IOP Conf. Series: Materials Science and Engineering **428** (2018) 012051 doi:10.1088/1757-899X/428/1/012051

Motion Control of Manipulators Based on Model-free Adaptive Control

Ziqiao Zhang and Songyi Dian

Department of Automation, College of Electrical Engineering and Information Technology, Sichuan University, Chengdu, Sichuan, China

scudiansy@scu.edu.cn

Abstract. In this paper, the model-free adaptive control (MFAC) method which is based on compact form dynamic linearization (CFDL) of multiple input and multiple output (MIMO) nonlinear systems is applied to the control of Puma560 manipulator. The controller design requires only the input and output data of the system. The simulation results verify the effectiveness of the model-free adaptive control method. Based on the structure of the controller, the function of controller parameters is analysed. By adding external disturbances, the anti-interference of Puma560 based on model-free adaptive control can be verified. The model free adaptive controller is optimized.

1. Introduction

At present, the modern control theory has been continuously developed and perfected. Model-free control theory was proposed and has been gradually developed to solve the problems faced by model-based control. Model-free adaptive control(MFAC) is a data-driven model-free control theory, which only requires the input/output data of system with no need for system information to design controller. It also has easy implementation, small computational burden and strong robustness [1]. MFAC has been widely applied in linear motor control, liquid level control system, robot control system, etc.

The manipulator system is nonlinear, highly coupled and time-varying [2]. Even in well-structured environment, the system is inevitably influenced by structural and non-structural uncertainties. The use of MFAC method can reduce the influence of these uncertainties [3]. It will also achieve data-driven control of manipulators by using online data [4, 5]. In this paper, the MFAC method is applied to Puma560 manipulator system. Simulation results show the effectiveness of the MFAC method.

2. Model-free adaptive control algorithm design

Considering a class of multiple input and multiple output discrete-time nonlinear systems [6]

$$\boldsymbol{y}(k+1) = \boldsymbol{f}\left(\boldsymbol{y}(k), \dots, \boldsymbol{y}(k-n_y), \boldsymbol{u}(k), \dots, \boldsymbol{u}(k-n_u)\right) \tag{1}$$

where $\boldsymbol{u}(k) \in \mathbf{R}^m$ is control input, $\boldsymbol{y}(k) \in \mathbf{R}^m$ is system output at time k; n_y and n_u are unknown integers; $\boldsymbol{f}(\dots) = \left(f_1(\dots),\ \dots, f_m(\dots)\right)^T \in \prod_{n_u+n_y+2} \mathbf{R}^m \mapsto \mathbf{R}^m$ is an unknown nonlinear function.

For nonlinear system (1), assumptions are made as follows:

Assumption 1: the partial derivatives of $f_i(\dots), i = 1, \dots, m$, with respect to every component of $\boldsymbol{u}(k)$ are continuous.

Assumption 2: system (1) satisfies the general Lipschitz condition, that is

$$\|\boldsymbol{y}(k_1+1)-\boldsymbol{y}(k_2+1)\| \leq b\|\boldsymbol{u}(k_1)-\boldsymbol{u}(k_2)\| \tag{2}$$

for any $k_1 \neq k_2, k_1 \geq 0, k_2 \geq 0$ and $\boldsymbol{u}(k_1) \neq \boldsymbol{u}(k_2)$, b is a positive constant.

When nonlinear system satisfies these assumptions and for any given time k $\Delta\boldsymbol{u}(k) \neq 0$, the compact form dynamic linearization (CFDL) of this system can be written as follows:

$$\boldsymbol{y}(k+1)=\boldsymbol{y}(k)+\boldsymbol{\Phi}_c(k)\Delta\boldsymbol{u}(k) \tag{3}$$

where $\boldsymbol{\Phi}_c(k)=\begin{bmatrix}\phi_{11}(k) & \phi_{12}(k) & \cdots & \phi_{1m}(k)\\ \phi_{21}(k) & \phi_{22}(k) & \cdots & \phi_{2m}(k)\\ \vdots & \vdots & \vdots & \vdots\\ \phi_{m1}(k) & \phi_{m2}(k) & \cdots & \phi_{mm}(k)\end{bmatrix} \in \mathbf{R}^{m\times m}$ is the pseudo Jacobian matrix (PJM) of the system.

Use the following cost function of control input to make estimation of system controlling effect:

$$\boldsymbol{J}\big(\boldsymbol{u}(k)\big)=\|\boldsymbol{y}^*(k+1)-\boldsymbol{y}(k+1)\|^2+\lambda\|\boldsymbol{u}(k)-\boldsymbol{u}(k-1)\|^2 \tag{4}$$

where $\lambda > 0$ is the weighting factor and $\boldsymbol{y}^*(k+1)$ is the desired system output.

Differentiating (4) with respect to $\boldsymbol{u}(k)$, set this derivative to zero to get the following control law,

$$\boldsymbol{u}(k)=\boldsymbol{u}(k-1)+\left(\lambda\boldsymbol{I}+\boldsymbol{\Phi}_c{}^T(k)\boldsymbol{\Phi}_c(k)\right)^{-1}\boldsymbol{\Phi}_c{}^T(k)\big(\boldsymbol{y}^*(k+1)-\boldsymbol{y}(k)\big) \tag{5}$$

Control law (5) requires the calculation of matrix inversion and a simplified control law is introduced to avoid this process,

$$\boldsymbol{u}(k)=\boldsymbol{u}(k-1)+\frac{\rho\boldsymbol{\Phi}_c{}^T(k)\big(\boldsymbol{y}^*(k+1)-\boldsymbol{y}(k)\big)}{\lambda+\|\boldsymbol{\Phi}_c(k)\|^2} \tag{6}$$

where $\rho \in (0,1]$ is the step factor.

Cost function of PJM estimation is given as follows,

$$\boldsymbol{J}\big(\boldsymbol{\Phi_c}(k)\big)=\|\Delta\boldsymbol{y}(k)-\boldsymbol{\Phi}_c(k)\Delta\boldsymbol{u}(k-1)\|^2+\mu\left\|\boldsymbol{\Phi}_c(k)-\widehat{\boldsymbol{\Phi}}_c(k)\right\|^2 \tag{7}$$

where $\mu > 0$ is the weighting factor.

Minimizing cost function (7) and improved projection algorithm can be acquired as follows,

$$\begin{aligned}\widehat{\boldsymbol{\Phi}}_c(k)=\widehat{\boldsymbol{\Phi}}_c(k-1)+\Big(\Delta\boldsymbol{y}(k-1)-\widehat{\boldsymbol{\Phi}}_c(k-1)\Delta\boldsymbol{u}(k-1)\Big)\times\Delta\boldsymbol{u}^T(k-1)\\ \times\left(\mu\boldsymbol{I}+\Delta\boldsymbol{u}(k-1)\Delta\boldsymbol{u}^T(k-1)\right)^{-1}\end{aligned} \tag{8}$$

An improved algorithm is introduced to avoid the calculation of matrix inversion,

$$\widehat{\boldsymbol{\Phi}}_{\boldsymbol{c}}(\boldsymbol{k})=\widehat{\boldsymbol{\Phi}}_{\boldsymbol{c}}(\boldsymbol{k}-\boldsymbol{1})+\frac{\boldsymbol{\eta}\Big(\Delta\boldsymbol{y}(\boldsymbol{k})-\widehat{\boldsymbol{\Phi}}_{\boldsymbol{c}}(\boldsymbol{k}-\boldsymbol{1})\Delta\boldsymbol{u}(\boldsymbol{k}-\boldsymbol{1})\Big)\Delta\boldsymbol{u}^{\boldsymbol{T}}(\boldsymbol{k}-\boldsymbol{1})}{\boldsymbol{\mu}+\|\Delta\boldsymbol{u}(\boldsymbol{k}-\boldsymbol{1})\|^2} \tag{9}$$

where $\eta \in (0,2]$ is a step factor and $\widehat{\boldsymbol{\Phi}}_c(k)=\begin{bmatrix}\hat{\phi}_{11}(k) & \hat{\phi}_{12}(k) & \cdots & \hat{\phi}_{1m}(k)\\ \hat{\phi}_{21}(k) & \hat{\phi}_{22}(k) & \cdots & \hat{\phi}_{2m}(k)\\ \vdots & \vdots & \vdots & \vdots\\ \hat{\phi}_{m1}(k) & \hat{\phi}_{m2}(k) & \cdots & \hat{\phi}_{mm}(k)\end{bmatrix} \in \mathbf{R}^{m\times m}$ is the estimation of PJM $\boldsymbol{\Phi}_c(k)$.

The CFDL-MFAC scheme for MIMO nonlinear system can be built by combining control law (6) with PJM estimation algorithm (9) as follows,

$$\widehat{\boldsymbol{\Phi}}_c(k)=\widehat{\boldsymbol{\Phi}}_c(k-1)+\frac{\eta\Big(\Delta\boldsymbol{y}(k)-\widehat{\boldsymbol{\Phi}}_c(k-1)\Delta\boldsymbol{u}(k-1)\Big)\Delta\boldsymbol{u}^T(k-1)}{\mu+\|\Delta\boldsymbol{u}(k-1)\|^2} \tag{10}$$

$$\widehat{\boldsymbol{\phi}}_{ii}(k)=\widehat{\boldsymbol{\phi}}_{ii}(1),\ \text{if}\left|\widehat{\boldsymbol{\phi}}_{ii}(k)\right|<b_2 \text{ or } \left|\widehat{\boldsymbol{\phi}}_{ii}(k)\right|>\alpha b_2 \text{or sign}\left(\widehat{\boldsymbol{\phi}}_{ii}(k)\right)\neq \text{sign}\left(\widehat{\boldsymbol{\phi}}_{ii}(1)\right) \tag{11}$$

$$\widehat{\boldsymbol{\phi}}_{ij}(k)=\widehat{\boldsymbol{\phi}}_{ij}(1),\ \text{if}\left|\widehat{\boldsymbol{\phi}}_{ij}(k)\right|>b_1 \text{ or sign}\left(\widehat{\boldsymbol{\phi}}_{ij}(k)\right)\neq \text{sign}\left(\widehat{\boldsymbol{\phi}}_{ij}(1)\right),\ i\neq j \tag{12}$$

$$\boldsymbol{u}(k)=\boldsymbol{u}(k-1)+\frac{\rho\widehat{\boldsymbol{\Phi}}_c{}^T(k)\big(\boldsymbol{y}^*(k+1)-\boldsymbol{y}(k)\big)}{\lambda+\left\|\widehat{\boldsymbol{\Phi}}_c(k)\right\|^2} \tag{13}$$

where $\lambda > 0,\ \mu > 0,\ \rho \in (0,1],\ \eta \in (0,2]$;$\hat{\phi}_{ij}(1)$ is the initial value of $\hat{\phi}_{ij}(k)$.

3. Modelling of Puma560 manipulator system

3.1. Basic structure of Puma560 manipulator system

The system block diagram is shown in figure 1. The desired rotation angle $\boldsymbol{q}^*$, angular velocity $\dot{\boldsymbol{q}}^*$ and angular acceleration $\ddot{\boldsymbol{q}}^*$ are taken as desired signal $\boldsymbol{y}^*$. The actual rotation angle $\boldsymbol{q}$, angular velocity $\dot{\boldsymbol{q}}$ and angular acceleration $\ddot{\boldsymbol{q}}$ are used as actual signal $\boldsymbol{y}$. Desired and actual signals are used as input signals of the controller. These signals are calculated and processed using MFAC method in the controller. Control signal $\boldsymbol{u}$, which is used to control Puma560 manipulator, can be obtained as the output signals of the controller.

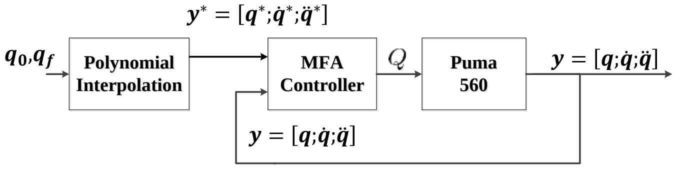


Figure 1. Block diagram of Puma560 manipulator system based on MFAC

The algorithm used by the controller is MFAC algorithm, with no requirement for information of system parameter or structure, establishes a dynamic linearized data model based on input/output data, and carries out real-time data-driven control based on the online data for the manipulator. By establishing the dynamic linearized data model, the driving torque of each joint can be obtained to control the movement of manipulator.

3.2. Controller design

An MFA controller is designed by using the MFAC scheme. Since Puma560 has 6 degrees of freedom, we can determine the dimension of $\boldsymbol{q}^*$, $\dot{\boldsymbol{q}}^*$, $\ddot{\boldsymbol{q}}^*$, $\boldsymbol{y}^*$, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, $\boldsymbol{y}$, $\boldsymbol{u}$:

$$\boldsymbol{q}^*, \dot{\boldsymbol{q}}^*, \ddot{\boldsymbol{q}}^*, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{u} \in \mathbf{R}^{6\times 1}, \boldsymbol{y}^* = [\boldsymbol{q}^*; \dot{\boldsymbol{q}}^*; \ddot{\boldsymbol{q}}^*] \in \mathbf{R}^{18\times 1}, \boldsymbol{y} = [\boldsymbol{q}; \dot{\boldsymbol{q}}; \ddot{\boldsymbol{q}}] \in \mathbf{R}^{18\times 1}.$$

The Puma560 manipulator based on MFAC can be written as (1), where $\boldsymbol{u}(k) \in \mathbf{R}^{6\times 1}$; $\boldsymbol{y}(k) \in \mathbf{R}^{18\times 1}$; $\boldsymbol{f}(\dots) = \left(f_1(\dots), \ \dots, f_{18}(\dots)\right)^T \in \prod_{n_u+n_y+2} \mathbf{R}^{6\times 1} \longmapsto \mathbf{R}^{18\times 1}$ is unknown nonlinear function.

The Puma560 system satisfies the Assumption 1 and Assumption 2. The CFDL of this system can be written as,

$$\boldsymbol{y}(k+1) = \boldsymbol{y}(k) + \boldsymbol{\Phi}_c(k)\Delta\boldsymbol{u}(k) \tag{14}$$

where $\boldsymbol{\Phi}_c(k) = \begin{bmatrix} \phi_{1,1}(k) & \phi_{1,2}(k) & \cdots & \phi_{1,6}(k) \\ \phi_{2,1}(k) & \phi_{2,2}(k) & \cdots & \phi_{2,6}(k) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{18,1}(k) & \phi_{18,2}(k) & \cdots & \phi_{18,6}(k) \end{bmatrix} \in \mathbf{R}^{18\times 6}$ is the PJM of the system.

The CFDL-MFAC scheme of this system can be written as,

$$\widehat{\boldsymbol{\Phi}}_c(k) = \widehat{\boldsymbol{\Phi}}_c(k-1) + \frac{\eta\left(\Delta\boldsymbol{y}(k) - \widehat{\boldsymbol{\Phi}}_c(k-1)\Delta\boldsymbol{u}(k-1)\right)\Delta\boldsymbol{u}^T(k-1)}{\mu + \|\Delta\boldsymbol{u}(k-1)\|^2} \tag{15}$$

$$\widehat{\boldsymbol{\phi}}_{ii}(k) = \widehat{\boldsymbol{\phi}}_{ii}(1), \text{ if} \left|\widehat{\boldsymbol{\phi}}_{ii}(k)\right| < b_2 \text{ or } \left|\widehat{\boldsymbol{\phi}}_{ii}(k)\right| > \alpha b_2 \text{or sign}\left(\widehat{\boldsymbol{\phi}}_{ii}(k)\right) \neq \text{sign}\left(\widehat{\boldsymbol{\phi}}_{ii}(1)\right) \tag{16}$$

$$\widehat{\boldsymbol{\phi}}_{ij}(k) = \widehat{\boldsymbol{\phi}}_{ij}(1), \text{ if} \left|\widehat{\boldsymbol{\phi}}_{ij}(k)\right| > b_1 \text{ or sign}\left(\widehat{\boldsymbol{\phi}}_{ij}(k)\right) \neq \text{sign}\left(\widehat{\boldsymbol{\phi}}_{ij}(1)\right), \ i \neq j \tag{17}$$

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \frac{\rho\widehat{\boldsymbol{\Phi}}_c{}^T(k)(\boldsymbol{y}^*(k+1) - \boldsymbol{y}(k))}{\lambda + \left\|\widehat{\boldsymbol{\Phi}}_c(k)\right\|^2} \tag{18}$$

where $\lambda > 0$, $\mu > 0$, $\rho \in (0,1]$, $\eta \in (0,2]$; $\hat{\phi}_{ij}(1)$ is the initial value of $\hat{\phi}_{ij}(k)$.

In addition to determining input and output signals of the controller, the MFAC controller design also needs to initialize input/output signals of the controller, and to determine the weight factors λ, μ, the step factor ρ, η used in the control algorithm.

4. Simulation and analysis

Take initial joint position $q_0 = [0\ 0\ 0\ 0\ 0\ 0]$, final joint position $q_f = \left[\frac{\pi}{4}\ \ -\frac{\pi}{2}\ \frac{\pi}{2}\ 0\ 0\ 0\ 0\right]$, simulation time $t_{max} = 10s$, time interval $\Delta t = t_{max}/10000 = 0.001s$, weight factors $\lambda = 5$, $\mu = 11$, step factors $\rho = 1$, $\eta = 0.45$. By using kinematics model, the motion of the terminal actuator can be observed and analyzed in the Cartesian space according to the rotation angle of each joint.

4.1. Parameters adjustment

4.1.1 Weighting factors. The influence of weighting factors on control effect is analyzed by changing weighting factors of CFDL-MFAC scheme.

The simulation of λ=0.1,1,5,10,100 and μ=0.1,1,11,50, 200 is carried out respectively. The motion of the terminal actuator is plotting in figure 2 and 3, and the comparative analysis is made.

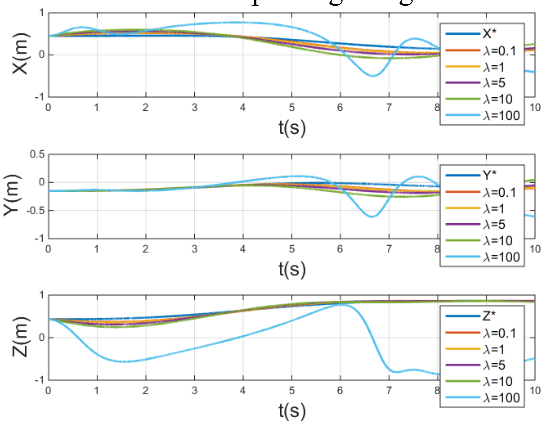


Figure 2. Changing weighting factor λ

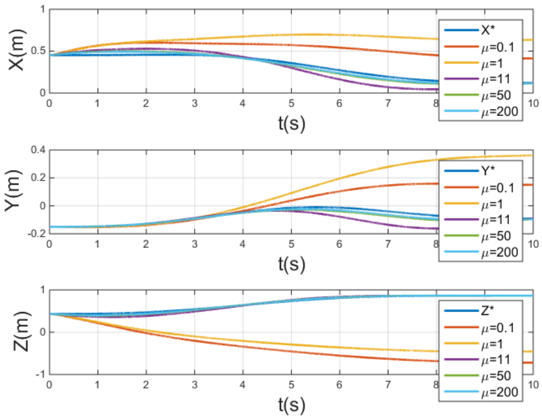


Figure 3. Changing weighting factor μ

From the simulation results, we can see that when λ is too large, the CFDL-MFAC scheme cannot get satisfactory control effect. It can be seen from the control scheme (18) that, when other conditions are the same, the smaller λ is, the smaller $\Delta\boldsymbol{u}(k)$, which means that the change of control input becomes smaller, and this will lead to the decrease of manipulator control's sensibility. When lambda is smaller, $\Delta\boldsymbol{u}(k)$ is mainly dependent on the value of $\widehat{\boldsymbol{\Phi}}_c(k)$, $\boldsymbol{y}^*(k+1)$ and $\boldsymbol{y}(k)$, which means that $\Delta\boldsymbol{u}(k)$ is related to the desired trajectory, the actual trajectory and the structure of the system. For this reason, the change of λ will not have great influence on $\Delta\boldsymbol{u}(\boldsymbol{k})$, and affect control effect.

As for the weighting factor μ, when μ is too small, desired control effect cannot be achieved. It can be seen from the control scheme (15) that, when other conditions are the same, the smaller μ is, the bigger $\Delta\widehat{\boldsymbol{\Phi}}_{\boldsymbol{c}}(k)$, then $\Delta\boldsymbol{u}$ will decrease which leads to unsatisfactory joint rotation of manipulator. When μ becomes bigger, $\Delta\widehat{\boldsymbol{\Phi}}_{\boldsymbol{c}}(k)$ becomes smaller and will have little effect on $\Delta\boldsymbol{u}$. Therefore, when μ is big enough, changing μ will have little influence on manipulator control.

Therefore, the selection of weighting factor has effect on the control effect of the system to some extent.

4.1.2 Step factors. Chang the step size factors of CFDL-MFAC scheme ρ, η, respectively. The simulation results are shown in figure 4 and 5.

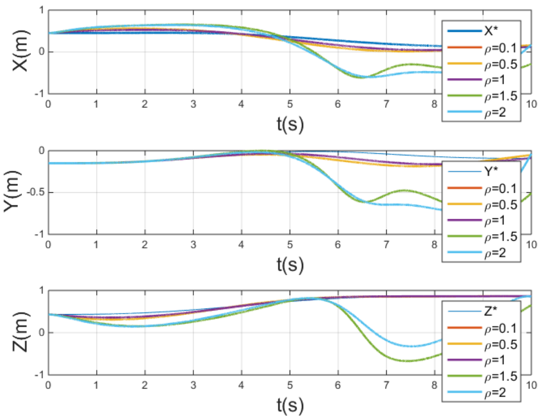


Figure 4. Changing step factor ρ

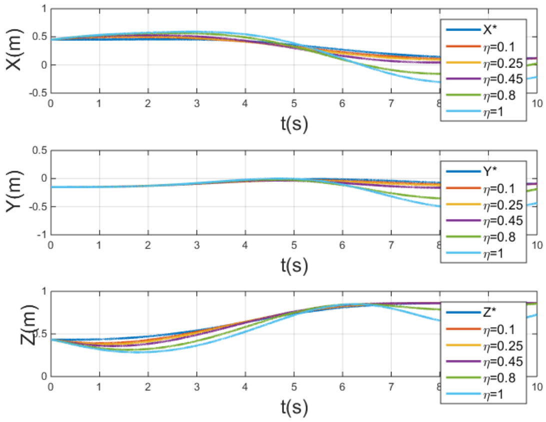


Figure 5. Changing step factor η

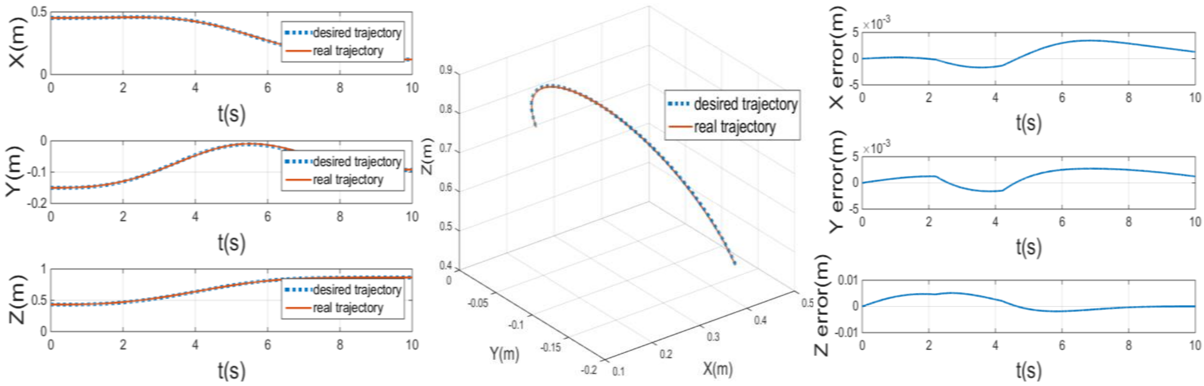
As shown in figure 4, the change of ρ will affect the control effect, and when ρ is over 1, the control effect of the controller will be worse. Compared with the desired trajectory, the actual trajectory changes sharply and the stability is poor. It can be seen from the control scheme (18) that, when other conditions are the same, the smaller ρ is, the bigger $\Delta\boldsymbol{u}(k)$, and manipulator control's sensibility will increase, which means that manipulator will react strongly to minor changes in control input or even resulting in a larger fluctuation. When ρ becomes smaller, $\Delta\boldsymbol{u}(k)$ mainly depends on the value of $\widehat{\boldsymbol{\Phi}}_c(k)$, $\boldsymbol{y}^*(k+1)$ and $\boldsymbol{y}(k)$. $\Delta\boldsymbol{u}(k)$ is related to the desired trajectory, the actual trajectory and the structure of the system, so satisfying control effect can be achieved.

As shown in figure 5, the change of η will affect the control effect, and when η is big, the control effect of the controller will be worse. It can be seen from the control scheme (15) that, when other conditions are the same, the bigger η is, the smaller $\Delta\widehat{\boldsymbol{\Phi}}_c(k)$, then $\Delta\boldsymbol{u}(k)$ will increase, which means that manipulator will react strongly to minor changes in control input.

From the control scheme, it can be concluded that the adjustment of the step factor can be used to modify the control effect without affecting the overall trend.

4.2. External disturbance

During simulation process, a short time disturbance, τ_1=[2 2 2 0 0 0]kg·m, is applied to the drive torque in 2-4s. By observing the curve of actual and desired position, and comparing with the curve without disturbance, the anti-interference performance of the Puma560 manipulator under the MFAC scheme is obtained.



(a) (b) (c)

Figure 6. Simulation results with disturbance τ_1

In the Cartesian space, the influence of disturbance $\boldsymbol{\tau_1}$ on the Puma560 manipulator is observed. The simulation results are shown in figure 6. When $\boldsymbol{\tau_1}$ is applied, the motion of end effector is still stable, and follows the desired trajectory. It can be seen from the coordinate error curve that in the 2-4s, $\boldsymbol{\tau_1}$ produces a small mutation, and the rate of change is not continuous. Therefore, the application of the disturbance, $\boldsymbol{\tau_1}$, has a small interference effect on the motion of the end actuators, but does not affect the smooth motion of the end actuators following the desired trajectory. Though being disturbed by $\boldsymbol{\tau_1}$, the controller can still realize the motion control of the Puma560 manipulator, which shows the robustness of the MFAC algorithm.

4.3. Simplifying the dimension of controller input signal

Considering that the actual joint rotation $\boldsymbol{q}$ is the main factor that determines the trajectory of manipulator, the actual joint speed $\dot{\boldsymbol{q}}$ and acceleration $\ddot{\boldsymbol{q}}$ can be obtained by differential calculation of $\boldsymbol{q}$, so we consider reducing the dimension of input signal of controller to reduce the dimension to satisfy the control effect and improve the efficiency of operation at the same time.

Take $\boldsymbol{q}$,$\dot{\boldsymbol{q}}$,$\boldsymbol{q}^*$,$\dot{\boldsymbol{q}}^*$ as input signals of the controller. The corresponding calculation and processing of these signals are carried out in the controller by using the MFAC scheme. The driving torque of each joint is obtained as the output signal from the controller. The actual input signal dimension of the controller is 12 dimensions, and the desired input signal dimension is 12 dimensions.

The simulation results are shown in figure 7.

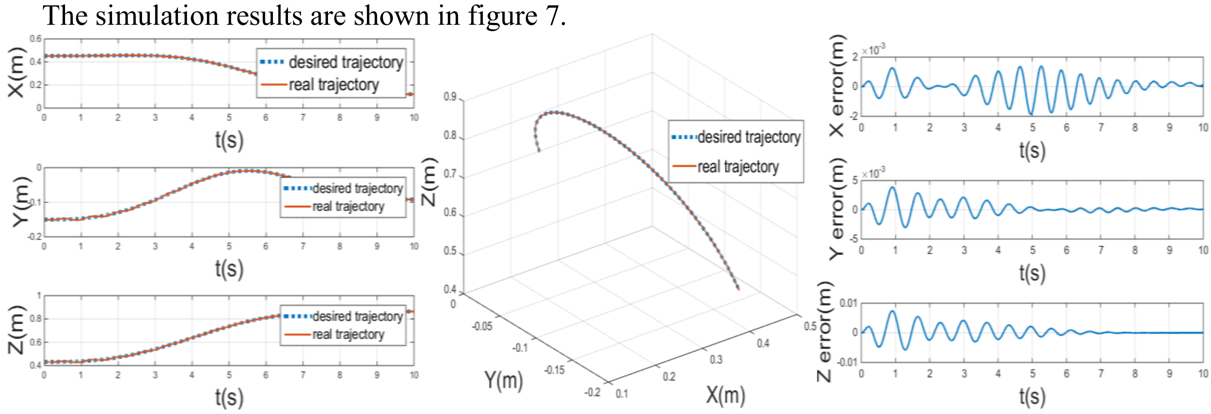


Figure 7. Simulation results with $y = [q; \dot{q}], y^* = [q^*; \dot{q}^*]$

As can be seen from figure 7, the joint rotation signals and speed signals are used as input, and the calculation process is carried out in the controller. The drive torque, as the output of the controller, can drive manipulator to track the desired trajectory well. Compared with the system with joint rotation, speed and acceleration as the input signal of the controller, the tracking error of end actuator is small, and the controller under the simplified method can guarantee the control effect of the original system.

The simulation time of the original system is 8.816931s while the simulation time of the simplified system is 5.189842s. Therefore, simplifying the input signal dimension of the controller can improve the computing speed to some extent.

Using the signal of joint rotation and velocity as the input signal of the controller, the operation speed can be improved and the control effect can also be kept.

5. Conclusion

In this paper, the Puma560 manipulator is used as the research object, and the control method of the manipulator's motion control is studied. The model-free adaptive control method is used to control the motion of the manipulator, which can avoid the unmodeled dynamics caused by the uncertainty of the structure and the unstructured uncertainty. The controller is designed by using the input and output data of the system. During this process, no mathematical model, parameter structure, or other related information of the system are required. Robustness and adaptability of the system can be guaranteed.

Acknowledgments

This work is supported by Fundamental Research. Special Funds for Central Universities (20826041A4133).

References

[1] Z. S. Hou and S. T. Jin, Model Free Adaptive Control: Theory and Applications. Boca Raton, FL, USA: CRC Press, 2013.
[2] D. M. Jia, Manipulator control based on iterative sliding mode algorithm (Master Thesis) Zhengzhou University, Zhengzhou, Henan, China, 2011.
[3] Y.Dai, Research on Intelligent PID control method for Robot Manipulators. (Master Thesis) Shengyang Ligong University, Shengyang, Liaoning, China, 2015..
[4] X. F. Wang, X. Li, J. H. Wang, Active interaction exercise control of exoskeleton upper limb rehabilitation robot using model-free adaptive methods. Acta Automatica Sinica, 2016, 42(12): 1899-1914.
[5] Y. Wang, Y. Q. Yan and L. B. Song, .Study on control method of modular mobile robot based on MFA theory. Journal of Mechanical & Electrical Engineering, Apr. 2012: 400-403
[6] Z. S. Hou and S. T. Jin, “A novel data-driven control approach for a class of discrete-time nonlinear systems,” IEEE Trans. Control Syst. Technol., vol.19, no.6, pp. 149–1558, Nov. 2011.