

# Model Free Adaptive Control for Robotic Manipulator Trajectory Tracking

Yin Yanling\*

School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, 454000, China

**Abstract:** This paper considers the problem of trajectory tracking of robotic manipulator system by using model free adaptive control (MFAC). The dynamic linearization technique is first introduced, and then the controller can be designed only by I/O data of the robotic manipulator system via MFAC approach, not includes any explicit model information. With some given conditions for controller parameters, the stability of MFAC for robotic manipulator system can be given. It is shown that the tracking error of robotic manipulator can converge to zero and the better tracking performance can be obtained. Simulation result for two-link robot manipulator further given to valid the effective of the proposed method.

**Keywords:** Model free adaptive control; robot manipulator; trajectory tracking.

## I. INTRODUCTION

Robotic manipulators have been widely applied as a kind of highly automatic equipment at present. They have the capability of improving the product quality and increasing the production efficiency [1, 2]. However, the model of robotic manipulator is of strong times variation and high non-linearity, which results in the PID controllers are weak robustness and poor performance for robotic manipulators. To solving the problem, some controllers with strong robustness have been developed, such as beck-stepping control [3], neural networked control [4, 5], fuzzy control [6], adaptive control [7, 8], sliding mode control [9, 10] and robust control [11, 12], etc. These controllers can obtain accurate trajectory tracking with fast error convergence even though the existing of kinds of various disturbance.

The aforementioned approaches provide a design methodology based on the mathematical model of robotic manipulators. However, the accurate model of an actual robotic manipulators system is often difficult to be obtained due to the complexity of practical operation environment, sometimes, it is impossible. Though neural networked control and fuzzy control not need the exact model of robotic manipulators, the establishing of fuzzy rules or training neural network should understand some necessary system information, such as the training system input and output data for neural network. Moreover, the performance of these controllers is determined by fuzzy rules or the neural network model [13].

Model free adaptive control (MFAC) is a data-driven approach, which design controller just depending on the input and output data of the plant and the adaptive control can be realized even for systems with time-varying parametric and time-varying structural [14]. A novel concept is given in

MFAC called pseudo-partial derivative (PPD), which is the key technique for controller design. By using PPD, some time-varying equivalent dynamical linearized models can be given for a class of nonlinear systems at the operation points [14]. The time varying PPD can also be estimated merely depending on the input and output data of the plant. For the different application of the data length, the dynamic linearization method can be categorized as the compact form of dynamic linearization (CFDL), partial form dynamic linearization (PFDL), and full form dynamic linearization (FFDL) [14]. Based on these dynamic linearization approaches, the relevant MFAC schemes can be given. This technique has been extensively studied with significant progress in both theoretical aspects and applications [15-19]. In this paper, we proposed an application of MFAC to the robotic manipulators system. Comparing with existing robotic manipulator control methods, there are some distinguished features for MFAC as follows. First of all, only the measurement I/O data of the robotic manipulators system, rather than any information about system model, are required for controller design. Second, MFAC is a low cost control, since no experimental signal, test signal or training process is needed. Third, it is easy to implement MFAC with low computational burden. Finally, the controller design does not depend on model information, thus it has strong robustness.

The rest of this paper is organized as follows. In Section 2, the nonlinear robot manipulator system model is given based on Lagrangian formulation, and the MFAC controller is designed in Section 3. A numerical example is given to validate the effectiveness of the algorithm in section 4. Conclusions are given in Section 5.

## 2. PROBLEM FORMULATION

Using the Lagrangian formulation, the equation of  $n$  degrees-of-freedom rigid manipulator can be expressed by [20-22].

\*Address correspondence to this author at the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, 454000, China; Tel: 0391-3987597; E-mail: [jzityy1@126.com](mailto:jzityy1@126.com)

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

where  $t$  denotes the time. The signals  $q(t) \in R^n, \dot{q}(t) \in R^n$  and  $\ddot{q}(t) \in R^n$  are the joint position, joint velocity and joint acceleration vectors, respectively.  $M(q(t)) \in R^{n \times n}$  is the inertia matrix,  $C(q(t), \dot{q}(t)) \in R^{n \times n}$  is a vector resulting from Coriolis and centrifugal forces.  $G(q(t)) \in R^n$  is the vector resulting from the gravitational forces.  $\tau(t) \in R^n$  is the control input vector containing the torques and forces to be applied at each joint [20].

In this paper, we assume that position and velocity signal are both available. The purpose of here is to design a control law  $\tau(t)$ , such that  $q(t)$  tends to desired reference trajectory  $q_d(t)$  as  $t$  tends to infinity.

Due to the positive of  $M(q(t))$ , left-multiplying  $M^{-1}(q(t))$  in both of (1) we can obtain

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

that is

$$\begin{bmatrix} \dot{q}(t) \\ \ddot{q}(t) \end{bmatrix} = \begin{bmatrix} \dot{q}(t) \\ -M^{-1}(q(t))C(q(t), \dot{q}(t))\dot{q}(t) - M^{-1}(q(t))G(q(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}(q(t))\tau(t) \end{bmatrix}. \quad (2)$$

Define  $y(t) = [\dot{q}(t), \ddot{q}(t)]^T, u(t) = \tau(t)$ , then Eq.(2) can be expressed as

$$\dot{y}(t) = f(y(t), u(t)), \quad (3)$$

If the sample time is  $h$ , the Eq.(3) can also be described in a discrete-time domain as

$$y(t+1) = f(y(t), u(t)). \quad (4)$$

To this end, the problem of robot manipulator trajectory tracking can be given as: For a desired trajectory  $y_d(t)$ , design a control input  $u(t)$  such that the system output satisfied  $y(t) = y_d(t)$  when  $t \rightarrow \infty$ .

### 3. CONTROLLER DESIGN

From the model (2), we can give the following assumptions for system (4):

**A1[14]:** the partial derivative of  $f(\dots)$  with respect to control input  $u(t)$  is continuous.

**A2[14]:** the system (4) is generalized Lipschitz, that is,  $\|\Delta y(t+1)\| \leq b \|\Delta u(t)\|$  for any  $t$  and  $\|\Delta u(t)\| \neq 0$  with

$\Delta y(t+1) = y(t+1) - y(t), \Delta u(t) = u(t) - u(t-1)$  and  $b$  is a positive constant.

In the following theorem, it is shown that, if the robot manipulator system (4) satisfying assumptions A1-A2, then it can be transformed into an equivalent dynamical form linearization model.

**Theorem 1[14]:** For the robot manipulator(4) with assumptions A1 and A2, there exists a  $\phi(t)$ , called pseudo-partial-derivative (PPD) matrix, such that if  $\|\Delta u(t)\| \neq 0$ , the system (4) can be described as the following model

$$\Delta y(t+1) = \phi(t)\Delta u(t), \quad (5)$$

$$\text{where } \phi(t) = \begin{bmatrix} \phi_{11}(t) & \phi_{12}(t) \\ \phi_{21}(t) & \phi_{22}(t) \end{bmatrix} \text{ and } \|\phi(t)\| \leq b.$$

**Proof :** From (4), we have

$$\begin{aligned} \Delta y(t+1) &= y(t+1) - y(t) \\ &= f(y(t), u(t)) - f(y(t-1), u(t-1)) \\ &= f(y(t), u(t)) - f(y(t), u(t-1)) \\ &\quad + f(y(t), u(t-1)) - f(y(t-1), u(t-1)). \end{aligned} \quad (6)$$

Using differential mean value theorem and Assumption 3, Eq.(6) becomes

$$\Delta y(t+1) = \frac{\partial f^*}{\partial u} \Delta u(t) + \Psi(t), \quad (7)$$

where

$$\frac{\partial f^*}{\partial u} = \begin{bmatrix} \frac{\partial f_1^*}{\partial u_1} & \frac{\partial f_2^*}{\partial u_1} \\ \frac{\partial f_1^*}{\partial u_2} & \frac{\partial f_2^*}{\partial u_2} \end{bmatrix},$$

$$\Psi(t) = f(y(t), u(t-1)) - f(y(t-1), u(t-1)).$$

$\frac{\partial f_i^*}{\partial u_j}, i=1,2, j=1,2$  represents the partial derivative value of  $f_i$  at some point in the interval  $[u_j(t-1), u_j(t)]$ .

Considering the following equation

$$\Psi(t) = \eta(t)\Delta u(t) \quad (8)$$

where  $\eta(k)$  is a matrix. Since the condition  $\|\Delta u(t)\| \neq 0$ , (8) must have a solution  $\eta(k)$ . Let

$$\phi(t) = \frac{\partial f^*}{\partial u(t)} + \eta(t),$$

then (7) can be written as  $\Delta y(t+1) = \phi(t)\Delta u(t)$ .

**Remark 1:** In the condition  $\|\Delta u(t)\| \neq 0$  and not too large altitude of  $\Delta u(t)$  is necessary. Hence, some adjustable pa-

parameter should be added in the control input criterion to guarantee the condition.

Rewritten (5) as

$$y(t+1) = y(t) + \phi(t)\Delta u(t). \quad (9)$$

Considering the following cost function

$$J(u(t)) = \left\| y^*(t+1) - y(t+1) \right\|^2 + \lambda \left\| u(t) - u(t-1) \right\|^2, \quad (10)$$

where  $\lambda$  is a positive weighted constant,  $y^*(t+1)$  is the desired output signal.

From (9) and (10), solving the equation  $\frac{\partial J(u(t))}{\partial u(t)} = 0$ , we can obtain

$$u(t) = u(t-1) + \frac{\rho \phi^T(t)}{\lambda + \|\phi(t)\|^2} \left( y^*(t+1) - y(t) \right), \quad (11)$$

where  $\rho$  is the step factor.

The objective function for parameter estimation is selected as

$$J(\phi(t)) = \left\| y(t) - y(t-1) - \phi(t)\Delta u(t-1) \right\|^2 + \mu \left\| \phi(t) - \hat{\phi}(t-1) \right\|^2.$$

Using the similar procedure of control law equations, the parameter estimation algorithm can be obtained as follows:

$$\hat{\phi}(t) = \hat{\phi}(t-1) + \frac{\eta \left( \Delta y(t) - \hat{\phi}(t-1)\Delta u(t-1) \right) \Delta u^T(t-1)}{\mu + \|\Delta u(t-1)\|^2}. \quad (12)$$

Combining (12) and (11), the MFAC scheme can be constructed as follows:

$$\hat{\phi}(t) = \hat{\phi}(t-1) + \frac{\eta \left( \Delta y(t) - \hat{\phi}(t-1)\Delta u(t-1) \right) \Delta u^T(t-1)}{\mu + \|\Delta u(t-1)\|^2}. \quad (13)$$

$$\hat{\phi}_{ij}(t) = \hat{\phi}_{ij}(1), \quad \text{if } \text{sign}(\hat{\phi}_{ij}(t)) \neq \text{sign}(\hat{\phi}_{ij}(1)), i=1,2, j=1,2$$

$$\hat{\phi}_{ij}(t) = \hat{\phi}_{ij}(1), \quad \text{if } \left| \hat{\phi}_{ij}(t) \right| \leq \varepsilon \quad (14)$$

$$u(t) = u(t-1) + \frac{\rho \phi^T(t)}{\lambda + \|\phi(t)\|^2} \left( y^*(t+1) - y(t) \right), \quad (15)$$

where  $\mu, \lambda$  are weight factors,  $\eta, \rho$  are the step-size and they are usually set as  $\eta, \rho \in (0,1)$ .  $\hat{\phi}_{ij}(1)$  is the initial value of  $\hat{\phi}_{ij}(t)$ ,  $\varepsilon$  is a small positive constant.

**Remark 2:** As stated in Remark 2, to ensure the condition  $\|\Delta u(t)\| \neq 0$ , a reset algorithm as shown in (14) as been added into this MFAC scheme. Meanwhile, the reset algorithm can also improve the tracking ability of parameter estimation algorithm for time-varying parameter.

**Remark 3:** Note that the control law (15) has no relationship with any model information of robot manipulator system. The control scheme is designed only using input and output data of the plant.

**Theorem 2:** For the robot manipulator nonlinear system (4) with Assumptions A1, A2, and using the MFAC algorithm (13)-(15), when  $y^*(t) = y^* = \text{const}$ , if  $\mu, \eta, \lambda$  are chosen property, then the system output satisfies  $\lim_{t \rightarrow \infty} (y^*(t) - y(t)) = 0$ .

**Proof:** The similar proof can be founded in [14].

**Remark 4:** In addition, compared with robust control and neural network control for robotic manipulator systems, MFAC also has its own advantages. Robust control is designed for robotic manipulator systems with uncertainties, and it is largely dependent upon the structure and accuracy of the model of the robotic manipulator systems. The MFAC controller is constructed only based on an equivalent time-varying linearization data model, in which neither the system model is utilized explicitly nor the unmodeled dynamics is involved. The control performance of MFAC would not be influenced by robotic manipulator unmodeled dynamics. Neural network control has been also applied for robotic manipulator systems successfully. For the control method that approximate the controller directly by the neural network, some prior knowledge of the robotic manipulator system, such as system order, is also needed, while an adequate training process and a heavy computational burden are the other two issues. Comparatively, the MFAC scheme neither requires any prior knowledge of the robotic manipulator system nor needs the training process, and has low computational burden.

#### 4. SIMULATION RESULTS

Let us consider a two degrees-of-freedom robot manipulator with revolute joints is shown in Fig. (1), where  $m_1, m_2$  are mass of link 1 and link 2,  $l_1, l_2$  are length of link 1 and link 2, respectively.  $q_1, q_2$  are joint angle. According to the Lagrange method, we can obtain the model described by (1) with

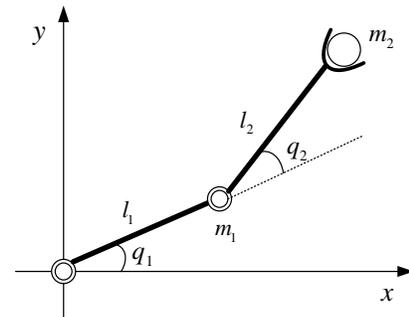


Fig. (1). The diagram of two freedom degree robot manipulator.

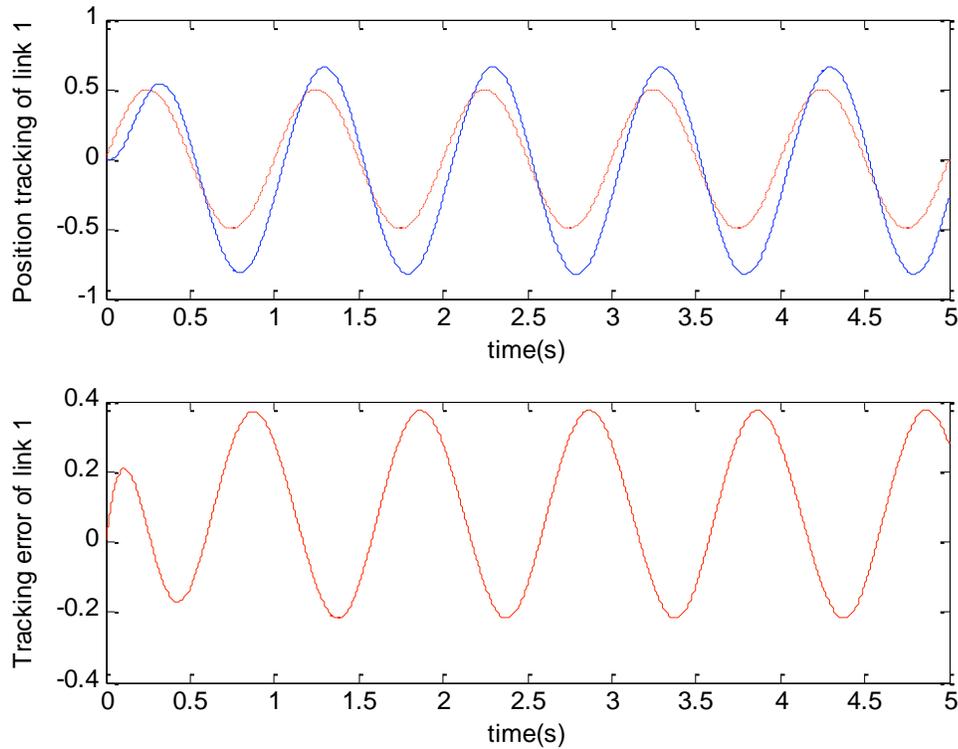


Fig. (2). The tracking performance of link 1 for PID control.

$$M = \begin{bmatrix} m_1 l_1 + m_2 (l_1^2 + l_2^2 + 2l_1 l_2 \cos q_2) + I_1 + I_2 & & & \\ & m_2 (l_2^2 + l_1 l_2 \cos q_2) + I_2 & & \\ & & m_2 (l_2^2 + l_1 l_2 \cos q_2) + I_2 & \\ & & & m_2 l_2^2 + I_2 \end{bmatrix},$$

$$C = \begin{bmatrix} -m_2 l_1 l_2 \dot{q}_2 \sin q_2 & -m_2 l_1 l_2 (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ m_2 l_1 l_2 \dot{q}_1 \sin q_2 & 0 \end{bmatrix},$$

$$G = \begin{bmatrix} (m_1 l_1 + m_2 l_1) g \cos q_1 + m_2 l_2 g \cos(q_1 + q_2) \\ m_2 l_2 g \cos(q_1 + q_2) \end{bmatrix}.$$

In this simulation, the parameters for robot manipulator are given as:  $m_1 = m_2 = 1Kg$ ,  $l_1 = l_2 = 0.5m$ ,  $I_1 = I_2 = 0.1Kg \cdot m^2$ ,  $g = 9.8m/s^2$ . the desired trajectory of  $q_1, q_2$  are given as  $q_{1d}(t) = 0.5\sin 2\pi t$ ,  $q_{2d}(t) = 0.5\cos 2\pi t$ . It is worth pointing out that the mathematical model and parameters here just serve as I/O data generator for the systems to be controlled, no any information of them will be included in the controller design. The simulation time is  $t \in [0,5]$  and the sample time is  $0.01s$ . The initial value are  $q_1(0) = 0$ ,  $q_2(0) = 0$ , and the initial control input is  $\tau_1(0) = \tau_2(0) = 0$ .

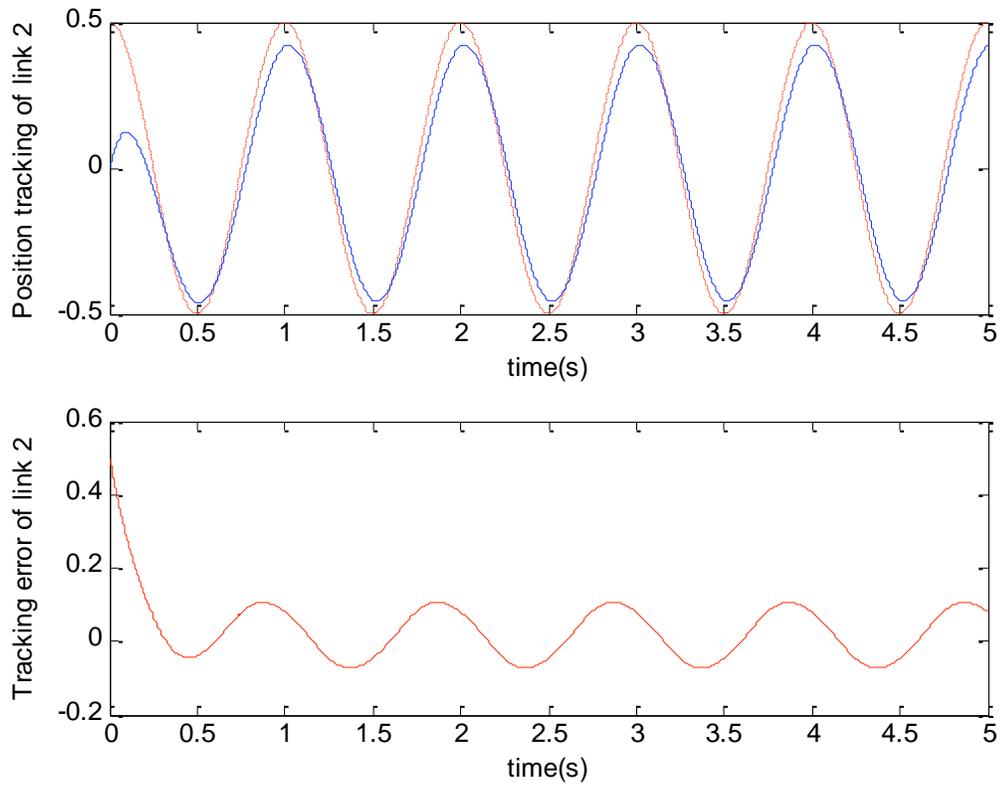
To check the performance of the proposed method, we first give simulation tests using PID control, and PID parameters are given as  $K_p = \begin{bmatrix} 50 & \\ & 50 \end{bmatrix}, K_i = 0,$

$K_d = \begin{bmatrix} 20 & \\ & 20 \end{bmatrix}$ . The simulation results are given in Fig.

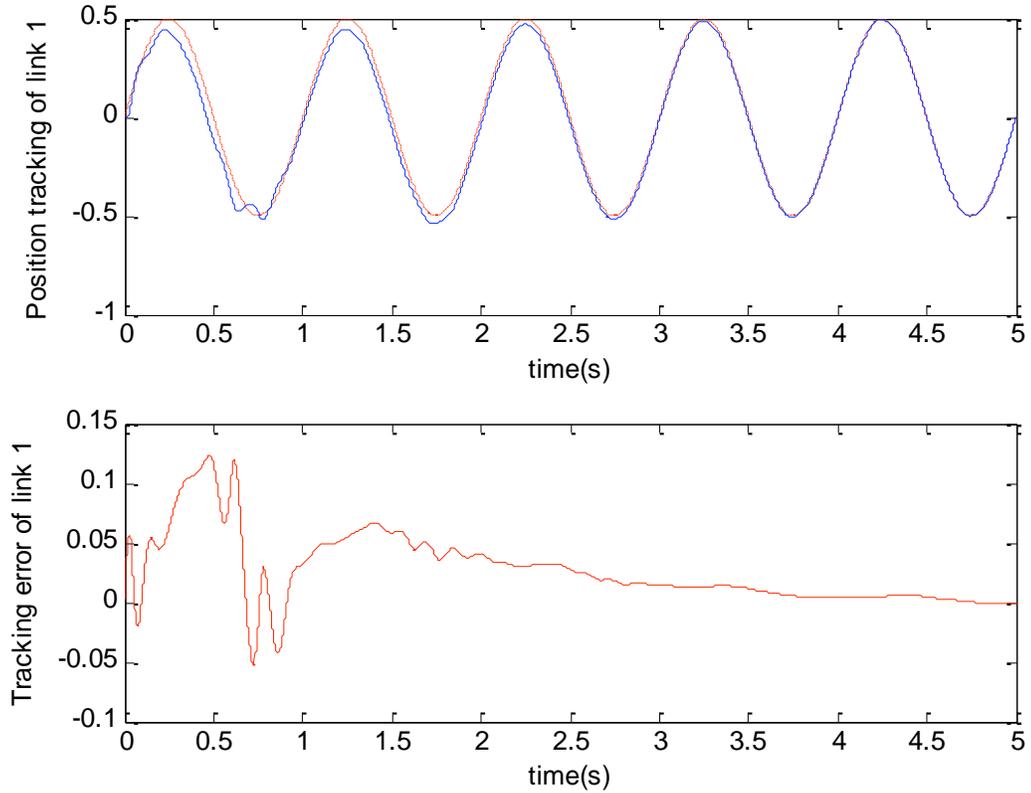
(2 and 3). The uppers of the two figures are tracking performance with dot line is desired trajectory and solid line is system output. The lowers of the two figures are tracking errors. It is obvious that PID controller cannot obtain satisfied tracking performance. Then, the MFAC algorithm is used and the controller parameters are given as

$\mu = 0.5, \eta = 0.5, \rho = 0.5, \lambda = 5, \hat{\phi}(0) = \begin{bmatrix} -0.5 & 0 \\ 0 & -0.5 \end{bmatrix}$ . The

simulation results are given in Fig. (4-7). Where the tracking performance is plotted in Fig. (4 and 5), control inputs are plotted in Fig. (6) and estimation of PPD is plotted in Fig. (7). It is obvious that the better performance can be obtained by MFAC approach. From the simulation results, we can see that the MFAC scheme has the following merits. The PPD behavior of the robot manipulator system is relatively simple, and is a slowly time-varying scalar parameter. It has a relationship with the robot manipulator system dynamics, control input, and so on, it is not sensitive to parameter of the robot manipulator system. The MFAC scheme has a very



**Fig. (3).** The tracking performance of link 2 for PID control.



**Fig. (4).** The tracking performance of link 1 for MFAC.

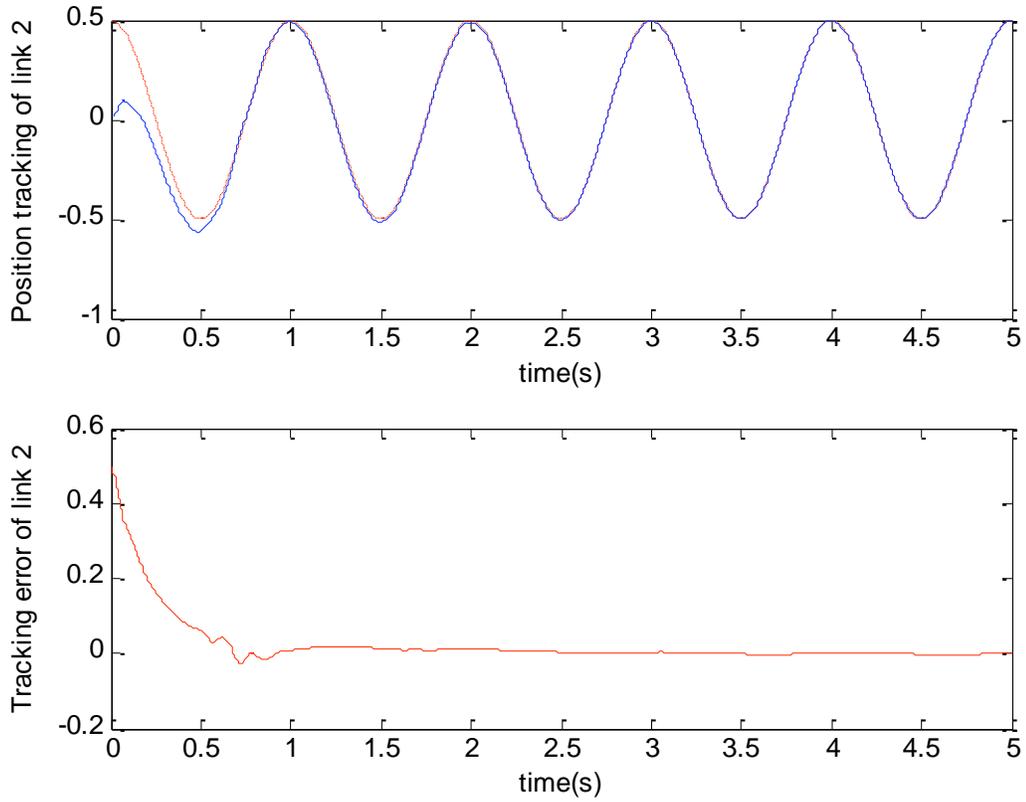


Fig. (5). The tracking performance of link 2 for MFAC.

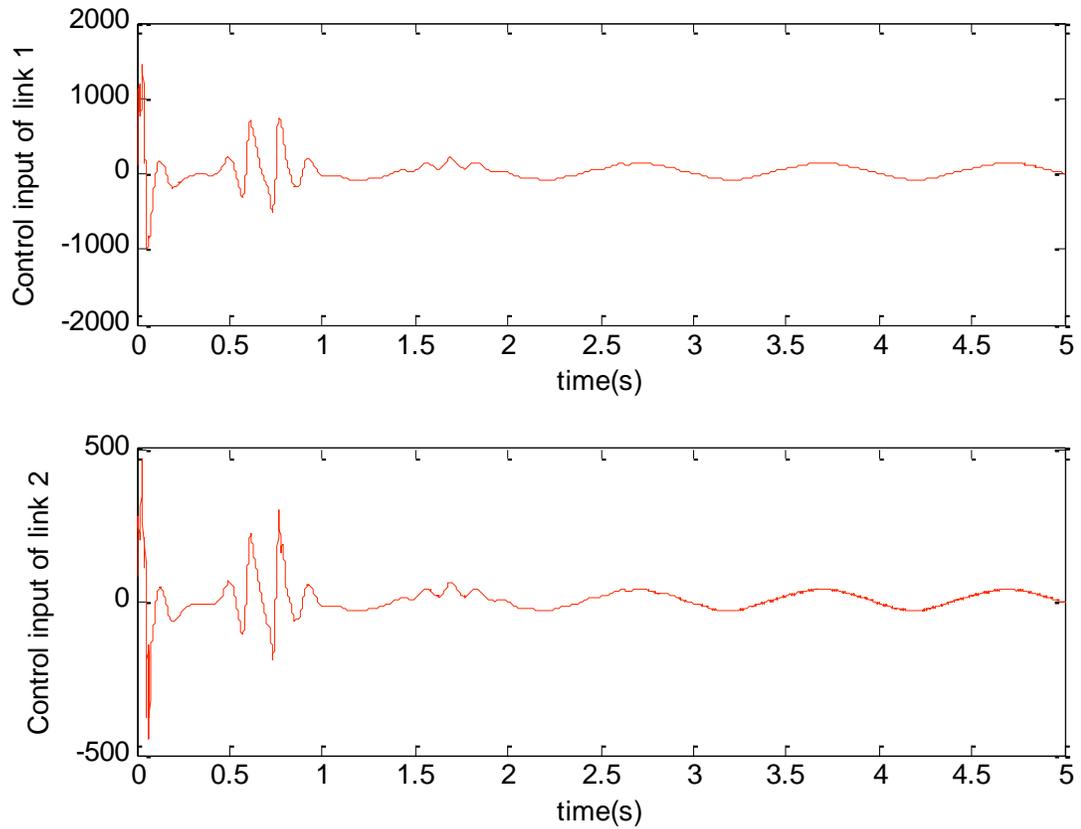


Fig. (6). The control inputs of MFAC scheme.

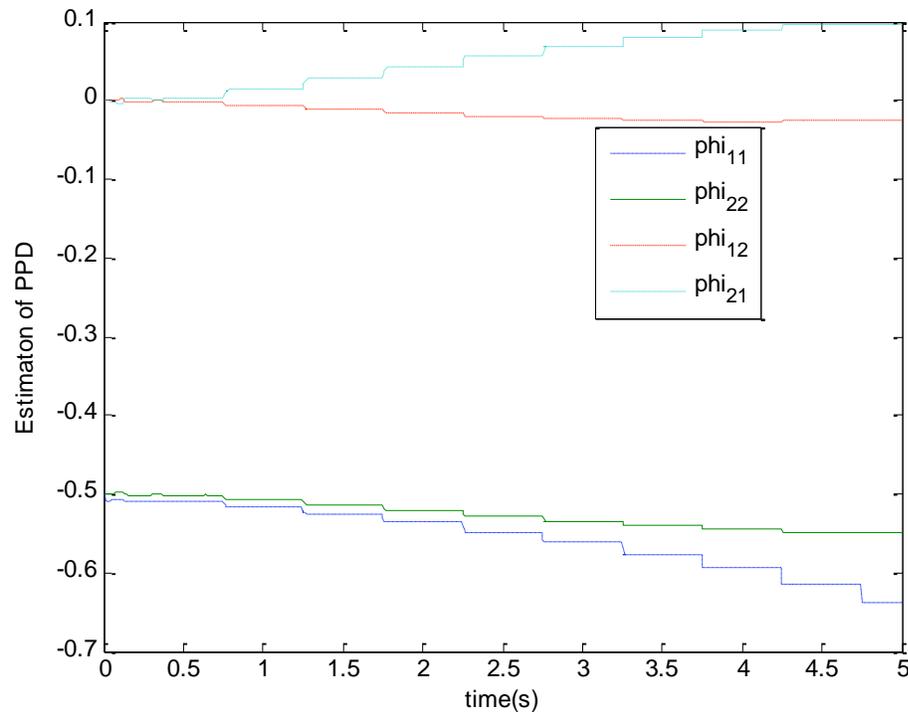


Fig. (7). Estimated PPD of MFAC schemes.

simple structure and is easy to implement with low computational burden, since there is only one scalar parameter to tune online. Hence, from the simulation result, it is obvious that this method can be easily implemented using real robots manipulator.

## 5. CONCLUSION

In this paper, we design a model free control scheme for robot manipulator trajectory tracking based on MFAC. The method first transfers the nonlinear robot manipulator system into a linear time-varying system, and then the controller is designed only by I/O data, not includes any explicit model information of the robotic manipulator system. It is shown that MFAC can guarantee system output tends to the desired trajectory under some given conditions of controller parameters, and the better tracking performance can be obtained. The example for two-link robot manipulator is further valid the effective of the proposed method.

## CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

## ACKNOWLEDGEMENTS

This work is supported by the program of Natural Science of Henan Provincial Education Department (12A510-013).

## REFERENCES

- [1] D. Luca, and B. Siciliano, "An asymptotically stable joint PD controller for robot arms with flexible links under gravity", *IEEE Conference on Decision and Control*, Tucson, AZ, 1992, pp. 325-326.
- [2] I. Cervantes, J. Alvarez-Ramirez, "On the PID tracking control of robot manipulators", *Systems & Control Letters*, vol. 42, no. 1, pp. 37-46, 2001.
- [3] J. H. Oh, and J. S. Lee, "Control of flexible joint robot system by backstepping design approach", *IEEE International Conference on Robotics and Automation*, New Mexico, 1997, pp. 3435-3440.
- [4] S. S. Ge, C. C. Huang, and L. C. Woon, "Adaptive neural network control of robot manipulators in task space", *Transactions on Industrial Electronics Magazine*, vol. 44, no. 6, pp. 746-752, 1997.
- [5] L. Chen, Z. G. Hou and M. Tan, "Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model", *Automatica*, vol. 45, no. 10, pp. 2312-2318, 2009.
- [6] N. Goléa, A. Goléa, K. Barra and T. Bouktir, "Observer-based adaptive control of robot manipulators: Fuzzy systems approach", *Applied Soft Computing*, vol. 8, no. 1, pp. 778-787, 2008.
- [7] D. Seo and M. R. Akella, "Non-certainty equivalent adaptive control for robot manipulator systems", *Systems*, vol. 58, no. 4, pp. 304-308, 2009.
- [8] J.J. Craig, P. Hsu and S. S. Sastry, "Adaptive control of mechanical manipulators", *International Journal of Robotics Research*, vol. 6, no. 2, pp. 16-28, 1987.
- [9] Z. Meysar, and N. Leila, "Adaptive sliding mode control with uncertainty estimator for robot manipulators", *Mechanism and Machine Theory*, vol. 45, no. 1, pp. 80-90, 2010.
- [10] J. Lee, P. H. Chang, and C. Choi, "Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control", *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3593-3601, 2009.
- [11] L. and P. Rocco, "Revising the robust control design for rigid robot manipulators," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 180-187, 2010.

- [12] J. P. Kolhe, M. Shaheed, T. S. Chandar and S.E. Talole, "Robust control of robot manipulators based on uncertainty and disturbance estimation", *International Journal of Robust and Nonlinear Control*, vol. 28, no. 9, pp.759-768, 2011.
- [13] Z. S. Hou and S. T. Jin, "Data-driven model-free adaptive control for a class of mimo nonlinear discrete-time systems", *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2173-2188, 2011.
- [14] Z. S. Hou and W. H. Huang, "The model-free learning adaptive control of a class of SISO nonlinear systems", *Proceedings of the American Control Conference*, New Mexico, USA, IEEE, pp. 343-344, 1997.
- [15] Z. S. Hou and S. T. Jin, "A novel data-driven control approach for a class of discrete-time nonlinear systems", *IEEE Transactions on Control Systems Technology*, vol. 9, no. 6, pp. 1549-1558, 2011.
- [16] Z. S. Hou, and X. H. Bu, "Model free adaptive control with data dropouts", *Expert Systems with Applications*, vol. 38, no. 8, pp. 10709-10717, 2011.
- [17] X. H. Bu, Z. S. Hou and S. T. Jin, "The robustness of model-free adaptive control with disturbance suppression", *Journal of Control Theory & Applications*, vol. 28, no. 3, pp. 358-362, 2011.
- [18] X. H. Bu, Z. S. Hou, F. S. Yu and Wang F. Z., "Robust model free adaptive control with measurement disturbance", *IET Control Theory & Application*, vol. 6, no. 9, pp. 1288-1296, 2012.
- [19] X. H. Bu, F. S. Yu., Z. S. Hou, and H. W. Zhang, "Model free adaptive control algorithm with data dropout compensation", *Mathematical Problems in Engineering*, vol. 9, no. 2, pp. 1-14, 2012.
- [20] Chien and A. Tayebi, "A one-parameter structure for adaptive iterative learning control of robot manipulator", *22nd IEEE International Symposium on Intelligent Control*, Singapore, 1-3 October, 327-332, 2007.
- [21] A. Tayebi, "Adaptive iterative learning control for robot manipulators", *Automatica*, vol. 40, no. 7, pp. 1195-1203, 2008.
- [22] H. T. Liu, T. Zhang, "Neural network-based robust finite-time control for robotic manipulators considering actuator dynamics", *Robotics and Computer-Integrated Manufacturing*, vol.29, no.2, pp. 301-308, 2013.

---

Received: October 30, 2014

Revised: January 09, 2015

Accepted: February 02, 2015

© Yin Yanling; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.