The Observer-based Neural Network Adaptive Robust Control of Underwater Hydraulic Manipulator

Yihong Xu
Mechanical Engineering Department
Zhejiang University
Hangzhou, China 310007

Yaoyao Wang
Mechanical Engineering Department
Zhejiang University
Hangzhou, China 310007

Xiaoxu Cao
Mechanical Engineering Department
Zhejiang University
Hangzhou, China 310007

Linyi Gu
Mechanical Engineering Department
Zhejiang University
Hangzhou, China 310007

***Abstract*—Hydraulic system is widely used in industry where large actuation forces are needed such as electro-hydraulic positioning system, active suspension control, and so on. One of the main problem in hydraulic system is the nontriviality, for instance, the deadbands in valve operation, frictions of links, saturations, pressure dynamics, etc. both system unknown nonlinearities and external disturbances should be considered in controller design. For underwater hydraulic manipulator, because of the environment of sea and the large nonlinearities of the hydraulic manipulator system, both system unknown nonlinearities and external disturbances should be considered in controller design. A lot of control methods have been proposed to solve the problem properly, such as sliding mode control, adaptive control, fuzzy adaptive control, adaptive robust control, ect. However, due to limited knowledge about hydraulic manipulator system, it is hard to precisely estimate the nonlinearities for a possible higher performance. In this paper, a neural network adaptive robust control method is proposed to help control an underwater hydraulic manipulator. In order to achieve better performance using adaptive robust control strategy, unknown nonlinearities in the system are approximated by the outputs of radial basis function neural network (RBFNNs). A projection law has been used to bound the NN weights when they are tuning on-line to avoid possible adaptive divergence, and the tracking transient performance is guaranteed by robust control law. Simulation results illustrate that manipulator has a good tracking performance.**

***Keywords*—component; formatting; style; styling; insert (key words)**

I. INTRODUCTION (*Heading 1*)

Hydraulic system has been widely used in industry which large actuation force are needed such as hydraulic suspension system, hydraulic crane, hydraulic wrench and so on. For work-class deep-sea ROV, the hydraulic manipulator is the key component ensuring the task would be accomplished. There have been a lot of control methods proposed for robot manipulators which are often driven by motors. For example, sliding mode control was applied in [1], and adaptive control was used in [2]. The adaptive robust control [3] which was proposed by Yao and Tomizuka combined the adaptive control and robust control. These are all effective control methods to help design controllers for manipulator. However, for underwater hydraulic manipulator, the environment of deep-sea is quite complex with large external disturbances, hydraulic manipulator itself is much more complicate than electric motor driven robot manipulators because of the strong nonlinearities and system parameter uncertainties. What's more, some of the system state may not be able to obtain, because there would be hard for deep sea manipulators to mount a lot of sensors. So most model-based algorithms which are listed above would be more difficult to design a high performance controller for such a system. To deal with these problems, [4,5] proposed model-free neural network(NN) control algorithm.

Neural network is one of the most widely used intelligent computation approaches since it has the virtue of inherent learning and the ability to approximate of nonlinear functions. Especially the on-line learning ability of adaptive neural network algorithm makes NN a popular method in adaptive control. Gong and Yao first generally build a frame of neural network based adaptive robust control [7]. The form allows unknown nonlinearities existing in both system model and the virtual input channel of each intermediate step of backstepping procedure, and the unknown nonlinearities could include nonrepeatable nonlinearities such as external disturbances as well. Radial basis function neural network (RBFNN) is a special multilayer neural network. Because of the Gaussian function as its excitation function, it has a lot of virtue such as

978-0-933957-43-5 ©2015 MTS

it has a very fast rate of convergence, less computing and has no problem of local minimum.

In this paper, we mainly focus on the control of the elbow joint of underwater hydraulic manipulator. A adaptive robust controller is designed following backstepping procedure. And with RBF neural network is applied, a model-free RBFNN adaptive law with projection bound is built. The RBFNN is adapted to approximate the uncertainties and external disturbance of the system. Lacking of sensors for angular velocity, a high gain observer is used to detect the angular velocity of the joint to realize the output feedback of the system.

This paper is organized as follows. In section 2, the structure of RBF neural network is briefly introduced. In section 3, the model of valve controlled swing cylinder is built. In section 4, the adaptive robust controller and RBF adaptive law is proposed. Finally the simulation of the system and conclusion is shown in section 5&6.

II. STRUCTURE OF RBF NEURAL NETWORK

RBF neural network often has a three layer structure, which includes input layer, hidden layer and output layer. The main difference between RBF and multilayer neural network is that RBF use radial basis function which is usually Gaussian function as activation function from input layer to hidden layer, that is

$$\varphi_j(r) = exp\left(-\frac{\|x-r_j\|^2}{2\sigma_j^2}\right), \sigma > 0 \tag{1}$$

Where r_j is the center of jth neuron, and σ_j is the width of the jth neuron.

And the output of the RBF neural network is the linear weighting function

$$y = W^T\varphi = \sum_{j=1}^{K} \omega_{ji}\varphi_j \ , i = 1, \cdots, n \tag{2}$$

where ω is the output weight from hidden layer to output layer, n is the number of neuron nodes.

III. THE DYNAMIC MODEL OF THE ELBOW OF HYDRAULIC MANIPULATOR

The elbow model shown in Fig 1 which is considered is mainly based on[6]. The outer revolute joint is consist of cylinder block(1) and piston(4), the inner revolute joint is consist of piston(4) and output shaft(3). When chamber A and B alternately connected to high pressure and low pressure, the piston will move back and forth with helical motion. These features of swing cylinder make it quite similar to the model of valve-controlled double-rod cylinder. We aim to have the angle of swing cylinder track the trajectory we designed.

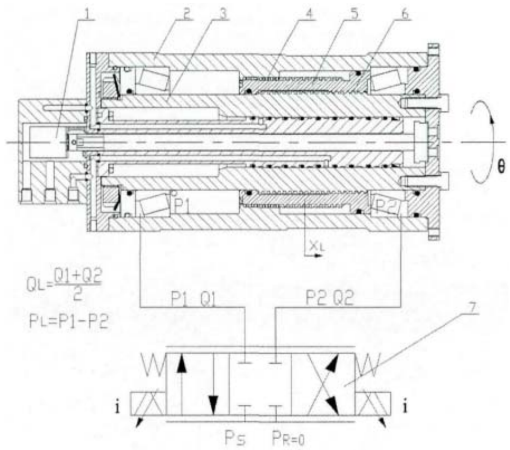


Fig1. Valve controlled swing cylinder (elbow of hydraulic manipulator)

The dynamics of the inertia of load can be written as

$$I(\theta)\ddot{\theta} + b(\theta, \dot{\theta})\dot{\theta} = \tau + \tau_d; \tag{3}$$

$$\tau = K_T A P_L \tag{4}$$

where

θ	the angular of swing cylinder
$I(\theta)$	the inertia of swing cylinder
$b(\theta, \dot{\theta})$	viscous coefficient of swing cylinder
τ	the torque applied on swing cylinder
τ_d	external disturbances
K_T	combined coefficient of load pressure and the torque which is determined by the mechanical structure of swing cylinder
A	the ram area of chambers
P_L	the load pressure which is the difference pressure value of two chambers

The hydraulic flow dynamics of swing cylinder can be described as follow

$$\frac{V_t}{4\beta_e}\dot{P}_L = -AK_\theta\dot{\theta} - C_m P_L + Q_L \tag{5}$$

where

V_t	total control volume of the chamber
β_e	effective bulk modulus
K_θ	the coefficient of the angular and axial displacement of swing cylinder
C_m	the leakage of chamber, which is a very small value and usually to be considered as 0 especially for a swing cylinder
Q_L	flow rate through the valve

Q_L is the function of displacement of spool, which can be modeled by

$$Q_L = K_q x_v \sqrt{P_s - sgn(x_v)P_L} \tag{6}$$

where

K_q	the flow gain coefficient
x_v	the displacement of valve
P_s	the system pressure

Valve dynamics is often relatively faster than the system. So ignoring it, we can use the displacement of the valve x_v as input u.

$$x_v = K_v u \tag{7}$$

IV. CONTROLLER DESIGN

The controller design will apply the backstepping procedure based on Lyapunov functions. According to the [6], as we could directly acquire the expression between the input and the flow rate

$$Q_L = K_q K_v u \sqrt{P_s - sgn(Q_L) P_L} \tag{8}$$

We could assume the flow rate as the system input. Define the state variables as $x = [x_1, x_2, x_3] \triangleq [\theta, \dot{\theta}, P_L]$, And rewrite the equations in the state space form as follow

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = \frac{K_T A}{I(\theta)} x_3 - \frac{b(\theta,\dot{\theta})}{I(\theta)} x_2 + \frac{\tau_d}{I(\theta)} \\ \dot{x_3} = \frac{4\beta_e}{V_t}[-A K_\theta x_2 - C_m x_3 + Q_L] \end{cases} \tag{9}$$

The above system can be considered as a semi-strict feedback form as [7], that is:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = b_2(x_1) x_3 + f_2(x_1, x_2) + \Delta_2 \\ \dot{x_3} = f_3(x_1, x_2, x_3) + b_3(x_1) u \end{cases} \tag{10}$$

Where $b_2(x_1) = \frac{K_T A}{I(\theta)}$, $f_2(x_1, x_2) = -\frac{b(\theta,\dot{\theta})}{I(\theta)} x_2$, $\Delta_2 = \frac{\tau_d}{I(\theta)}$, $f_3(x_1, x_2, x_3) = -\frac{4\beta_e}{V_t}[A K_\theta x_2 + C_m x_3]$, $b_3(x_1) = \frac{4\beta_e}{V_t}$,in which we assume that $\frac{K_T A}{I(\theta)}$ as a constant that only affected by the angular displacement x_1, also we consider $\frac{4\beta_e}{V_t}$ was mainly affected by the state of x_1.

A. Observer design

Since state x_2, that is the velocity of the angular could not be measured, to apply the RBF neural network to help controller design, an observer should be designed to develop an output feedback control. According to [10], because the form of $b_2(x_1)$ and $f_2(x_1, x_2)$ are unknown functions, the design method of a nonlinear Luenberger-type observer can't be used. In [11] a high-gain observer is applied here.

Suppose the basic system which is considered has the following form

$$x^{(n)} = f(x) + b(x) u(t) + \Delta \tag{11}$$

The above system should satisfy three assumptions as mentioned in [7]. Suppose there exist a following linear system

$$\begin{cases} \varepsilon \dot{\xi}_1 = \xi_2 \\ \vdots \\ \varepsilon \dot{\xi}_{n-1} = \xi_n \\ \xi_n = -b_1 \xi_n - \cdots - b_{n-1} \xi_2 - b_1 + y \end{cases} \tag{12}$$

where y has its first nth derivatives which are bounded, ε is a small positive constant, and $b_1, \cdots, b_{n-1}$ should satisfies the condition of that $s^n + b_1 s^{n-1} + \cdots + b_{n-1} s + 1$ is Hurwits. Then, there must be positive constant D_k, where $k = 2, \cdots, n$

$$\begin{cases} \varepsilon \psi^{(k+1)} = y^{(k)} - \xi_{k+1} / \varepsilon^k \\ \varepsilon D_{k+1} \geq \left| y^{(k)} - \xi_{k+1} / \varepsilon^k \right| \end{cases} \tag{13}$$

where $\psi = \xi_n + b_1 \xi_{n-1} + \cdots + b_{n-1} \xi_1$, $\bullet^{(k)}$ is represented the kth derivative.

Define a concise tracking error metric as [7,8]

$$s(t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x}(t) \tag{14}$$

where $\tilde{x}(t) = x(t) - x_d(t)$, λ is a positive constant, and the equation could also be rewritten as $s(t) = \lambda^T \tilde{x}(t)$. So $s(t) = 0$ defines a hyperplane in the $\mathcal{R}^n$ state space, $\tilde{x}(t)$ will finally exponentially converges to zero. And we also have the state estimation $\hat{x} = \left[x, \frac{\xi_2}{\varepsilon}, \cdots, \frac{\xi_n}{\varepsilon^{n-1}}\right]^T$. Proof referred to the Lemma of 6.2 of [13].

B. Backstepping procedure

First we consider the following subsystem, treat the state x_3 as virtual input α_2, x_1 is the state that we can measure, and x_2 is the state that we need observer to obtain. Thus we can solve the subsystem as the first step of backstepping procedure. By using the NNARC proposed as [10], we can define the following variables

$$(\text{subsystem}) \begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = b_2(x_1) x_3 + f_2(x_1, x_2) + \Delta_2 \end{cases} \tag{15}$$

$$\hat{x} = \left[x_1, \frac{\xi_2}{\varepsilon}\right]^T \tag{16}$$

$$\zeta = \hat{x} - x_d \tag{17}$$

$$\psi = \left[0, \ \psi^{(2)}\right]^T \tag{18}$$

$$\hat{s} = \lambda^T \zeta = s - \varepsilon \lambda^T \psi \tag{19}$$

And we propose the following virtual input α_2 and error equations.

$$\alpha_2 = \alpha_{2a} + \alpha_{2s} \tag{20}$$

$$\alpha_{2a} = -\frac{1}{\hat{b}_2}\left(\hat{f}_2 + \hat{a}_r\right) \tag{21}$$

where $a_r = \lambda \dot{z}_1 - \ddot{x}_d$, $\hat{a}_r = a_r - \varepsilon \ddot{\psi}$, α_{2a} is a term that use RBF neural network to compensate the system, α_{2s}will be considered later.

$$\dot{\hat{s}} = \left[f_2(x) - {W_{f2}}^T \varphi_{f2} + d_{f2}\right] - {\widetilde{W}_{f2}}^T \varphi_{f2}(\hat{x}) + \left[b_2(x) - {W_{b2}}^T \varphi_{b2} + d_{b2}\right] - {\widetilde{W}_{b2}}^T \varphi_{b2}(\hat{x}) \alpha_{2a} + b_2(\hat{x}) \alpha_{2s} + b_2(\hat{x}) z_3 + \varepsilon(\ddot{\psi} - \ddot{\psi}) + \Delta_2 \tag{22}$$

The adaptation laws can be derived

$$\tau_{f2}^{(1)} = \hat{s} \varphi_{f2} \tag{23}$$

$$\tau_{b2}^{(1)} = \hat{s} \varphi_{b2} \alpha_{2a} \tag{24}$$

Here we consider the α_{2s} term. We usually considered it consists of two parts $\alpha_{2s} = \alpha_{2s1} + \alpha_{2s2}$. $\alpha_{2s1}$is a term of state feedback $\alpha_{2s1} = -\frac{k_{2s1}}{b_{2l1}} \hat{s}$. While α_{2s2} is a robust term satisfying the conditions as [9,10] proposed.

$$\begin{cases} \hat{s}[f_2(x) + a_r(t)] + b_2(x) \alpha_{2a} + b_2(x)\ \alpha_{2s2} + \Delta \leq \epsilon_{s0} \\ \hat{s}\ \alpha_{2s2} \leq 0 \end{cases} \tag{25}$$

where

$$\epsilon_{s0} = \epsilon_{ss} + \left(\frac{\varepsilon \lambda (\ddot{\psi} - \ddot{\psi})}{D}\right)^2 \epsilon_6 \tag{26}$$

$$\epsilon_{ss} = \left|1+\frac{\left|\|\rho_{f2}\|_2-\|\hat{\rho}_{f2}\|_2\right|}{\|\hat{\rho}_{f2}\|_2}\right|^2\epsilon_1+\left|1+\frac{\left|\|\rho_{b2}\|_2-\|\hat{\rho}_{b2}\|_2\right|}{\|\hat{\rho}_{b2}\|_2}\right|^2\epsilon_2+d_{f2}\epsilon_3+d_{b2}\epsilon_4+\epsilon_5\|d\|_\infty^2 \tag{27}$$

In which $\epsilon_i, i=1,\cdots,6$ and D are positive constant, and $\rho_\bullet$ represent the bound of $\bullet$.

The method of choosing α_{2s2} is introduced in [10]. It is showed as below

$$\alpha_{2s2} = -k_{2s2}\hat{s} \tag{28}$$

Where k_{s2} is a large enough nonlinear gain satisfying the conditions

$$k_{s2} \geq \sum_{i=1}^{6}\frac{h_i^{\,2}}{4\epsilon_i} \tag{29}$$

with $h_1 \geq \left(\frac{1}{\sqrt{b_{2l1}}}\right)\left\|\varphi_{f2}(\hat{x})\right\|_2\left(\left\|\hat{w}_{f2}\right\|_2+\left\|\hat{\rho}_{f2}\right\|_2\right)$, $h_2 \geq \left(\frac{1}{\sqrt{b_{2l1}}}\right)\|\varphi_{b2}(\hat{x})\|_2(\|\hat{w}_{b2}\|_2+\|\hat{\rho}_{b2}\|_2)|\alpha_{2a}|$, $h_3 \geq (1/\sqrt{b_{2l1}})\delta_{f2}$, $h_4 \geq (1/\sqrt{b_{2l1}})\delta_{b2}|\alpha_{2a}|$, $h_5 \geq (1/\sqrt{b_{2l1}})\delta_{sys}$, $h_6 \geq (1/\sqrt{b_{2l1}})D$. Since $\varphi(\hat{x})$ is Gaussian Function, both $\left\|\varphi_{f2}(\hat{x})\right\|_2$ and $\|\varphi_{b2}(\hat{x})\|_2$ satisfy $\|\varphi(\hat{x})\|_2 \leq 1$. Thus we choose h_1 as $h_1 = 2\sqrt{r_1}\left\|\hat{\rho}_{f2}\right\|_2$, with r_1 is the number of neurons in adaptation of f_2, and also we can choose h_2 as the same way.

Afterwards, we continue to design the controller as backstepping procedure. We aim to make state x_3 track the virtual input α_2. So define a tracking error $z_3 = x_3 - \alpha_2$, from the third equation of the system(10)

$$\dot{z}_3 = \dot{\alpha}_2 + f_3(x_2,x_3) + b_3(x_1)u \tag{30}$$

From [7], we could get tuning function for this subsystem and the input u as the same way we have done above.

$$u = \alpha_3 = \alpha_{3a}+\alpha_{3s} = \frac{1}{\hat{b}_3}\left[-\hat{f}_3+\frac{\partial\alpha_2}{\partial\hat{x}_2}\hat{f}_2+\frac{\partial\alpha_2}{\partial x_1}x_2+\frac{\partial\alpha_2}{\partial\hat{x}_2}\hat{b}_2x_3+\frac{\partial\alpha_2}{\partial t}-\hat{b}_2(\hat{x}_2-\dot{x}_d)\right]-(k_{3s1}+k_{3s2})z_3 \tag{31}$$

$$\tau_{f3}^{(2)} = z_3\varphi_{f3} \tag{32}$$

$$\tau_{f2}^{(2)} = \tau_{f2}^{(1)} - \frac{\partial\alpha_2}{\partial x_2}z_3\varphi_{f2} \tag{33}$$

$$\tau_{b3}^{(2)} = z_3\varphi_{b3}\alpha_{3a} \tag{34}$$

$$\tau_{b2}^{(2)} = \tau_{b2}^{(1)} - \frac{\partial\alpha_2}{\partial\hat{x}_2}z_3\hat{s}\varphi_{b2} + z_3\hat{s}\varphi_{b2} \tag{35}$$

Thus the weighting adaptation law should be

$$\dot{\hat{w}}_{f3} = \mathrm{Proj}_{\hat{w}_{f3}}(\Gamma_{f3}z_3\varphi_{f3}) \tag{36}$$

$$\dot{\hat{w}}_{b3} = \mathrm{Proj}_{\hat{w}_{b3}}(\Gamma_{b3}z_3\varphi_{b3}\alpha_{3a}) \tag{37}$$

$$\dot{\hat{w}}_{f2} = \mathrm{Proj}_{\hat{w}_{f2}}(\Gamma_{f2}(\tau_{f2}^{(1)} - \frac{\partial\alpha_2}{\partial\hat{x}_2}z_3\varphi_{f2})) \tag{38}$$

$$\dot{\hat{w}}_{b2} = \mathrm{Proj}_{\hat{w}_{b2}}(\Gamma_{b2}(\tau_{b2}^{(1)} - \frac{\partial\alpha_2}{\partial\hat{x}_2}x_3\hat{s}\varphi_{b2} + z_3\hat{s}\varphi_{b2})) \tag{39}$$

Where $proj(\bullet)$ is the bounded projection of $\bullet$ which means

$$\mathrm{proj}_{\hat{\theta}}(\bullet) = \begin{cases} 0 & \text{if } \hat{\theta} \geq \theta_{max} \text{ and } \bullet > 0 \\ 0 & \text{if } \hat{\theta} \leq \theta_{max} \text{ and } \bullet < 0 \\ \bullet & \text{else} \end{cases} \tag{40}$$

V. SIMULATION AND RESULTS

To test the proposed neural network adaptive robust control for the swing joint, we build up a simulation in Matlab/Simulink for the swing cylinder. The parameters in the simulation are as follows.

$Ps = 10\mathrm{Mpa}, I = 0.022\mathrm{kg\cdot m^2}, A = 0.00481\mathrm{m^2}$,
$\beta_e = 7\times10^8\mathrm{N/m^2}, V_t = 7.104\times10^{-6}\mathrm{m^3}$,
$K_t = 0.0078\mathrm{m}, K_\theta = 0.01042\mathrm{m/rad}, b = 20\mathrm{N\cdot m\cdot s/rad}$

Before we start to build up the simulation system, we should scale the parameters. Define the coefficient $Sc = 5\times10^6$ so that $Ps = Sc\times\overline{Ps}$. Use $\overline{Ps}$ as the state x_3, and rebuild the system, than the state are all in the domain of [-2,2]. The other coefficients are chosen as follows, $\lambda = 100, k2 = 30, k3 = 5, \Gamma_1 = \Gamma_2 = \Gamma_3 = \Gamma_4 = 100$.

From (8), we get an expression of input voltage u with respect to flow rate Q_L, that is

$$u = (K_qK_v\sqrt{P_s - sgn(Q_L)P_L})^{-1}Q_L$$

Restricted by physical condition of input voltage, the input u is confined within the range of [-1.5v,1.5v]

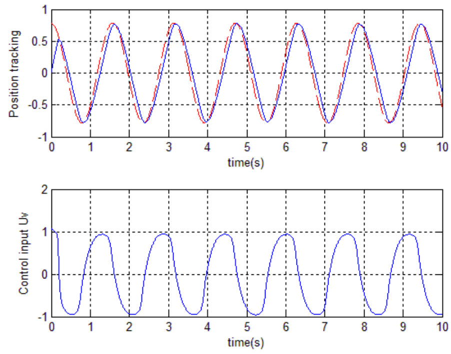
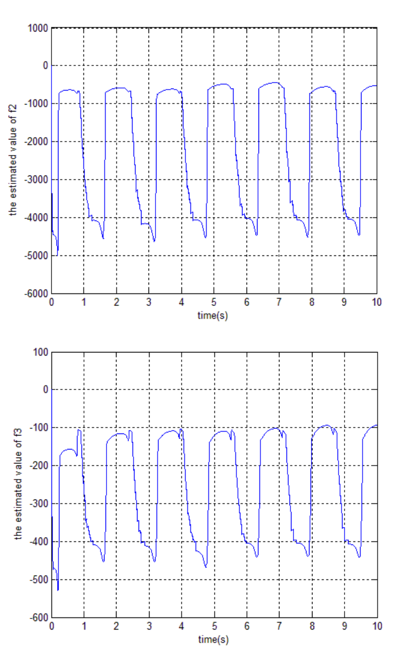


Fig 2. The position tracking of the swing cylinder and the control input Uv



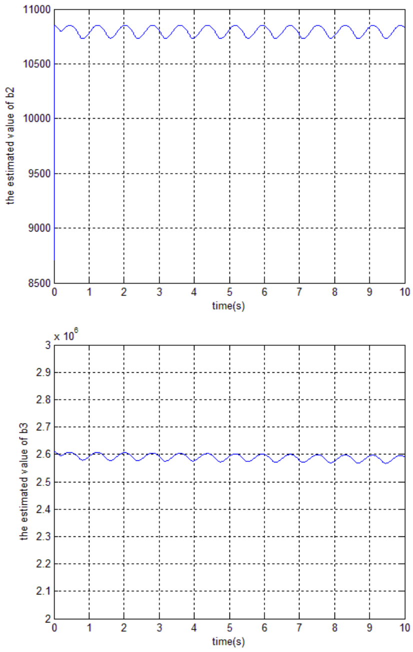


Fig 3. The estimation of functions f2, b2, f3, b3 with sinusoidal input

As it is shown in Fig 2, we can see the angular position tracking is perfect, and the control input have no large overshoot and chartering even there is a added torque $\tau_d = 30 + 10 \times sin\,(\frac{\pi}{2}t)$ as external disturbance input. Fig 3 illustrates the result of estimation of functions in the system. The result shows that the RBFNN adaptive law affected the functions to estimate the target function. And the estimated value can be unable to converge to the real value happen if the system doesn't satisfy the persistent condition. However, what we care much more about, the stability and asymptotical tracking have been obtained quite well.

REFERENCES

[1] Su. C. Y., Leung. T. P. A Sliding Mode Controller with Bound estimation for robotic manipulators. IEEE Transactions on Robotics and Automation, 1993,9(2):208~214

[2] JJE Slotine, W. Li On the adaptive control of robot manipulators The International Journal of Robotics Research, 1987 vol. 6 no. 349-59

[3] B Yao, M Tomizuka, Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form Automatica, Volume 33, Issue 5, 1997, Pages 893 – 900

[4] FL Lewis Neural network controlof robot manipulators IEEE Intelligent Systems, 1996, Issue No.03 vol.11 pp: 64-75

[5] T. R. Sun, H. L. Pei, Y. P. Pan, H. B. Zhou, and C. H. Zhang, "Neural network-based sliding mode adaptive control for robot manipulators," Neurocomputing, vol. 74, no. 14/15, pp. 2377–2384, Jul. 2011

[6] Gaosheng Luo, Jiawang Chen, Linyi Gu, Research on a Nonlinear Robust Adaptive Control Method of the Elbow Joint of a Seven-Function Hydraulic Manipulator Based on Double-Screw-Pair Transmission, Mathematical Problems in Engineering Volume 2014 (2014), Article ID 184968

[7] J. Q. Gong and B.Bin Yao, "Neural network adaptive robust control of nonlinear systems in semi-strict feedback form," Automatica, vol. 37,pp. 1149 –1160, 2001

[8] Sanner, R. M., and Slotine, J.-J. E. Gaussian Networks for Direct Adaptive Control, IEEE Trans. Neural Netw.,3(6), pp. 837–863. 1992

[9] Gong, J. Q., and Yao, B. Neural Network-Based Adaptive Robust Control of a Class of Nonlinear Systems in Normal Form. In Proceedings of the American Control Conference, June 28–30 , Chicago, IL, PP. 149–1423 , 2000

[10] Gong.J.Q., and Yao.B. 2006 Output Feedback Neural Network Adaptive Robust Control With Application to Linear Motor Drive System. Journal of Dynamic Systems, Measurement, and Control, 28, 227-235.

[11] Ge, S. S., Hang, C. C., and Zhang, T. Neural-Based Direct Adaptive Neural Network Control of Nonlinear Systems by State and Output Feedback, IEEE Trans. Syst., Man, Cybern., Part B: Cybern., 29(6), pp. 818–828, 1999

[12] Juntao Fei, Hongfei Ding, Adaptive sliding mode control of dynamic system using RBF neural network, Nonlinear Dyn (2012) 70:1563–1573

[13] Behtash, S., 1990, Robust Output Tracking for Nonlinear Systems, Int. J.Control, 51(6), pp. 1381–1407.