*J*ournal of ***AI*** and ***D****ata* ***M****ining*
Vol. 2, No. 2, 2014, 125-133.

Designing an adaptive fuzzy control for robot manipulators using PSO

F. Soleiman Nouri*, M. Haddad Zarif and M. M. Fateh

Department of Electrical Engineering and Robotics, University of Shahrood, Iran.

Received 10 December 2013; accepted 28 June 2014
**Corresponding author: solaimannourifatemeh@yahoo.com (F. Soleiman Nouri).*

Abstract
This paper presents a designing an optimal adaptive controller for tracking down the control of robot manipulators based on particle swarm optimization (PSO) algorithm. PSO algorithm has been used to optimize parameters of the controller and hence to minimize the integral square of errors (ISE) as a performance criteria. In this paper, an improved PSO using a logic is proposed to increase the convergence speed. In this case, the performance of PSO algorithms such as an improved PSO (IPSO), an improved PSO using fuzzy logic (F-PSO), a linearly decreasing inertia weight of PSO (LWD-PSO) and a nonlinearly decreasing inertia weight of PSO (NDW-PSO) are with parameter accuracy and convergence speed. As a result, the simulation results show that the F-PSO approach presents a better performance in the tracking down the control of robot manipulators than other algorithms.

Keywords: *Particle Swarm Optimization (PSO), Robot Manipulators, Adaptive Controller, Improved PSO Using Fuzzy Logic (F-PSO), Integral Square of Errors (ISE).*

1. Introduction

Robot manipulators are multi-input/multi-output (MIMO) nonlinear system with couplings that have to face many structured and unstructured uncertainties such as payload parameter, un-modeled dynamics, external disturbance and friction. The design robust controller for robot manipulators and their application is one of the considerable topics in a control field; so many control techniques have been proffered to control robot manipulator such as the PID control method [1], adaptive control [2,3], combined adaptive sliding mode controllers [4], optimal control [5,6] and intelligent approaches [7].
The PSO algorithm comprises a simple structure, and it is easy to be implemented, independent from initial guess and does not need any objective function's gradient. Due to the good characteristics of this algorithm, it has been applied in the diversity of investigation field. For instance, in [9-11], PSO is presented to setting the optimal parameter of PID controller. In [12], proposed to use PSO and its application to train weights of artificial neural network. In [13], the author employed the PSO algorithm to optimize the parameter of tracking a controller. In [14], PSO is proffered to solve the systems of nonlinear equations. In [15], the proposed algorithm has been used to solve nonlinear optimal control. In [16], the PSO algorithm is used to optimize the parameters of controller to position/force control of constrained robot manipulators.
Fuzzy logic is based on fuzzy set theory. A fuzzy logic controller is composed of its rule base and membership function. Fuzzy logic system was used to approximate any nonlinear function [22,23].
In this paper, the particle swarm optimization utilized to drive the optimal parameters of adaptive controller for robot manipulators. The performance of an improved PSO using fuzzy logic (F-PSO) is compared with PSO with linearly decreasing inertia weight (LDW-PSO), nonlinear inertia weight PSO (NDW-PSO) and improved PSO (IPSO). The simulation results confirmed that the F-PSO has better performance than other algorithm mentioned above. The rest of paper is organized as follows: Section 2 presents the mathematical description of robot manipulator.

Section 3 illustrates the particle swarm optimization. Section 4 shows the design of controller parameters based on PSO. Section 5 illustrates the simulation results on a robot manipulator and comparisons between algorithms. Section 6 concludes the paper.

2. Dynamics of robot manipulators

In the absence of friction or other disturbance, the dynamic equation of a multi-input/multi-output robot manipulator system can be written as [2, 4]:

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

Where q is a $n\times 1$ vector of generalized coordinate, the position vector of a robot manipulator. $\dot{q}$ is a $n\times 1$ vector of first derivative of generalized coordinate, the velocity of a robot manipulator. $\ddot{q}$ is a $n\times 1$ vector of second derivative of generalized coordinate, the acceleration of a robot manipulator. $M(q)$ is a $n\times n$ symmetric positive definite matrix of manipulator inertia. $C(q,\dot{q})$ is a $n\times 1$ vector of centrifugal and coriolis torque. $G(q)$ is a $n\times 1$ vector of gravitational torque. τ is a $n\times 1$ vector of generalized control input torque or force. The (1) can be stated as follows [2]:

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=Y(q,\dot{q},\ddot{q})\beta=\tau \qquad (2)$$

Where $Y(q,\dot{q},\ddot{q})$ is a $n\times p$ matrix called regressor. β is a $p\times 1$ uncertain vector .

A number of useful properties of robot dynamic is expressed as follows [8]:

Property 1. An appropriate definition of coriolis and centrifugal matrix makes that the $N(q,\dot{q})=M(q)-2C(q,\dot{q})$ is skew symmetric. This property is very important to stability analysis.

Property 2. The $M(q)$ is a symmetric positive definite matrix, such that:

$$0<\mu_1 I\le M(q)\le \mu_2 I$$

μ_1,μ_2 are positive constant and I is the identity matrix.

2.1. Adaptive controller design

The control law has been given as follows [2]:

$$\tau=M(q)(\ddot{q}_d-\Lambda(\dot{q}-\dot{q}_d))+ C(q,\dot{q})(\dot{q}_d-\Lambda(q-q_d))+G(q)+K\sigma \qquad (3)$$

Where k is a definite positive matrix, σ is an error of velocity.

$\tilde{q}$, $\dot{\tilde{q}}$, $\ddot{q}_r$, $\dot{q}_r$ are defined as:

$$\tilde{q}=q-q_d\,,\ \dot{\tilde{q}}=\dot{q}-\dot{q}_d\,,\ \dot{q}_r=\dot{q}_d-\Lambda\tilde{q}\,,\quad \ddot{q}_r=\ddot{q}_d-\Lambda\dot{\tilde{q}} \qquad (4)$$

Where $\tilde{q}$ indicates the position tracking error, $\dot{\tilde{q}}$ represents the velocity, $\dot{q}_r$ is called reference Velocity that is utilized to guarantee the convergence of the tracking error, $\ddot{q}_r$ is the reference acceleration, Λ is a positive definite matrix and σ is obtained as:

$$\sigma=\dot{q}_r-\dot{q}=\dot{\tilde{q}}+\Lambda\tilde{q} \qquad (5)$$

In the presence of uncertainties, a control law is proposed as:

$$\tau=\hat{M}(q)\ddot{q}_r+\hat{C}(q,\dot{q})\dot{q}_r+\hat{G}(q)+K\sigma =Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\hat{\beta}+K\sigma \qquad (6)$$

Where $\hat{M}(q)$ is the estimate of the $M(q)$, $\hat{C}(q,\dot{q})$ is the estimate of the $C(q,\dot{q})$, $\hat{G}(q)$ presented the estimate of the $G(q)$ and also $\hat{\beta}$ denoted the estimate of the β.

Attention to replace the recent control law in the (2), so modeling errors consists of:

$$\tilde{M}=\hat{M}-M\quad \tilde{C}=\hat{C}-C\quad \tilde{G}=\hat{G}-G \qquad (7)$$

In order to analysis the stability of the system and obtain convergence tracking error, the Lyapunov function candidate is suggested as follows:

$$v(t)=\frac{1}{2}\left[\sigma^{\mathrm{T}}\mathrm{H}\sigma+\tilde{\beta}^{\mathrm{T}}\Gamma^{-1}\tilde{\beta}\right] \qquad (8)$$

The adaptation law can be expressed as:

$$\dot{\hat{\beta}} = -\Gamma Y^{T}\sigma \qquad (9)$$

Using this upper equation, the derivative of $v(t)$ is given as:

$$\dot{v}(t) = -\sigma^{T}K_{D}\sigma \le 0 \qquad (10)$$

3. Particle swarm optimization

Particle swarm optimization algorithm is a stochastic evolutionary computation approach. It is inspired by the social behavior such as a flock of bird or a school of fish. This algorithm introduced by Eberhart and Kennedy in 1995 [17]. PSO contains a group of solutions that called particles.

These particles are moved in and evaluates the cost function of its position that has been placed in space. Particle adjusted its movement based on corresponding experience of particle and associated experiences of particle that led to the particle moves in the direction of better solution [15]. At each iteration, each particle for updating its velocity and position utilized equations in the following order:

$$V_i^{k+1} = wV_i^{k} + c_1 rand_1 \times (\text{Pbest}_i^k - X_i^k) + c_2\, rand_2 \times (\text{Gbest}^k - X_i^k) \qquad (11)$$

$$X_i^{k+1} = X_i^k + T_s V_i^{k+1} \qquad (12)$$

Where X_i^k is the current position of i^{th} particle at the k^{th} iteration. T_s is the sampling period. V_i^k is the Current velocity of i^{th} particle at the k^{th} iteration. w is the inertia weight which acquires an important task in the PSO convergence behavior since it is used to balance the global and local search ability. c_1, c_2 are positive constants, correspond to cognitive and social parameter respectively, called learning factors. $rand_1, rand_2$ are random numbers with uniform distribution in the range of 0 to 1. Pbest_i^k is the best position of i^{th} particle at the k^{th} iteration called as personal best. Gbest^k is the global best position among all the particles in the swarm at the k^{th} iteration called global best. The algorithm is repeated several times until the pause condition such as number of iteration or sufficiently good fitness [15].

PSO does exhibit some shortages. It may convergence to a local minimum, therefore researchers try to improve the performance of the PSO with different settings, e.g. w , C_1, C_2 [15].

In this work, we employed the IPSO, NDW-PSO, LDW-PSO and F-PSO, they are approaches that improved the performance of PSO and finally, F-PSO algorithm is compared with the other algorithms.

3.1. Linearly decreasing inertia weight PSO

Linearly decreasing inertia weight PSO was abbreviated to LDW-PSO, the inertia weight decreases linearly from $w_{\max}$ to $w_{\min}$, the equation is used for adapting the inertia weight in PSO as follows [19, 20]:

$$w^t = w_{\min} + \frac{iter_{\max} - t}{iter_{\max}}.(w_{\max} - w_{\min}) \qquad (13)$$

$iter_{\max}$ Denotes to maximum number of iteration and t denotes to current of iteration.

3.2. Nonlinear inertia weight PSO

Nonlinear inertia weight PSO was abbreviated to NDW-PSO. In this mechanism, the inertia weight decreases as same pervious approach but nonlinearity [18].

$$w^t = w_{\min} + \left(\frac{iter_{\max} - t}{iter_{\max}}\right)^n.(w_{\max} - w_{\min}) \qquad (14)$$

3.3. Improved PSO

The values of w, c_1, c_2 is very important to ensure convergent behavior and to optimally trade-off exploration and exploitation. In [21], Author used an improved PSO as follows:

$$w^t = 1/\left(1 + \exp(-\alpha F(\text{gbest}^t))^n\right) \qquad (15)$$

$$c_i = 1/\left(1 + \exp(-\alpha F(\text{gbest}^t))^n\right) \qquad (16)$$

$$\alpha = 1/\text{F}\left(gbest^t\right) \qquad (17)$$

This adaptation appliance changes in conformity to the rate of the global best fitness improvement.

3.4. Particle swarm optimization with using fuzzy

Fuzzy is used for designing and modeling for system that need to advance mathematics and probabilities. The important part of fuzzy system was a knowledge base that is comprised fuzzy IF-THEN rules. Fuzzy is used to improve the performance of PSO. A fuzzy system will be employed to adjust the learning factors c_1, c_2 with best fitness and iteration. The best fitness measure the performance of the best solution found so far. To design a fuzzy-PSO need to have ranges of best fitness and iteration. Therefore, the best fitness and iteration have to normalize into $\left[0,1\right]$ that defined as follows [22, 23]:

$$NCBPE = \frac{CBPE \_CBPE_{\min}}{CBPE_{\max} - CBPE_{\min}} \tag{18}$$

Where $CBPE$ is the current fitness value, $CBPE_{\min}$ is the best fitness value and $CBPE_{\max}$ is the worst fitness value.

$$Iteration = \frac{iteration}{iteration_{\max}} \tag{19}$$

In this mechanism, the best fitness and iteration are inputs and c_1, c_2 are outputs in the fuzzy system. The c_1, c_2 obtained from fuzzy were used to PSO and for adjusting w , we employed the IPSO that mentioned in [15]:

$$w^t = 1/\left(1+\exp(-\alpha F(\text{gbest}^t))^n\right) \tag{20}$$

$$\alpha = 1/\mathrm{F}\left(gbest^t\right) \tag{21}$$

We suggest fuzzy rules:

1. If (iteration is low) and (CPBE is low) then (c1 is low)(c2 is high)
2. If (iteration is low) and (CPBE is medium) then (c1 is medium low)(c2 is medium high)
3. If (iteration is low) and (CPBE is high) then (c1 is medium)(c2 is medium)
4. If (iteration is medium) and (CPBE is low) then (c1 is medium low)(c2 is high)
5. If (iteration is medium) and (CPBE is medium) then (c1 is medium)(c2 is high)
6. If (iteration is medium) and (CPBE is high) then (c1 is medium high)(c2 is low)
7. If (iteration is high) and (CPBE is low) then (c1 is high)(c2 is low)
8. If (iteration is high) and (CPBE is medium) then (c1 is medium high)(c2 is medium low)
9. If (iteration is high) and (CPBE is high) then (c1 is low)(c2 is medium low)

For designing the rules of fuzzy system, it was decided that in early iterations the PSO algorithm must explore and finally exploit.

These approaches usually start with large inertia values, which decrease over time to smaller values. Large values for w facilitate exploration, with increased diversity. A small w promotes local exploitation.

4. PSO controller tuning

The parameters of adaptive control law such as Γ_1 , Γ_2 , Γ_3 , Γ_4 , Λ_1 and k_1 is found using PSO.

All the parameters of controller are adjusted to minimize the fitness function based on the integral square of errors that is defined as follows:

$$f = \int_0^T \sum_{i=1}^{2} e_i(\mathrm{t})^2 dt \tag{22}$$

Where $e_i(\mathrm{t})$ is the value of tracking error and T is the control system running time.

5. Simulation results

The dynamics of a two links manipulator has been mentioned in section (2), so the element of this equation such as $M\left(q\right)$, $C\left(q,\dot{q}\right)$ and $G\left(q\right)$ are given as follows [4]:

$$\begin{pmatrix}\tau_1\\ \tau_2\end{pmatrix} = \begin{pmatrix}M_{11} & M_{12}\\ M_{21} & M_{22}\end{pmatrix}\begin{pmatrix}\ddot{q}_1\\ \ddot{q}_2\end{pmatrix} + \begin{pmatrix}-C\dot{q}_2 & -C\left(\dot{q}_1+\dot{q}_2\right)\\ C\dot{q}_1 & 0\end{pmatrix}\begin{pmatrix}\dot{q}_1\\ \dot{q}_2\end{pmatrix}, \tag{23}$$

$G(q)=0$

Where:

$$M_{11} = a_1 + 2a_3\cos q_2 + 2a_4\sin q_2 \tag{24}$$

$$M_{12} = M_{21} = a_2 + a_3\cos q_2 + a_4\sin q_2 \tag{25}$$

$$M_{22} = a_2 \tag{26}$$

$$C = a_3\sin q_2 - a_4\cos q_2 \tag{27}$$

$$a_1 = I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2 \tag{28}$$

$$a_2 = I_e + m_e l_{ce}^2 \tag{29}$$

$$a_3 = m_e l_1 l_{ce}\cos\delta_e \tag{30}$$

$$a_4 = m_e l_1 l_{ce}\sin\delta_e \tag{31}$$

In the simulations, the below values have been used in the following order:

$m_1 = 1$, $l_1 = 1$, $m_e = 2$, $\delta_e = \frac{\pi}{6}$, $I_1 = 0.12$, $l_{c1} = 0.5$, $I_e = 0.25$, $l_{ce} = 0.6$

The components of matrix of Y $(\mathrm{q},\dot{q},\dot{q}_r,\ddot{q}_r)$ can be written explicitly:

$\mathrm{Y}_{11} = \ddot{q}_{r1}$, $\mathrm{Y}_{12} = \ddot{q}_{r2}$, $\mathrm{Y}_{21} = 0$, $\mathrm{Y}_{22} = \ddot{q}_{r1} + \ddot{q}_{r2}$

$$\mathrm{Y}_{13} = \left(2\ddot{q}_{r1} + \ddot{q}_{r2}\right)\cos q_2 - \left(\dot{q}_2\dot{q}_{r1} + \dot{q}_1\dot{q}_{r2} + \dot{q}_2\dot{q}_{r2}\right)\sin q_2$$

$$\begin{aligned} Y_{14} &= \left(2\ddot{q}_{r1}+\ddot{q}_{r2}\right)\sin q_2 + \\ &\left(\dot{q}_2\dot{q}_{r1}+\dot{q}_1\dot{q}_{r2}+\dot{q}_2\dot{q}_{r2}\right)\cos q_2 \\ Y_{23} &= \ddot{q}_{r1}\cos q_2 + \dot{q}_1\dot{q}_{r1}\sin q_2 \\ Y_{24} &= \ddot{q}_{r1}\sin q_2 - \dot{q}_1\dot{q}_{r1}\cos q_2 \end{aligned} \tag{32}$$

The desired trajectory is chosen as:

$$\begin{aligned} q_{d1}(t) &= \frac{\pi}{6}\left(1-\cos(2\pi t)\right) \\ q_{d2}(t) &= \frac{\pi}{4}\left(1-\cos(2\pi t)\right) \end{aligned} \tag{33}$$

$\Gamma = diag\begin{bmatrix}3.3 & 0.97 & 1.04 & 0.6\end{bmatrix}, \Lambda = 20I$,
$K = 100I$

The controller parameters have been set with PSO, such as :

$\Gamma = diag\begin{bmatrix}\Gamma_1 & \Gamma_2 & \Gamma_3 & \Gamma_4\end{bmatrix}, \Lambda = \Lambda_1 I \ , K = K_1 I$

The searching ranges are set as follows:

$0 \le \Gamma_1 \le 0.07$, $0 \le \Gamma_2 \le 0.05$, $0 \le \Gamma_3 \le 0.15$, $0 \le \Gamma_4 \le 0.3$, $0 \le \Lambda_1 \le 20$, $0 \le K_1 \le 100$

In all PSO algorithms, $c_1 = c_2 = 2$ [17], w decreases from 0.9 to 0.4, in NWD-PSO n=1.2 [18] and in IPSO n=1.5 [21], population size is set to 10 and maximum number of iteration is set to 50 and each algorithm runs 25 times.

Table 1. Results of comparison between LDW-PSO, NDW-PSO, IPSO, F-PSO.

Control parameters	Real value	LDW-PSO	NDW-PSO	IPSO	F-PSO
Γ_1	0.03	0.0595	0.0591	0.0420	0.0415
Γ_2	0.05	0.0500	0.0499	0.0482	0.0500
Γ_3	0.1	0.1499	0.1499	0.1499	0.1500
Γ_4	0.3	0.3000	0.2999	0.2388	0.2996
Λ_1	20	19.9996	19.9978	19.9986	19.9991
K_1	100.000	99.9987	99.9981	99.9960	99.9869

Table 2. Results of LDW-PSO, NDW-PSO, IPSO and F-PSO algorithm.

Algorithms	Best result	Mean result	Worst result	Std
LDW-PSO	0.0037074	0.0037089	0.0037173	2.5757×10^{-6}
NDW-PSO	0.0037074	0.0037090	0.0037154	2.1170×10^{-6}
IPSO	0.0037083	0.0037322	0.0038063	2.2897×10^{-5}
F-PSO	0.0037076	0.0037111	0.0037155	2.1454×10^{-6}

Table 3. Iteration and time required by LDW-PSO, NDW-PSO, IPSO and F-PSO.

Algorithms	Best result		Average result		Worst result	
	Iterations	Elapse time(s)	Iterations	Elapse time(s)	Iteration	Elapse time(s)
LDW-PSO	35	24228	41	24534	48	24591
NDW-PSO	30	22935	34	23216.7857	35	23456
IPSO	33	24571	45	24673	47	24696
F-PSO	28	27531	32	27561	35	27695

Table 1 exhibits the average of results obtained for adaptive controller parameters and table 2 shows the results ISE for LDW-PSO, NDW-PSO, IPSO and F-PSO, where each algorithm runs 25 times and table 3 shows iteration and necessary time to reach the best, mean and worst results.

Figures 1-6 confirm the success of optimization by F-PSO algorithm compared with the other algorithms for parameters of optimal controller $\Lambda_1, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, K_1$.

These figures are represented from iteration 1 to iteration 50. Figure 7 exhibits the convergence of the optimal ISE. It confirms the superiority of F-PSO algorithm in terms of convergence speed without the premature convergence problem.

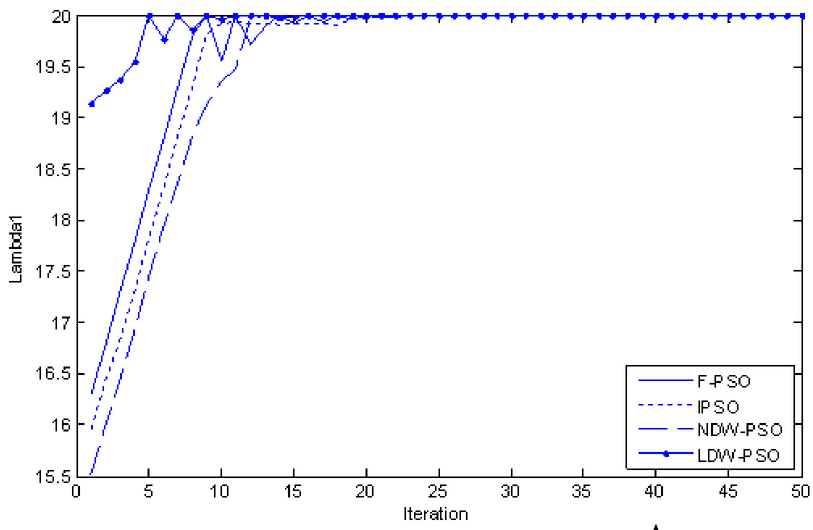

Figure 1. Comparison of trajectories of parameter Λ_1 **.**

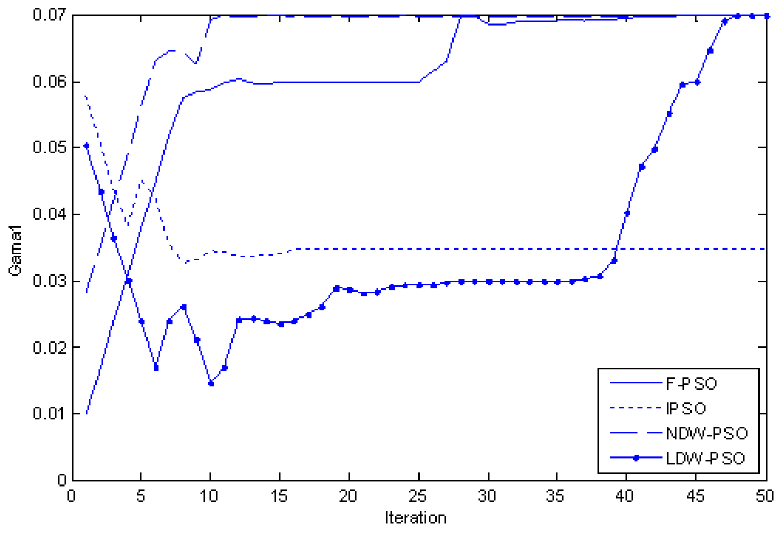

Figure 2. Comparison of trajectories of parameter Γ_1 **.**

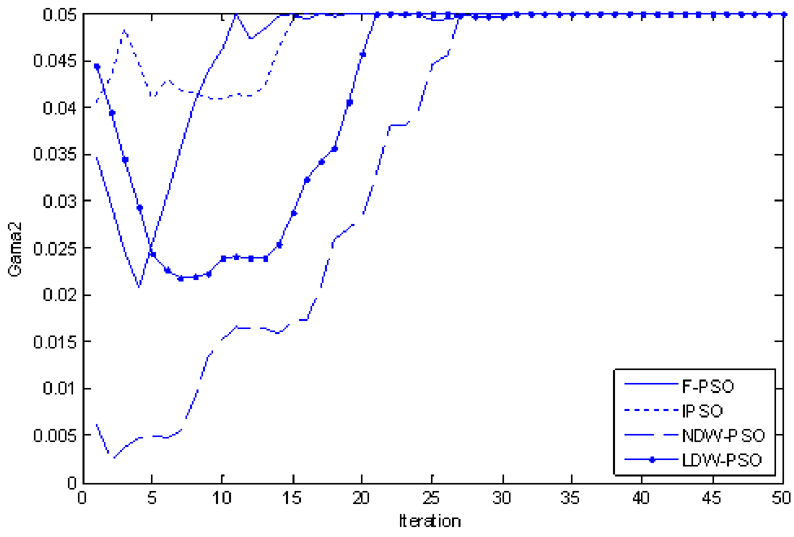

Figure 3. Comparison of trajectories of parameter Γ_2 **.**

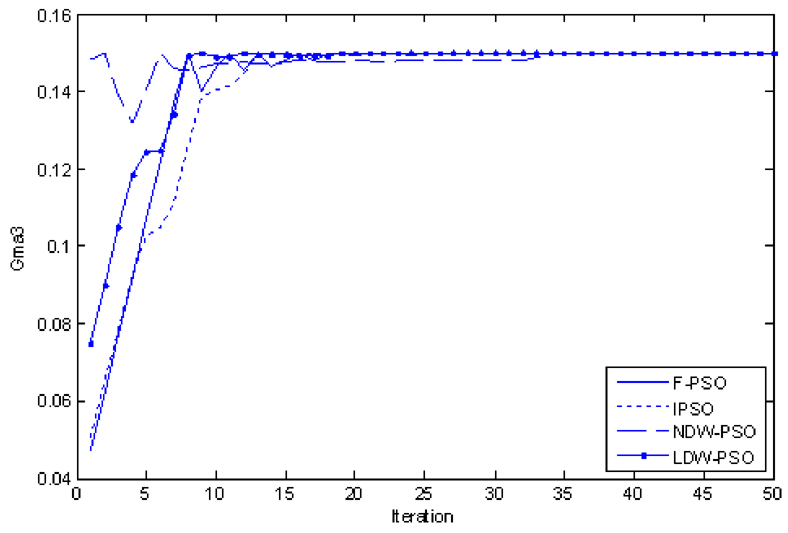

Figure 4. Comparison of trajectories of parameter Γ_3 .

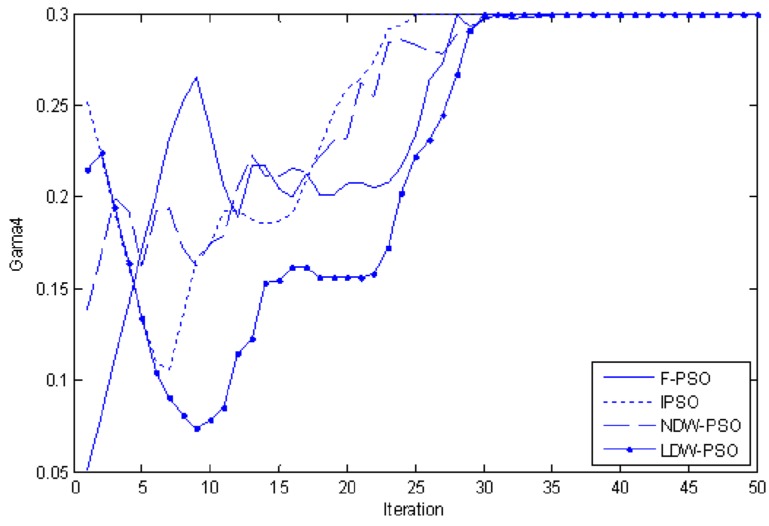

Figure 5. Comparison of trajectories of parameter Γ_4 .

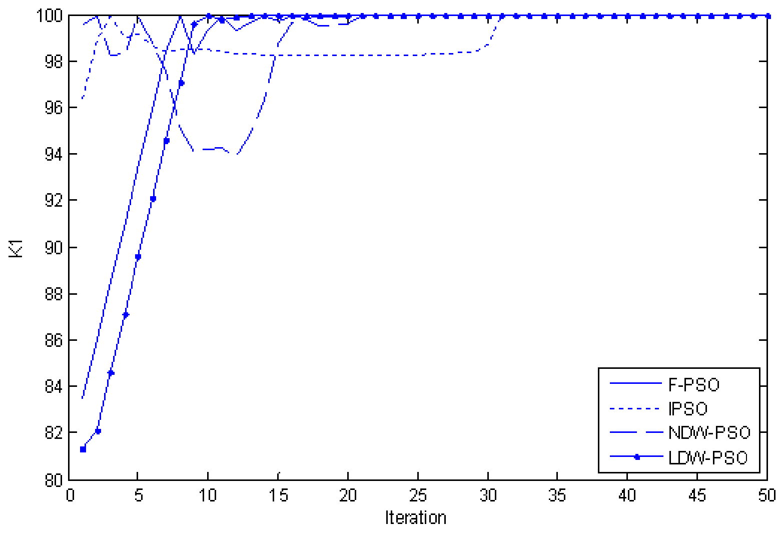

Figure 6. Comparison of trajectories of parameter K_1 .

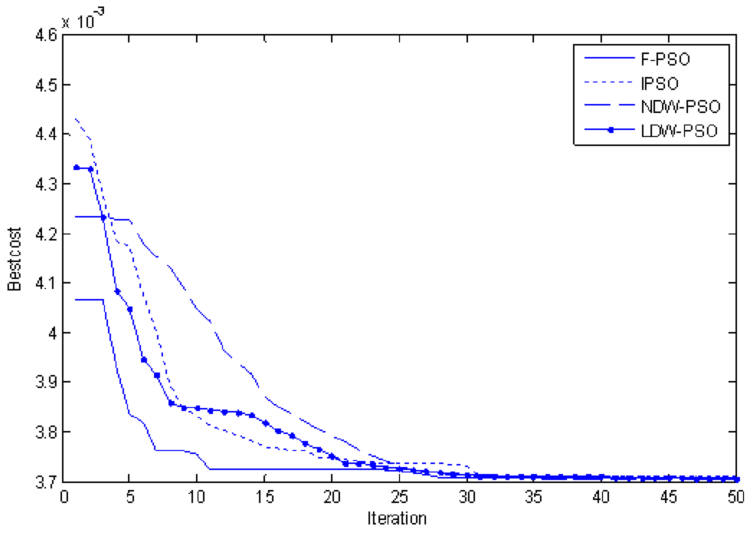

Figure 7. Comparison of convergence of objective function.

6. Conclusion

PSO has been efficient to design the adaptive controller by finding the optimal control parameters. The fuzzy system was proposed for adjusting the parameters for particle swarm optimization. It can improve the quality of result of method in the particle swarm optimization. The simulation results obtained from F-PSO, NDW-PSO, LDW-PSO and IPSO algorithms were compared . The simulation results also show the F–PSO has a better performance for purposes of parameter accuracy and convergence speed than the other algorithms.

References

[1] Alvarez-Ramirez, J., Cervantes, I. & Kelly, R. (2000). PID regulation of robot manipulators. stability and performance. System & Control Letters, vol. 41, pp. 73-83.

[2] Burkan, R. & Uzmay, I. (2005). A model of parameter adaptive law with time varying function for robot control. Applied Mathematical Modelling, vol. 29, pp. 361-371.

[3] Faieghi, M. R., Delavari, H. & Baleanu, D. (2012). A novel adaptive controller for two-degree of reedom polar robot with unknown perturbations. Commun Nonlinear SciNumer Simulate, vol. 17, pp. 1021-1030.

[4] Zeinali, M. & Notash, L. (2010). Adaptive sliding mode control with uncertainty estimator for robot manipulators. Mechanism and Machine Theory, vol. 45, pp. 80-90.

[5] Choi, Y., Chung, W. K. & Youm, Y. (2001). On the Optimal PID Performance Tuning for Robot Manipulators. IEEE/RSJ International Conference On Advanced Intelligent Robots and Systems, Maui, Hawaii, US, 2001.

[6] Wai, R.J., Tu, C. H. & Hsieh, K. Y. (2003). Design of Intelligent Optimal Tracking Control for Robot Manipulator. IEEE/ASME International Conference On Advanced Intelligent Mechatronics, 2003.

[7] Perez P, J., Perez, J. P., Soto, R., Flores, A., Rodriguez, F. & Meza, J. L. (2012). Trajectory Tracking Error Using PID Control Law for Two Link Robot Manipulator via Adaptive Neural Networks. Procedia Technology, vol. 3, pp. 139-146.

[8] Tomei, P. (1991). Adaptive PD controller for robot manipulators. IEEE Trans. Robot. Automat, vol. 7, pp. 565–570.

[9] Girirajkumae, S. M., Jayaraj, D. & Kishan, A. R. (2010). PSO based Tuning of a PID Controller for a High Performance Drilling Machine. International Journal of Computer Applications, vol. 1, pp. 0975-8887.

[10] Cao, S., Tu, J. & Liu, H. (2010). PSO Algorithm-Based Robust design of PID Controller for PMSM. Sixth International Conference on natural Computation, 2010.

[11] Chang, W. D. & Shih, S. P. (2010). PID controller design of nonlinear systems using an improved particle swarm optimization approach. Commun Nonlinear SciNumerSimulat, vol. 15, pp. 3632-3639.

[12] Sun, S., Zhang, J., W, J. & X, L. (2011). The Application of New Adaptive PSO in AGC and AFC Combination Control System. Procedia Engineering, vol. 16, pp. 702-707.

[13] Chen, S. M. & Dong, Y. F. (2011). Satellite Attitude Tracking Controller Optimization based on Particle Swarm Optimization. Procedia Engineering, vol. 15, pp. 526-530.

[14] Jaberipour, M., Khorram, E. & Karimi, B. (2011). Particle swarm algorithm for solving systems of nonlinear equations. Computers and Mathematics with Application, vol. 62, pp. 566-576.

[15] Modares, H. & Naghibi Sistani, M. B. (2011). Solving nonlinear optimal control problems using a hybrid IPSO-SQP algorithm. Engineering Application of Artificial Intelligence, vol. 24, pp. 476-484.

[16] Mehdi, h. & Boubaker, O. (2011). Position/force control optimized by Particle Swarm intelligence for constrained robotic manipulator. 11th International Conference on Intelligent System Design and Applications, 2011.

[17] Kennedy, J. & Eberhart, R. C. (1995). Particle swarm optimization. IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948 .

[18] Chatterjee, A. & Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Computers and Operations research, vol. 33, no. 3, pp. 859-871.

[19] Shi, Y. and Eberhart, R. C. (1998a). Parameter selection in particle swarm optimization. Seventh Annual Conference on Evolutionary Programming, New York, pp. 591-600.

[20] Shi, Y. & Eberhart, R. C. (1998b). A modified particle swam optimizer. Conference on Evolutionary Computation, pp. 69-73.

[21] Modares, H., Alfi, A. & Fateh M. M. (2010). Parameter identification of chaotic dynamic systems through an improved particle swarm optimization, Expert Systems with Applications, vol. 37, pp. 3714-3720.

[22] Shi, Y. (2001). Fuzzy Adaptive Particle Swarm Optimization. Proceeding of the Congress on Evolutionary computation, vol. 1, pp. 101-106.

[23] Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J. & Valdez, M. (2013). Optimal design of fuzzy classification system using PSO with dynamic parameter adaptation through fuzzy logic. Expert system with applications, vol. 40, pp. 3196-3206.

نشریه هوش مصنوعی و داده کاوی

طراحی کنترل‌کننده فازی تطبیقی برای بازوی ربات با استفاده از الگوریتم بهینه‌سازی اجتماع پرندگان

فاطمه سلیمان نوری، محمد حداد ظریف و محمد مهدی فاتح

دانشکده مهندسی برق و رباتیک، دانشگاه شاهرود، شاهرود، ایران.

ارسال ۲۰۱۳/۱۲/۱۰؛ پذیرش ۲۰۱۴/۰۶/۲۸

چکیده:

در این مقاله، الگوریتم بهینه‌سازی اجتماع پرندگان برای طراحی کنترل‌کننده تطبیقی بهینه در حوزه کنترل ردیابی بازوی ربات ارائه شده است. الگوریتم بهینه‌سازی اجتماع پرندگان برای بهینه‌سازی پارامترهای کنترل‌کننده استفاده شده است و از این‌رو انتگرال مجذور خطا به عنوان شاخص عملکرد، مینیمم می‌شود. در این مقاله، الگوریتم بهینه‌سازی اجتماع پرندگان بهبود یافته با استفاده از منطق پیشنهاد شده، سرعت همگرایی را افزایش می‌دهد. در این مقاله، عملکرد الگوریتم‌های بهینه‌سازی پرندگان از قبیل الگوریتم بهبود یافته، الگوریتم بهبودیافته با استفاده از فازی، الگوریتم ضریب اینرسی کاهشی خطی، الگوریتم ضریب اینرسی کاهشی غیرخطی از نظر دقت پارامتر و سرعت همگرایی مقایسه شده است. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم بهبودیافته با استفاده از فازی در کنترل مسیر بازوی ربات عملکرد بهتری نسبت به بقیه الگوریتم‌ها ارائه داده است.

کلمات کلیدی: الگوریتم بهینه سازی پرندگان، بازوی ماهر ربات، کنترل تطیبقی، الگوریتم بهبودیافته با استفاده از فازی، انتگرال مربعات خطا.