Adaptive Backstepping Control for MAPK Cascade Models Using RBF Neural Networks

Kyriakos G. Vamvoudakis *Student Member, IEEE*, and Manolis A. Christodoulou *Senior Member, IEEE Technical University of Crete, 73100 Chania Crete Greece email:* manolis@ece.tuc.gr, kyriakos@arri.uta.edu

Abstract— **In this paper, an Adaptive Backstepping Neural Network control approach is used for a class of affine nonlinear systems which describe the Mitogen Activated Protein Kinase (MAPK) cascade models in the strict feedback form. We consider some of forms of the MAPK cascade [4]. The close loop signals are semiglobally uniformly ultimately bounded and the output of the system is proven to follow a desired trajectory. Simulation results are presented to show the effectiveness of the approach proposed in order to control the MAPK output.**

I. INTRODUCTION

NOWADAYS MAPK cascade models are being used to control the cell division processes and are based on the kinetic properties of kinases and phosphatases in specific signaling pathways. Biologists work is supported by various remarks from recent Nobel Prize winners. Because of that, recently interest in modeling of biological systems has been increased. In the present work we focus on the control of signals, the kinases produce as partial outputs (virtual control inputs) via a well known adaptive backstepping technique [1] through an appropriate selection of the controller. The system exhibits unknown nonlinearities. We use adaptive control [13] in order to track the desired output. We use specific adaptation laws in order to reduce uncertainty. This method can be used in drug discovery and sickness therapy, especially in personalized medicine (i.e., for a specific disease, for a specific person, the appropriate medicine). It is obvious that we do not need to know interconnections between the signaling pathways. We only need an appropriate input (control input) to make the output to follow the desired behavior.

The kinases and phosphatases of our model have the following properties:

i. Each kinase can be found in an active and an inactive form.
ii. Activation of kinases takes place by phosphorylation.

When the kinases are in their active forms they may phosphorylate other kinases and inactivation of the kinases takes place by dephosphorylation catalyzed by phosphatases which are sequentially active. Effects of multiple phosphorylations are neglected and we do not consider the action of scaffolds and adaptors.

Assumption: Initial activation of kinases occurs by their interaction with an external receptor $u(t)$.

II. PROBLEM ANALYSIS

A. System Description

The MAPK cascade model is described by the following set of differential equations [4]-[6]:

$$\frac{dX_i}{dt} = u(t)\delta_i \tilde{X}_i + \sum_{j \neq i}^{n} a_{ij} \tilde{X}_i X_j - \beta_i X_i \qquad (1)$$

where $\tilde{X}_i$ and X_i are the concentrations of the inactive and active forms respectively, of the ith kinase. They can be related with each other with the following equation $\tilde{X}_i + X_i = C = const.$, where a_{ij} and β_i are the rate constants of kinases and phosphatases, and $u(t)$ is the time dependent concentration of the receptor. The receptor converts the inactive input kinases into active. The rate constant δ_i is chosen such that: $\delta_i = 1$ for input kinases and $\delta_i = 0$ otherwise. The above hypothesis is made because the receptor affects only the input kinases.

Special cases of (1) can be expressed in (or transformed to) the following nonlinear state space form:

$$\begin{aligned} x_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, 1 \le i \le n-1 \\ x_i &= f_i(\bar{x}_i) + g_n(\bar{x}_n)u \;\; , n \ge 2 \\ y &= x_1 \end{aligned} \qquad (2)$$

where $\bar{x}_i = [x_1, x_2, \dots x_i]^T \in R^i, i = 1,\dots n, u \in R, y \in R$ are state variables, input and output respectively. Our purpose is to construct a specific adaptive Neural Network controller such that:

i) all the signals in the close loop remain semiglobally ultimately bounded
ii) the output signal y follows a desired trajectory signal y_d, with bounded derivatives up to $(m+1)th$ order.

In order to approximate some unknown nonlinearities we use Neural Networks [12], [14]. This approximation is guaranteed within some compact sets Ω.

Since $g_i(.)$, $i=1,...n$ are smooth functions, they are therefore bounded within some compact set. According to the previous we can make two assumptions.

Assumption 1: The signs of $g_i(.)$ are bounded for example there exist constants $g_{i1}(.) \geq g_{i0}(.) > 0$ such that, $g_{i1}(.) \geq |g_{i1}(.) \geq |g_{i0}(.), \forall \bar{x}_n \in \Omega \in R^n$. The $g_i(.)$ functions in the MAPK cascade are strictly positive because concentrations of kinases and phosphatases are positive numbers.

Assumption 2: There exist constants $g_{id}(.) > 0$ such that $g_i(.) \leq g_{id}(.)\ \forall \bar{x}_n \in \Omega \in R^n$.

B. RBF Neural Networks

Dynamical Neural Networks are well established tools used in the control of nonlinear and complex systems. We use RBF Neural Networks [9] in order to approximate the nonlinear functions of our systems [15]. The idea behind this is described fully at [2], [3], [8], [10], [11]. The RBF NN we use are of the general form $F(.) = \theta^{\mathrm{T}}\xi(.)$, where $\theta \in R^p$ is a vector of regulated weights and $\xi(.)$ a vector of RBF's. It has been shown that given a smooth function $\mathrm{F}:\Omega \rightarrow R$, where Ω is a compact subset of R^m (m is an appropriate integer) and $\varepsilon > 0$, there exists an RBF vector $\xi:\mathrm{R}^{\mathrm{m}} \rightarrow R^p$ and a weight vector $\theta^* \in R^p$ such that $\left|\mathrm{F(x)} - \theta^{*\mathrm{T}}\xi(\mathrm{x})\right| \leq \varepsilon\ \forall x \in \Omega.$ Here ε is called the network reconstruction error. The optimal weight vector is chosen as an appropriate value that minimizes the reconstruction error over Ω.

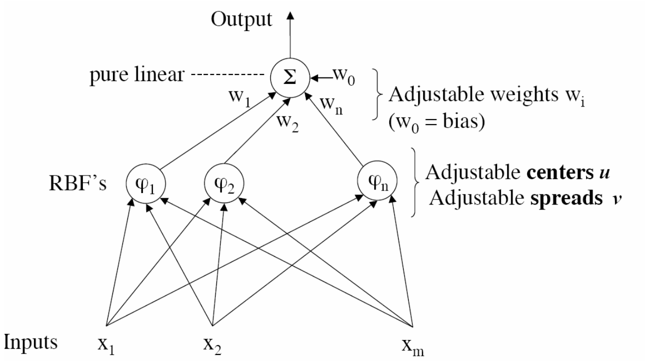
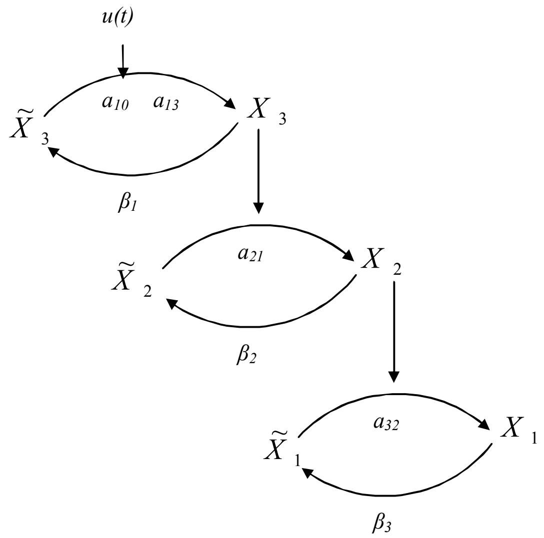


Fig1: Schematic representation of RMF Neural Networks

C. MAPK cascade models

In this paper we apply the backstepping technique to the following pathways consisting of three kinases and three phosphatases:

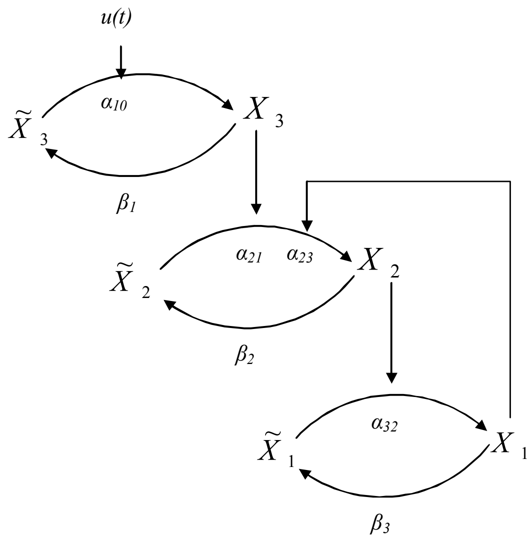
i)



The previous pathway is described by the following equations:

$$\dot{x}_3 = -\beta_1 x_3 + (c_1 - x_3)u(t)$$
$$\dot{x}_2 = -\beta_2 x_2 + a_{21}(c_2 - x_2)x_3$$
$$\dot{x}_1 = -\beta_3 x_1 + a_{32}(c_3 - x_1)x_2$$
$$y = x_1$$

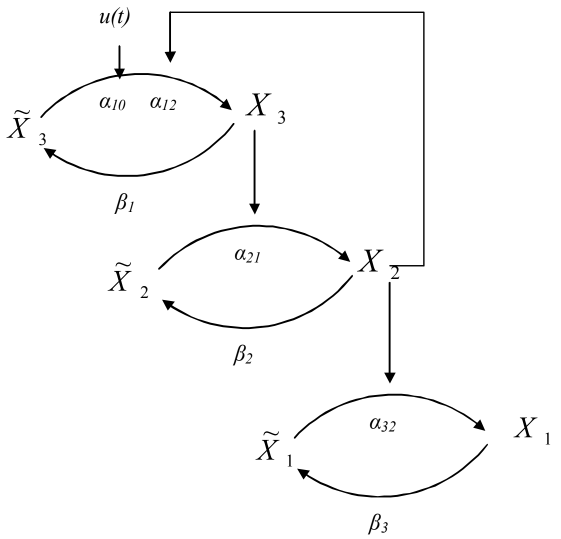
ii)



The previous pathway is described by the following equations:

$$\dot{x}_3 = -\beta_1 x_3 + (c_1 - x_3)u(t)$$
$$\dot{x}_2 = -\beta_2 x_2 + a_{23}(c_2 - x_2)x_1 + a_{21}(c_2 - x_2)x_3$$
$$\dot{x}_1 = -\beta_3 x_1 + a_{32}(c_3 - x_1)x_2$$
$$y = x_1$$

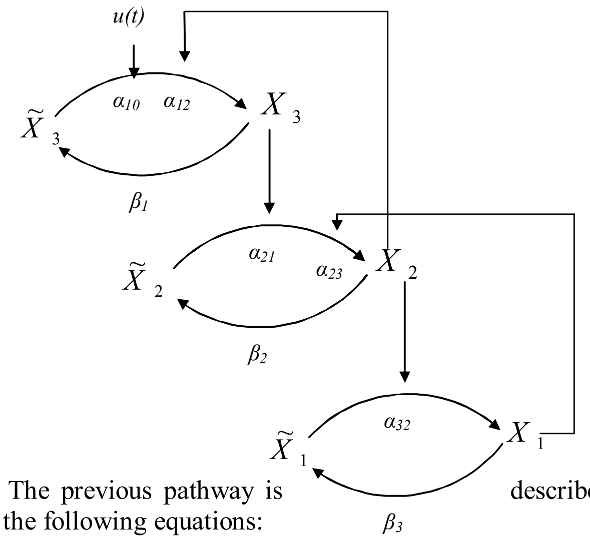
iii)



The previous pathway is described by the following equations:

$$\dot{x}_3 = -\beta_1 x_3 + a_{12}(c_1 - x_3)x_2 + (c_1 - x_3)u(t)$$
$$\dot{x}_2 = -\beta_2 x_2 + a_{21}(c_2 - x_2)x_3$$
$$\dot{x}_1 = -\beta_3 x_1 + a_{32}(c_3 - x_1)x_2$$
$$y = x_1$$

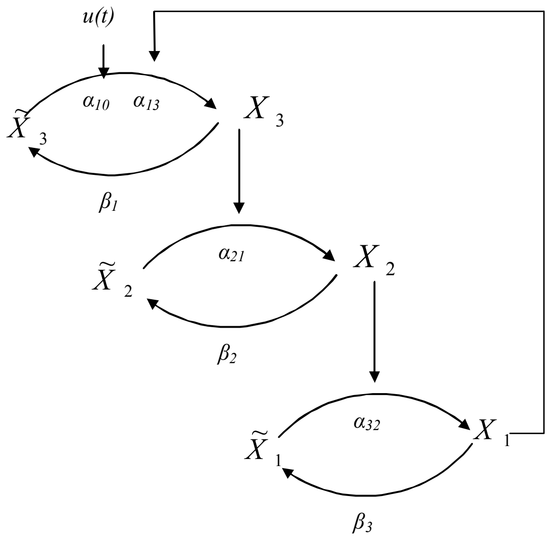
iv)



The previous pathway is described by the following equations:

$$\dot{x}_3 = -\beta_1 x_3 + a_{12}(c_1 - x_3)x_2 + (c_1 - x_3)u(t)$$
$$\dot{x}_2 = -\beta_2 x_2 + a_{23}(c_2 - x_2)x_1 + a_{21}(c_2 - x_2)x_3$$
$$\dot{x}_1 = -\beta_3 x_1 + a_{32}(c_3 - x_1)x_2$$
$$y = x_1$$

v)



The previous pathway is described by the following equations:

$$\dot{x}_3 = -\beta_1 x_3 + a_{13}(c_1 - x_3)x_1 + (c_1 - x_3)u(t)$$
$$\dot{x}_2 = -\beta_2 x_2 + a_{21}(c_2 - x_2)x_3$$
$$\dot{x}_1 = -\beta_3 x_1 + a_{32}(c_3 - x_1)x_2$$
$$y = x_1$$

C. Controller Design

In [16], a desired feedback control law was initially proposed for system (2) and Neural Networks are used to parameterize the desired feedback control law. Finally adaptation laws are used to tune the weights of neural networks for closed loop stability. In our paper we use the controller designed by Kaynak et al. [7]. The design procedure is described in 3 steps because in the MAPK cascade models above we have 3 states. Each backstepping stage results in a new virtual control design obtained from the preceding design stages. When the procedure ends, the feedback design for the control input is obtained, which achieves the original design objective.

Step1: In this step we want to make the error between x_1 and x_{1d} ($=y_d$) as small as possible.

The previous is described by the following equation:

$$e_1 = x_1 - x_{1d}$$
(3)

We take the derivative of e_1. After that we have:

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_{1d} \Rightarrow \dot{e}_1 = f_1(x_1) + g_1(x_1)x_2 - \dot{x}_{1d} \quad (4)$$

by using x_2 as the virtual control input. The previous equation can be changed by multiplication and division with $g_1(x_1)$ to the following form:

$$\dot{e}_1 = g_1(x_1)[g_1^{-1}(x_1)f_1(x_1) + x_2 - g_1^{-1}(x_1)\dot{x}_{1d}] \quad (5)$$

We choose the virtual controller as:

$$x_{2d} = x_2 = -g_1^{-1}(x_1)f_1(x_1) + g_1^{-1}(x_1)\dot{x}_{1d} - k_1 e_1 \quad (6)$$

where k_1 is a positive constant. In order to approximate the unknown nonlinearities (functions $f_1(x_1)$ and $g_1(x_1)$) we use RBF Neural Networks. A Neural Network based virtual controller is used as follows:

$$x_{2d} = -\theta_1^{\mathrm{T}}\xi_1(x_1) + \delta_1^{\mathrm{T}} n_1(x_1)\dot{x}_{1d} - k_1 e_1 \quad (7)$$

where we have substituted the unknown nonlinearities $g_1(x_1)^{-1}f_1(x_1)$ and $g_1(x_1)^{-1}$ with the RBF Neural Networks $\theta_1^{\mathrm{T}}\xi_1(x_1)$ and $\delta_1^{\mathrm{T}} n_1(x_1)$ respectively based on Lyapunov stability [16], [17].

We take the following adaptation laws (σ-modification) in order to avoid large values of the weights:

$$\begin{aligned}\dot{\theta}_1 &= \Gamma_{11}[e_1\xi_1(x_1) - \sigma_1\theta_1] \\ \dot{\delta}_1 &= \Gamma_{12}[-e_1 n_1(x_1)\dot{x}_{1d} - \gamma_1\delta_1]\end{aligned} \quad (8)$$

with σ_1, γ_1 small and positive constants and $\Gamma_{11}=\Gamma_{11}^T>0$, $\Gamma_{12}=\Gamma_{12}^T>0$ are the adaptive gain matrices.

Step 2: In this step we make the error between x_2 and x_{2d} as small as possible.

The previous is described by the following equation:

$$e_2 = x_2 - x_{2d} \quad (9)$$

We take the derivative of e_2. After that we have:

$$\begin{aligned}\dot{e}_2 &= \dot{x}_2 - \dot{x}_{2d} = f_2(\bar{x}_2) + g_2(\bar{x}_2)x_3 - \dot{x}_{2d} \\ &= g_2(\bar{x}_2)[g_2(\bar{x}_2)^{-1}f_2(\bar{x}_2) + x_3 - g_2(\bar{x}_2)^{-1}\dot{x}_{2d}]\end{aligned} \quad (10)$$

By taking the x_{3d} as a virtual control input and by substituting the unknown nonlinearities $g_2(\bar{x}_2)^{-1}f_2(\bar{x}_2)$ and $g_2(\bar{x}_2)^{-1}$ with the RBF Neural Networks $\theta_2^{\mathrm{T}}\xi_2(\bar{x}_2)$ and $\delta_2^{\mathrm{T}} n_2(\bar{x}_2)$ respectively based on Lyapunov stability [16], [17], we have:

$$x_{3d} = -e_1 - \theta_2^{\mathrm{T}}\xi_2(\bar{x}_2) + \delta_2^{\mathrm{T}} n_2(\bar{x}_2)\dot{x}_{2d} - k_2 e_2 \quad (11)$$

We take the following adaptation laws (σ-modification) in order to avoid large values of the weights:

$$\begin{aligned}\dot{\theta}_2 &= \Gamma_{21}[e_2\xi_2(\bar{x}_2) - \sigma_2\theta_2] \\ \dot{\delta}_2 &= \Gamma_{22}[-e_2 n_2(\bar{x}_2)\dot{x}_{2d} - \gamma_2\delta_2]\end{aligned} \quad (12)$$

with σ_2, γ_2 small and positive constants and $\Gamma_{21}=\Gamma_{21}^T>0$, $\Gamma_{22}=\Gamma_{22}^T>0$ are the adaptive gain matrices.

Step 3(Final): In this step we make the error between x_3 and x_{3d} as small as possible.

The previous is described by the following equation:

$$e_3 = x_3 - x_{3d} \quad (13)$$

We take the derivative of e_3. After that we have:

$$\begin{aligned}\dot{e}_3 &= \dot{x}_3 - \dot{x}_{3d} = f_3(\bar{x}_3) + g_3(\bar{x}_3)u - \dot{x}_{3d} \\ &= g_3(\bar{x}_3)[g_3(\bar{x}_3)^{-1}f_3(\bar{x}_3) + u - g_3(\bar{x}_3)^{-1}\dot{x}_{3d}]\end{aligned} \quad (14)$$

Where u is the control input and by substituting the unknown nonlinearities $g_3(\bar{x}_3)^{-1}f_3(\bar{x}_3)$ and $g_3(\bar{x}_3)^{-1}$ with the RBF Neural Networks $\theta_3^{\mathrm{T}}\xi_3(\bar{x}_3)$ and $\delta_3^{\mathrm{T}} n_3(\bar{x}_3)$ respectively, we have:

$$u = -e_2 - \theta_3^{\mathrm{T}}\xi_3(\bar{x}_3) + \delta_3^{\mathrm{T}} n_3(\bar{x}_3)\dot{x}_{3d} - k_3 e_3 \quad (15)$$

We take the following adaptation laws (σ-modification) in order to avoid large values of the weights:

$$\begin{aligned}\dot{\theta}_3 &= \Gamma_{31}[e_3\xi_3(\bar{x}_3) - \sigma_3\theta_3] \\ \dot{\delta}_3 &= \Gamma_{32}[-e_3 n_3(\bar{x}_3)\dot{x}_{3d} - \gamma_3\delta_3]\end{aligned} \quad (16)$$

with σ_3, γ_3 small and positive constants and $\Gamma_{31}=\Gamma_{31}^T>0$, $\Gamma_{32}=\Gamma_{32}^T>0$ are the adaptive gain matrices.

D. Simulation

In order to show the effectiveness and apply the above approach a simulation is presented for the (i) form of the MAPK cascade model (as described before):

$$\dot{x}_3 = -\beta_1 x_3 + (c_1 - x_3)u(t)$$
$$\dot{x}_2 = -\beta_2 x_2 + a_{21}(c_2 - x_2)x_3$$
$$\dot{x}_1 = -\beta_3 x_1 + a_{32}(c_3 - x_1)x_2$$
$$y = x_1$$

where x_1, x_2, x_3 and y are states (concentrations) and output of the system respectively. The initial conditions (concentrations) are $x_0=[x_{10}, x_{20}, x_{30}]^T=[0.7, 0.8, 1.0]^T$ and the desired output signal of the system is $y_d=sin(t)$. In order to select the desired output signal we based on two principles. First we test the previous algorithm with $y_d=sin(t)$ (shown in the following figures) because of its slow changes and then with $y_d=(atan(10*(t-10))/\pi)+0.5$ (hyperbolic tangent) to check the fast changes. These selections are not based on any experiments in the lab.

We make the assumption that $c_1>>x_1$, $c_2>>x_2$, $c_3>>x_3$ in order $g(.)$ functions to be strictly positive and $a_{21}=a_{32}=\beta_1=\beta_2=\beta_3=1$, $c_1=9.99$, $c_2=6.66$, $c_3=3.33$.

All the basis function of the NNs have the form $G(\bar{x}_i) = \exp[-\frac{(\bar{x}_i - u_i)^T(\bar{x}_i - u_i)}{v_i^2}]$ (as described in [8]) where $u_i=[u_{i1}, u_{i2}, ..., u_{ij}]^T$ are the centers of the receptive field and v_i are the widths of the Gaussian function.

The Neural Networks $\theta_1^T\xi_1(x_1)$ and $\delta_1^T\eta_1(x_1)$ have 5 nodes with centres u_j evenly spaced in [-6, 6] and widths $v_j=1$, $\theta_2^T\xi_2(\bar{x}_2)$ and $\delta_2^T\eta_2(\bar{x}_2)$ have 25 nodes with centres u_j evenly spaced in [-6, 6] x [-6, 6] and widths $v_j=1$ and $\theta_3^T\xi_3(\bar{x}_3)$, $\delta_3^T\eta_3(\bar{x}3)$ have 125 nodes with centers u_j evenly spaced in [-6, 6] x [-6, 6] x [-6, 6] and widths $v_j=1$. We select the design parameters of the above controller as $k_1=k_2=3.5$, $\Gamma_1=\Gamma_2=diag\{2\}$, $\sigma_1=\sigma_2=\gamma_1=\gamma_2=0.2$. The initial weights θ_1, θ_2, θ_3 are arbitrarily taken in [-1.2, 1.2] and δ_1, δ_2, δ_3 in [0, 1.2].

Figs. 1-6 show the simulation results of applying the controller for tracking the desired signal y_d. From figure 1 we can see that good tracking performance is obtained. Figure 2 shows the trajectory of the controller. Figure 3 shows the phase plane of the system. Figure 4 shows the error e_1, Figure 5 shows the error e_2 and finally Figure 6 shows the error e_3.

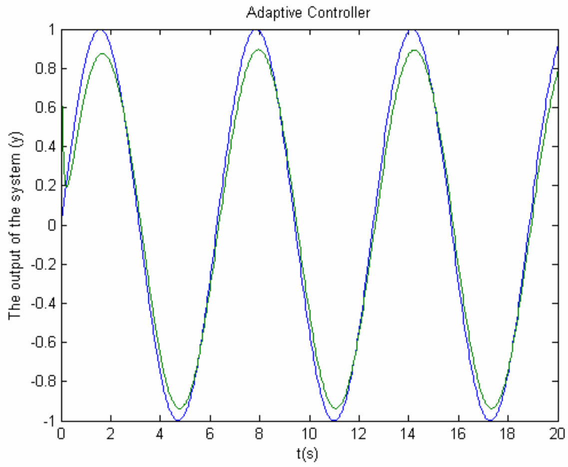


fig. 1.The output of the system under adaptive controller.

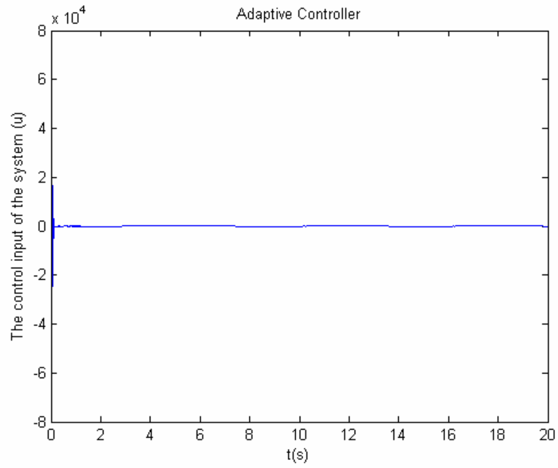


Fig2: The trajectory of the adaptive controller

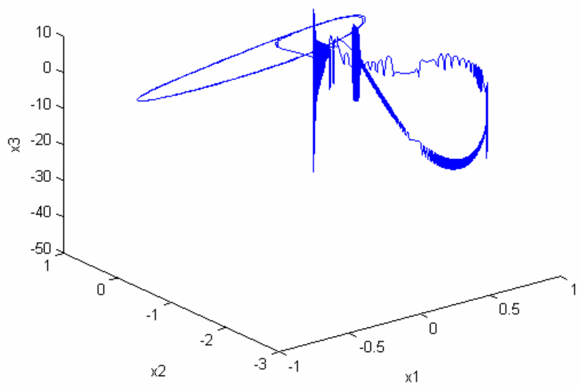


Fig3. The phase plane plot of the system

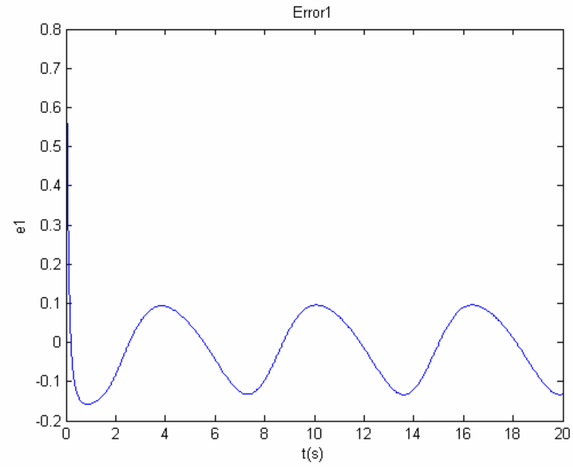


Fig4. Error e_1

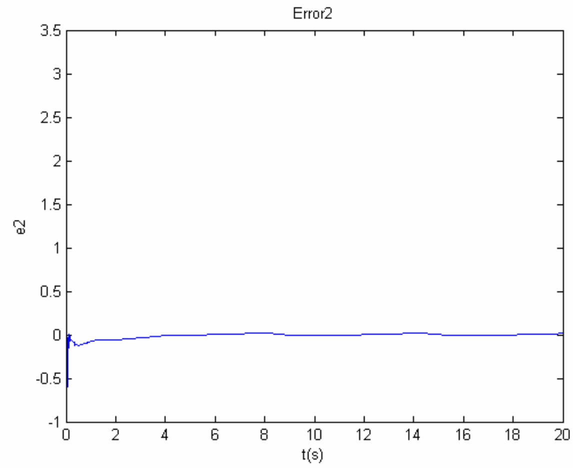


Fig5. Error e_2

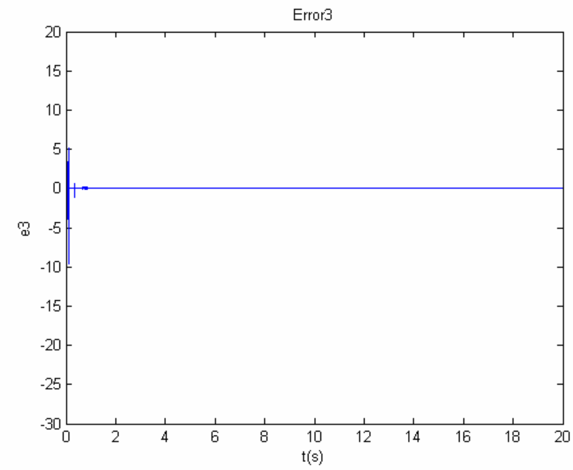


Fig6. Error e_3

III. Conclusion

In this paper, we apply the controller scheme [7] to control the output of the MAPK cascade models to reach a specific behavior without knowing the internal interconnections of them. The tracking error is bounded and is established on the basis of the Lyapunov approach. It is an adaptive control application that can be applied extensively into medicine and drug discovery.

References

[1] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, Nonlinear and Adaptive Control Design, New York: Wiley, 1995.

[2] F. L. Lewis, S. Jagannathan, and A. Yesildirek, Neural Network Control of Robot Manipulators and Nonlinear Systems, Taylor & Francis, 1999.

[3] G.A. Rovithakis, and M.A. Christodoulou, Adaptive Control with Recurrent high-order Neural Networks, Springer Verlag, New York, USA, 2000.

[4] B. Binder, R. Heinrich, "Interrelations between dynamical properties and structural characteristics of signal transduction networks," *Genome Inform.*, vol. 1, no. 1, pp. 13-23, 2004.

[5] B. Binder, R. Heinrich, "Structural and Dynamical Analyses of the Kinase Network Derived from the Transpath Database," *Genome Inform.*, vol. 16, no. 1, pp. 164-173, 2005.

[6] B. Binder, R. Heinrich, Dynamic Stability of Signal Transduction Networks Depending on Downstream and Upstream Specificity of Protein Kinases, Kluwer Academic Publishers. Printed in Netherlands., 2002.

[7] Yahui Li, Sheng Qiang, Xianyi Zhuang, and Okyay Kaynak, "Robust and Adaptive Backstepping Control for Nonlinear Systems Using RBF Neural Networks," *IEEE Trans. Neural Netowrks*, vol 15, pp 693-701, 2004.

[8] S. S.Shuzhi, S.S. Ge, and C.Wang, "Direct adaptive NN control of a class of nonlinear systems," *IEEE Trans. Neural Networks,* vol . 13, no. 1, pp. 214-221, 2002.

[9] R. M. Sanner and J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, pp. 837-863, Nov. 1992.

[10] G.A. Rovithakis, and M.A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 24, No. 3, Mar. 1994, pp. 400-412.

[11] Luis J. Ricalde, and Edgar N. Sanchez, "Inverse Optimal Design for Trajectory Tracking with Input Saturations via Adaptive Recurrent Neural Control," in Proc. of the 42nd IEEE Conference on Decision and Control, December 2003.

[12] S. Haykin, Neural Networks: A comprehensive Foundation, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[13] P. A. Ioannou and J. Sun, Robust Adaptive Control, Englewood Cliggs, NJ: Prentice-Hall, 1993.

[14] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks,* vol . 1, no. 1, pp. 4-27, 1990.

[15] B. Igelnik and Y-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net", *IEEE Trans. Neural Networks,* vol. 6, no. 6, pp. 1320-1329, November 1995.

[16] M. M. Polycarpou and P. A. Ioannou, "Identification and Control Using Neural Network Models: Design and Stability Analysis," Dept. Elect. Eng. Syst., Univ. Southern California, Los Angeles, Tech. Rep. 91-09-01, 1991.

[17] G. A. Rovithakis, "Tracking Control of multi-inputs affine nonlinear dynamical systems with unknown nonlinearities using dynamical neural networks, "*IEEE Trans. Syst., Man, Cybern.*, vol. 29, pp. 447-451, 1999.

[18] T. Zhang, S. S. Ge, and C. C. Hang, "Adaptive neural control for strict feedback nonlinear systems using backstepping," *Automatica,* vol. 36, pp. 1835-1846, 2000.