot manipulator," in *Proceedings of the 48th CIRP International Conference on Manufacturing Systems, CIRP CMS 2015*, pp. 771–776, Italy, June 2015.

[2] Q. Yu, G. Wang, X. Hua et al., "Base position optimization for mobile painting robot manipulators with multiple constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 56–64, 2018.

[3] M. V. Andulkar, S. S. Chiddarwar, and A. S. Marathe, "Novel integrated offline trajectory generation approach for robot assisted spray painting operation," *Journal of Manufacturing Systems*, vol. 37, pp. 201–216, 2015.

[4] D. Antonelli and S. Astanin, "Qualification of a collaborative human-robot welding cell," in *Proceedings of the 48th CIRP*

International Conference on Manufacturing Systems, CIRP CMS 2015, pp. 352–357, Italy, June 2015.

[5] I. Staretu, "Structural synthesis, work spaces and direct kinematic of the one serial kinematic chain with 8 axes for industrial robots," *Procedia Technology*, vol. 19, pp. 207–214, 2015.

[6] J. Fu, F. Gao, W. Chen, Y. Pan, and R. Lin, "Kinematic accuracy research of a novel six-degree-of-freedom parallel robot with three legs," *Mechanism and Machine Theory*, vol. 102, pp. 86–102, 2016.

[7] A. Morell, M. Tarokh, and L. Acosta, "Solving the forward kinematics problem in parallel robots using Support Vector Regression," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 7, pp. 1698–1706, 2013.

[8] P. K. Jamwal, S. Q. Xie, Y. H. Tsoi, and K. C. Aw, "Forward kinematics modelling of a parallel ankle rehabilitation robot using modified fuzzy inference," *Mechanism and Machine Theory*, vol. 45, no. 11, pp. 1537–1554, 2010.

[9] J. Zhao, Z. Feng, F. Chu, and N. Ma, "Kinematics with four points cartesian coordinates for spatial parallel manipulator," *Advanced Theory of Constraint and Motion Analysis for Robot Mechanisms*, vol. 10, pp. 235–262, 2014.

[10] M. Pfurner, "Closed form inverse kinematics solution for a redundant anthropomorphic robot arm," *Computer Aided Geometric Design*, vol. 47, pp. 163–171, 2016.

[11] A. T. Hasan, A. M. S. Hamouda, N. Ismail, and H. M. A. A. Al-Assadi, "An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator," *Advances in Engineering Software*, vol. 37, no. 7, pp. 432–438, 2006.

[12] R. Falconi, R. Grandi, and C. Melchiorri, "Inverse kinematics of serial manipulators in cluttered environments using a new paradigm of particle swarm optimization," in *Proceedings of the 19th IFAC World Congress on International Federation of Automatic Control, IFAC 2014*, pp. 8475–8480, South Africa, August 2014.

[13] H. Dong, Z. Du, and G. S. Chirikjian, "Workspace density and inverse kinematics for planar serial revolute manipulators," *Mechanism and Machine Theory*, vol. 70, pp. 508–522, 2013.

[14] N. Rokbani and A. M. Alimi, "Inverse kinematics using particle swarm optimization, A statistical analysis," *Procedia Engineering*, vol. 64, pp. 1602–1611, 2013.

[15] P. Constantin, "Lagrangian-Eulerian methods for uniqueness in hydrodynamic systems," *Advances in Mathematics*, vol. 278, pp. 67–102, 2015.

[16] X. Liu, H. Li, J. Wang, and G. Cai, "Dynamics analysis of flexible space robot with joint friction," *Aerospace Science and Technology*, vol. 47, pp. 164–176, 2015.

[17] P. Huang, Z. Hu, and Z. Meng, "Coupling dynamics modelling and optimal coordinated control of tethered space robot," *Aerospace Science and Technology*, vol. 41, pp. 36–46, 2015.

[18] S. Yamaguchi, K. Tsutsui, K. Satake, S. Morikawa, Y. Shirai, and H. T. Tanaka, "Dynamic analysis of a needle insertion for soft materials: Arbitrary Lagrangian-Eulerian-based three-dimensional finite element analysis," *Computers in Biology and Medicine*, vol. 53, pp. 42–47, 2014.

[19] S. González-Vázquez and J. Moreno-Valenzuela, "Time-scale separation of a class of robust PD-type tracking controllers for robot manipulators," *ISA Transactions®*, vol. 52, no. 3, pp. 418–428, 2013.

[20] S. Yamacli and H. Canbolat, "Simulation of a SCARA robot with PD and learning controllers," *Simulation Modelling Practice and Theory*, vol. 16, no. 9, pp. 1477–1487, 2008.

[21] C. Q. Huang, L. F. Xie, and Y. L. Liu, "PD plus error-dependent integral nonlinear controllers for robot manipulators with an uncertain Jacobian matrix," *ISA Transactions®*, vol. 51, no. 6, pp. 792–800, 2012.

[22] F. Reyes and A. Rosado, "Polynomial family of PD-type controllers for robot manipulators," *Control Engineering Practice*, vol. 13, no. 4, pp. 441–450, 2005.

[23] X. Xiang, D. Chen, C. Yu, and L. Ma, "Coordinated 3D path following for autonomous underwater vehicles via classic PID controller," in *Proceedings of the 3rd IFAC Conference on Intelligent Control and Automation Science, ICONS 2013*, pp. 327–332, China, September 2013.

[24] S. El Ferik, M. Tariq Nasir, and U. Baroudi, "A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space," *Applied Soft Computing*, vol. 44, pp. 117–127, 2016.

[25] H. R. Hassanzadeh, M.-R. Akbarzadeh-T, A. Akbarzadeh, and A. Rezaei, "An interval-valued fuzzy controller for complex dynamical systems with application to a 3-PSP parallel robot," *Fuzzy Sets and Systems*, vol. 235, pp. 83–100, 2014.

[26] H. Wang and Y. Xie, "Adaptive inverse dynamics control of robots with uncertain kinematics and dynamics," *Automatica*, vol. 45, no. 9, pp. 2114–2119, 2009.

[27] Y. Singh and M. Santhakumar, "Inverse dynamics and robust sliding mode control of a planar parallel (2-PRP and 1-PPR) robot augmented with a nonlinear disturbance observer," *Mechanism and Machine Theory*, vol. 92, pp. 29–50, 2015.

[28] E. Carrera and M. A. Serna, "Inverse dynamics of flexible robots," *Mathematics and Computers in Simulation*, vol. 41, no. 5-6, pp. 485–508, 1996.

[29] Y. Singh, V. Vinoth, Y. R. Kiran, J. K. Mohanta, and S. Mohan, "Inverse dynamics and control of a 3-DOF planar parallel (U-shaped 3-PPR) manipulator," *Robotics and Computer-Integrated Manufacturing*, vol. 34, pp. 164–179, 2015.

[30] S. Staicu, "Inverse dynamics of the 3-PRR planar parallel robot," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 556–563, 2009.

[31] Z. Bingül and O. Karahan, "A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control," *Expert Systems with Applications*, vol. 38, no. 1, pp. 1017–1031, 2011.

[32] S. Bouallègue, J. Haggège, M. Ayadi, and M. Benrejeb, "PID-type fuzzy logic controller tuning based on particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 484–493, 2012.

[33] R. Martínez-Soto, O. Castillo, and L. T. Aguilar, "Type-1 and Type-2 fuzzy logic controller design using a Hybrid PSO-GA optimization method," *Information Sciences*, vol. 285, pp. 35–49, 2014.

[34] A. B. Cara, C. Wagner, H. Hagras, H. Pomares, and I. Rojas, "Multiobjective optimization and comparison of nonsingleton type-1 and singleton interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 459–476, 2013.

[35] M. Biglarbegian, W. W. Melek, and J. M. Mendel, "Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 4, pp. 1371–1384, 2011.

[36] R. Hosseini, S. D. Qanadli, S. Barman, M. Mazinani, T. Ellis, and J. Dehmeshki, "An automatic approach for learning and tuning gaussian interval type-2 fuzzy membership functions applied to lung CAD classification system," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 224–234, 2012.

[37] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.

[38] G. Shankar and V. Mukherjee, “MPP detection of a partially shaded PV array by continuous GA and hybrid PSO,” *Ain Shams Engineering Journal*, vol. 6, no. 2, pp. 471–479, 2015.

[39] E. Assareh, M. A. Behrang, M. R. Assari, and A. Ghanbarzadeh, “Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran,” *Energy*, vol. 35, no. 12, pp. 5223–5229, 2010.

[40] F. Khoshahval, H. Minuchehr, and A. Zolfaghari, “Performance evaluation of PSO and GA in PWR core loading pattern optimization,” *Nuclear Engineering and Design*, vol. 241, no. 3, pp. 799–808, 2011.

[41] Y. Ye, C.-B. Yin, Y. Gong, and J.-J. Zhou, “Position control of nonlinear hydraulic system using an improved PSO based PID controller,” *Mechanical Systems and Signal Processing*, vol. 83, pp. 241–259, 2017.

[42] M. Vijay and D. Jena, “PSO based neuro fuzzy sliding mode control for a robot manipulator,” *Journal of Electrical Systems and Information Technology*, vol. 4, no. 1, pp. 243–256, 2017.

[43] E. Vinodh Kumar, G. S. Raaja, and J. Jerome, “Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter,” *Applied Soft Computing*, vol. 41, pp. 77–90, 2016.

[44] W. Zhang, J. Liu, L. Fan, Y. Liu, and D. Ma, “Control strategy PSO,” *Applied Soft Computing*, vol. 38, pp. 75–86, 2016.

[45] Y. Soufi, M. Bechouat, and S. Kahla, “Fuzzy-PSO controller design for maximum power point tracking in photovoltaic system,” *International Journal of Hydrogen Energy*, vol. 42, no. 13, pp. 8680–8688, 2017.

[46] H. E. A. Ibrahim, F. N. Hassan, and A. O. Shomer, “Optimal PID control of a brushless DC motor using PSO and BF techniques,” *Ain Shams Engineering Journal*, vol. 5, no. 2, pp. 391–398, 2014.

[47] P. D. Darade, B. Kumar, and S. Bhopale, “Displacement control of photostrictive actuators for vibration control with the aid of fuzzy logic control along with PSO algorithm,” in *Proceedings of the Materials Today*, vol. 4, pp. 3642–3651, 2017.

[48] C.-L. Chan and C.-L. Chen, “A cautious PSO with conditional random,” *Expert Systems with Applications*, vol. 42, no. 8, pp. 4120–4125, 2015.

[49] M. Marinaki, Y. Marinakis, and G. E. Stavroulakis, “Fuzzy control optimized by PSO for vibration suppression of beams,” *Control Engineering Practice*, vol. 18, no. 6, pp. 618–629, 2010.

[50] C. Zhang, M. Wu, and L. Luan, “An optimal PSO distributed precoding algorithm in QRD-based multi-relay system,” *Future Generation Computer Systems*, vol. 29, no. 1, pp. 107–113, 2013.

[51] M. Najjari and R. Guilbault, “Formula derived from particle swarm optimization (PSO) for optimum design of cylindrical roller profile under EHL regime,” *Mechanism and Machine Theory*, vol. 90, pp. 162–174, 2015.

[52] W. Tuvayanond and M. Parnichkun, “Position control of a pneumatic surgical robot using PSO based 2-DOF H∞ loop shaping structured controller,” *Mechatronics*, vol. 43, pp. 40–55, 2017.

[53] M. J. Mahmoodabadi, M. Taherkhorsandi, and A. Bagheri, “Optimal robust sliding mode tracking control of a biped robot based on ingenious multi-objective PSO,” *Neurocomputing*, vol. 124, pp. 194–209, 2014.

[54] Q. Chai and W. Wang, “A computational method for free terminal time optimal control problem governed by nonlinear time delayed systems,” *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 53, pp. 242–250, 2018.

[55] H. Shayeghi, A. Jalili, and H. A. Shayanfar, “Multi-stage fuzzy load frequency control using PSO,” *Energy Conversion and Management*, vol. 49, no. 10, pp. 2570–2580, 2008.

[56] C. Peng, L. Sun, and M. Tomizuka, “Constrained iterative learning control with PSO-youla feedback tuning for building temperature control,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3135–3141, 2017.

[57] E. V. Kumar, G. S. Raaja, and J. Jerome, “Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter,” *Applied Soft Computing*, vol. 41, pp. 77–90, 2016.

[58] M. Sadeghpour, H. Salarieh, G. Vossoughi, and A. Alasty, “Multi-variable control of chaos using PSO-based minimum entropy control,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 6, pp. 2397–2404, 2011.

[59] S. Sedghizadeh and S. Beheshti, “Particle swarm optimization based fuzzy gain scheduled subspace predictive control,” *Engineering Applications of Artificial Intelligence*, vol. 67, pp. 331–344, 2018.

[60] C. C. Soon, R. Ghazali, H. I. Jaafar, and S. Y. S. Hussien, “Sliding mode controller design with optimized PID sliding surface using particle swarm algorithm,” *Procedia Computer Science*, vol. 105, pp. 235–239, 2017.

[61] Z. Wang, Z. Chen, and J. Zhang, “On PSO based fuzzy neural network sliding mode control for overhead crane,” in *Practical Applications of Intelligent Systems*, Y. Wang and T. Li, Eds., vol. 124 of *Advances in Intelligent and Soft Computing*, pp. 563–572, Berlin, Germany, 2012.

[62] V. Jeyalakshmi and P. Subburaj, “PSO-scaled fuzzy logic to load frequency control in hydrothermal power system,” *Soft Computing*, vol. 20, no. 7, pp. 2577–2594, 2016.

[63] A. Al-Mamun and Z. Zhu, “PSO-optimized fuzzy logic controller for a single wheel robot,” in *Trends in Intelligent Robotics*, P. Vadakkepat, Ed., vol. 103 of *Communications in Computer and Information Science*, pp. 330–337, Berlin, Germany, 2010.

[64] R. Martinez, O. Castillo, L. T. Aguilar, and A. Rodriguez, “Optimization of type-2 fuzzy logic controllers using PSO applied to linear plants,” in *Soft Computing for Intelligent Control and Mobile Robotics*, O. Castillo, J. Kacprzyk, and W. Pedrycz, Eds., vol. 318 of *Studies in Computational Intelligence*, pp. 181–193, Berlin, Germany, 2011.