Chaotic Modeling and Simulation (CMSIM) 1: 109-122, 2012

Adaptive Backstepping Neural Network Control for Mechanical Pumps

Kyriakos G. Vamvoudakis[1], Manolis A. Christodoulou[2]

[1] Automation and Robotics Research Institute, The University of Texas at Arlington, 7300 Jack Newell Blvd. S., Ft. Worth, TX 76118
[2] Technical University of Crete, Chania, Crete, Greece
(E-mail: kyriakos@arri.uta.edu , manolis@isc.tuc.gr)

Abstract: In this paper, an Adaptive Backstepping Neural Network control approach is used for a class of affine nonlinear systems which describe the pump model in the strict feedback form. The close loop signals are semi globally 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 pump output.

1 Introduction

Recent technological developments have forced control engineers to deal with extremely complex systems that include uncertain and possibly unknown nonlinearities, operating in highly uncertain environments. Man has two principal objectives in the scientific study of his environment: he wants to understand and to control. The two goals reinforce each other, since deeper understanding permits firmer control, and, on the other hand, systematic application of scientific theories inevitably generates new problems which require further investigation, and so on.

Adaptive control [1], [10] is a powerful tool that deals with modeling uncertainties in nonlinear (and linear) systems by on line tuning of parameters. Very important research activities include on-line identification [11], [13] and pattern recognition inside the feedback control loop. Nonlinear control includes two basic forms of systems, the feedforward systems and the feedback systems.

The strict feedback systems can be controlled using the well known backstepping [1], [4], [15] technique. The purpose of backstepping is the recursive design of a controller for the system by selecting appropriate virtual controllers. Separate virtual controllers are used in order to stabilize every equation of the system. In every step we select appropriate update laws. The

Received: 29 June 2011 / Accepted 30 December 2011

ISSN 2241-0503

strict feedforward systems can be controlled using the forwarding technique that is something like backstepping but in reverse order. Other cases of systems that can be converted to the previous forms are part of a larger class of systems that are called interlaced systems as described by [17], and [18]. In these systems we combine backstepping and forwarding techniques together in order to recursively design feedback control laws. Interlaced systems are not in feedback form, nor in feedforward form. These systems have a specific methodology that differs from backstepping and forwarding. We don't start from the top equation, neither from the bottom.

Other special cases of systems are part of other forms that we call mixed interlaced and we introduce their study in the present paper. The methodology is based on classical interlaced systems and is developed by the authors. We want to make the systems solvable by one of the well known backstepping and forwarding methods. This can be reached after some specific steps that convert the system into a known form. We start from the middle equation and we continue with the top. The previous method is based on classical interlaced forms that are introduced by [17] and [18] and can be extended to more complicated systems.

A lot of researchers developed a series of results that generalized and explained the basic idea of nonlinear control. Teel [19] in his dissertation introduced the idea of nested saturations with careful selection of their parameters to achieve robustness for nonlinear controllers. After Teel [19], [17] proposed a new solution to the problem of forwarding that is based on a different Lyapunov solution.

In this paper we control a pump which is a fifth order nonlinear model, but for simplification purposes we use a third order reduced model that exists in the literature. The pump has inherent structural uncertainties with high degrees of uncertainty, thus we are forced to use our non-linear adaptive control techniques.

2 Problem Analysis

A. System Pump Description

Consider a Pump model found in the literature [20] which is presented by the following well known scheme. (The various variables are explained later in the paper. Here we give the basic figure)

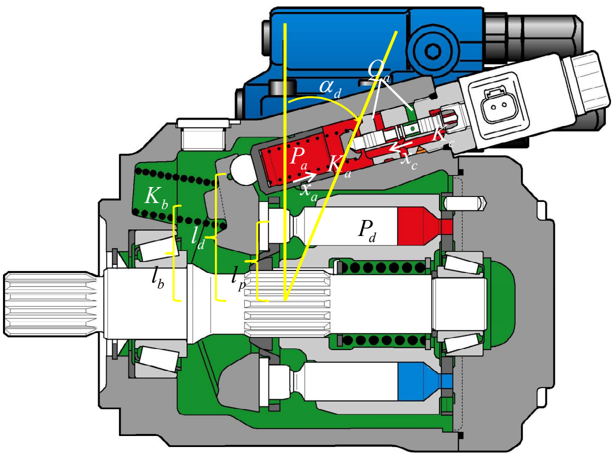

Fig 1: Schematic representation of a general Pump mechanical design

Assume we have second order dynamics

$$\begin{aligned} \dot{x}_c &= v_c \\ \dot{v}_c &= \frac{1}{m_c}\left(-x_c\left(K_c + K_a\right) - K_a l_d \tan\left(\alpha_d\right) - f_c v_c + \phi_s i_s\right) \end{aligned} \qquad (1)$$

Where x_c is the piston linear displacement, a quantity that can always be measured.

Consider certain volume within the hydraulic actuator $V_a\left(\alpha_d\right) = V_{a0} - S_a l_d \tan\left(\alpha_d\right)$; the dynamics of pressure within the control actuator are:

$$\Delta\dot{P}_a\left(t\right) = \frac{\beta_{oil}}{V_a\left(\alpha_d\right)}\left[-Q_a\left(x_c\right) + \frac{l_d S_a}{\cos^2\left(\alpha_d\right)}\omega_d\right], \qquad (2)$$

where the actuator ingoing/outgoing flow is

$$Q_a\left(x_c\right) = \begin{cases} -\mu_{a,in} f\left(x_c\right)\operatorname{sgn}\left(P_d - P_a\right)\sqrt{\left|\Delta P_d - \Delta P_a\right|} & x_c < 0 \\ \mu_{a,out} f\left(x_c\right)\operatorname{sgn}\left(P_a - P_i\right)\sqrt{\left|\Delta P_a - P_i\right|} & x_c > 0 \end{cases} \qquad (3)$$

Remark 1: The outgoing actuator flow has been assumed to be positive, i.e., $x_c > 0 \Leftrightarrow Q_a > 0$.

Remark 2: The valve stroke is modelled as

$$f(x_c) = a_1 |x_c| + a_2 x_c^2 \,. \tag{4}$$

The dynamic behaviour of the disc is governed by the following torque equations:

$$\dot{\alpha}_d = \omega_d$$
$$\dot{\omega}_d = \frac{1}{J(\alpha_d)} \Big[-K_T(\omega, \alpha_d)\tan(\alpha_d) - f_c(\alpha_d)\omega_d^2 - \tag{5}$$
$$f_d(\alpha_d, \Delta P_d)\omega_d - l_d K_a x_c(\alpha_d) - l_d S_a \Delta P_a - f_p(\alpha_d)\Delta P_d \Big]$$

with

$$f_p(\alpha_d) = \frac{N A_p l_p}{2\pi \cos^2(\alpha_d)} \gamma_0 ;$$
$$f_d(\alpha_d, P_d) = \frac{N C_p l_p^2}{2\cos^4(\alpha_d)} + K_d(P_d)$$
$$f_c(\alpha_d) = \frac{N M_p l_p^2}{\cos^4(\alpha_d)} \tan(\alpha_d); \tag{6}$$
$$K_T(\omega, \alpha_d) = l_d^2 K_a + K_b l_b^2 - \frac{N M_p l_p^2 \omega^2}{2\cos^2(\alpha_d)}$$
$$J(\alpha_d) = J_d + \frac{N M_p l_p^2}{2\cos^4(\alpha_d)}$$

The output flow of the pump is given by:

$$\begin{aligned} Q_p &= K_p(\omega) l_d \tan(\alpha_d), \quad x_c > 0 \\ Q_p &= K_p(\omega) l_d \tan(\alpha_d) - Q_a(x_c), \quad x_c < 0 \end{aligned} \tag{7}$$

The mechanical link between the disc and the actuator is provided by the following equation (assuming that the angle is small, we linearize the tangent)

$$x_a = l_d \tan(\alpha_d) \approx l_d \alpha_d \,, \tag{8}$$

Under all the above assumptions the fifth order nonlinear model for the pump is given by:

$$\begin{aligned}
\dot{x}_c &= v_c \\
\dot{v}_c &= \frac{1}{m_c}\left(-x_c\left(K_c+K_a\right)-K_a x_a - f_c v_c + \phi_s i_s\right) \\
\dot{x}_a &= v_a \\
\dot{v}_a &= \frac{1}{J}\left[-K_T x_a - f_c x_a v_a^2 - f_d v_a - l_d^2 K_a x_c - l_d^2 S_a \Delta P_a - f_p \Delta P_d\right] \\
\Delta\dot{P}_a(t) &= \frac{\beta_{oil}}{V_{a0}-S_a x_a}\left[-Q_a\left(x_c\right)+S_a v_a\right];
\end{aligned} \tag{9}$$

with

$$\begin{aligned}
f_p &= \frac{NA_p l_p l_d \gamma_0}{2\pi}; \qquad f_d\left(\Delta P_d\right) = \frac{NC_p l_p^2}{2} + K_d\left(\Delta P_d\right) \\
f_c &= \frac{NM_p l_p^2}{l_d^2}; \qquad K_T\left(\omega\right) = l_d^2 K_a + K_b l_b^2 - \frac{NM_p l_p^2 \omega^2}{2}; \\
J &= J_d + \frac{NM_p l_p^2}{2}
\end{aligned} \tag{10}$$

Assuming negligible valve dynamics we may get a reduced third order model as follows:

$$\begin{aligned}
\dot{x}_a &= v_a \\
\dot{v}_a &= \frac{1}{J}\left[-K_T x_a - f_c x_a v_a^2 - f_d v_a - l_d^2 K_a x_c - l_d^2 S_a \Delta P_a - f_p \Delta P_d\right] \\
\Delta\dot{P}_a(t) &= \frac{\beta_{oil}}{V_{a0}-S_a x_a}\left[-Q_a\left(x_c\right)+S_a v_a\right];
\end{aligned} \tag{11}$$

and

$$x_c = \frac{\phi_s i_s - K_a x_a}{K_c + K_a} = b_s i_s - b_a x_a \tag{12}$$

The pump is commanded assuming action on the actuator flow, and then we get:

$$\begin{aligned}
\dot{x}_a &= v_a \\
\dot{v}_a &= \frac{1}{J}\left[-K_T x_a - f_c x_a v_a^2 - f_d v_a - l_d^2 S_a \Delta P_a - f_p \Delta P_d\right] \\
\Delta\dot{P}_a(t) &= \frac{\beta_{oil}}{V_{a0}-S_a x_a}\left[-Q_a\left(x_c\right)+S_a v_a\right];
\end{aligned} \tag{13}$$

and $x_c = b_s i_s$.

Equation (11) can be expressed in (or transformed to) the following nonlinear state space form:

$$\begin{aligned} \dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, 1 \le i \le n-1 \\ \dot{x}_i &= f_i(\bar{x}_i) + g_n(\bar{x}_n)u \ , n \ge 2 \\ y &= x_1 \end{aligned} \tag{14}$$

where $\bar{x}_i = [x_1, x_2, ... x_i]^T \in R^i, i = 1,...n, u \in R, y \in R$ are state variables, input and output respectively. More accurately for the pump model (11) we have $f_1(x_1) = 0$, $g_1(x_1) = 1$, $f_2(\bar{x}_2) = \frac{1}{J}\left(-K_T x_a - f_c x_a v_a^2 - f_d v_a\right)$, $g_2(\bar{x}_2) = \frac{1}{J}\left(-l_d^2 S_a \Delta P_a - f_p \Delta P_d\right)$, $f_3(\bar{x}_3) = \frac{\beta_{oil}}{V_{a0} - S_a x_a} S_a v_a$, $g_3(\bar{x}_3) = -Q_a\left(b_s\right)\frac{\beta_{oil}}{V_{a0} - S_a x_a}$.

Our purpose is to construct a specific adaptive Neural Network controller (the proof is omitted due to space) such that:

i) all the signals in the close loop remain semi globally 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 [2], [3], [5], [9], [16]. 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 $g_{i1}(.) \ge g_{i0}(.) > 0$ such that, $g_{i1}(.) \ge \left|g_{i1}(.) \ge\right| g_{i0}(.)$, $\forall \bar{x}_n \in \Omega \subset R^n$.

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

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 [6] in order to approximate the nonlinear functions of our systems [14], [15]. The idea behind this is described fully at [2], [3], [7], [8], [9], [15]. 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

F: $\Omega \to R$, where Ω is a compact subset of R^m (m is an appropriate integer) and $\varepsilon > 0$, there exists an RBF vector ξ: $R^m \to R^p$ and a weight vector $\theta^* \in R^p$ such that $\left|F(x)-\theta^{*T}\xi(x)\right| \le \varepsilon \; \forall x \in \Omega$. 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.

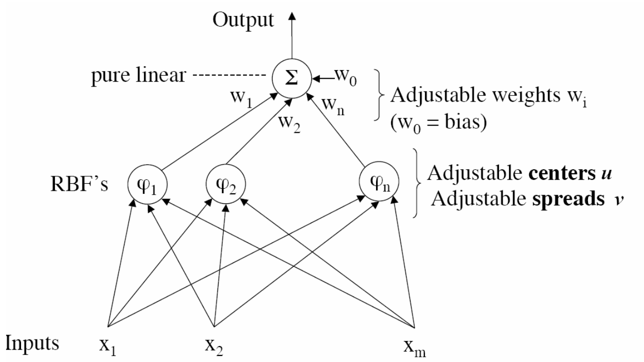

Fig 2: Schematic representation of RMF Neural Networks

C. Controller Design

In [15], a desired feedback control law was initially proposed for system (13) 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. [4]. The design procedure is described in 3 steps because in the pump model 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:

$$e_1 = x_1 - x_{1d} \tag{15}$$

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

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_{1d} \Rightarrow \dot{e}_1 = f_1(x_1) + g_1(x_1)x_2 - \dot{x}_{1d} \tag{16}$$

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:

$$\dot{e}_1 = g_1(x_1)[g_1^{-1}(x_1)f_1(x_1) + x_2 - g_1^{-1}(x_1)\dot{x}_{1d}] \tag{17}$$

We choose the virtual controller as:

$$x_{2d} = x_2 = -g_1^{-1}(x_1)f_1(x_1) + g_1^{-1}(x_1)\dot{x}_{1d} - k_1 e_1 \qquad (18)$$

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:

$$x_{2d} = -\theta_1^{\mathrm{T}}\xi_1(x_1) + \delta_1^{\mathrm{T}} n_1(x_1)\dot{x}_{1d} - k_1 e_1 \qquad (19)$$

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 [2].

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

$$\begin{aligned} \dot{\theta}_1 &= \Gamma_{11}[e_1\xi_1(x_1) - \sigma_1\theta_1] \\ \dot{\delta}_1 &= \Gamma_{12}[-e_1 n_1(x_1)\dot{x}_{1d} - \gamma_1\delta_1] \end{aligned} \qquad (20)$$

with σ_1, γ_1 small and positive constants and $\Gamma_{11} = \Gamma_{11}^{\mathrm{T}} > 0$, $\Gamma_{12} = \Gamma_{12}^{\mathrm{T}} > 0$ 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:

$$e_2 = x_2 - x_{2d} \qquad (21)$$

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

$$\begin{aligned} \dot{e}_2 &= \dot{x}_2 - \dot{x}_{2d} = f_2(\overline{x}_2) + g_2(\overline{x}_2)x_3 - \dot{x}_{2d} \\ &= g_2(\overline{x}_2)[g_2(\overline{x}_2)^{-1} f_2(\overline{x}_2) + x_3 - g_2(\overline{x}_2)^{-1}\dot{x}_{2d}] \end{aligned} \qquad (22)$$

By taking the x_{3d} as a virtual control input and by substituting the unknown nonlinearities $g_2(\overline{x}_2)^{-1} f_2(\overline{x}_2)$ and $g_2(\overline{x}_2)^{-1}$ with the RBF Neural Networks $\theta_2^{\mathrm{T}}\xi_2(\overline{x}_2)$ and $\delta_2^{\mathrm{T}} n_2(\overline{x}_2)$ respectively based on Lyapunov stability [2], we have:

$$x_{3d} = -e_1 - \theta_2^{\mathrm{T}}\xi_2(\overline{x}_2) + \delta_2^{\mathrm{T}} n_2(\overline{x}_2)\dot{x}_{2d} - k_2 e_2 \qquad (23)$$

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

$$\dot{\theta}_2 = \Gamma_{21}[e_2\xi_2(\bar{x}_2) - \sigma_2\theta_2]$$
$$\dot{\delta}_2 = \Gamma_{22}[-e_2 n_2(\bar{x}_2)\dot{x}_{2d} - \gamma_2\delta_2] \quad (24)$$

with σ_2, γ_2 small and positive constants and $\Gamma_{21} = \Gamma_{21}^{\mathrm{T}} > 0, \Gamma_{22} = \Gamma_{22}^{\mathrm{T}} > 0$ 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:

$$e_3 = x_3 - x_{3d} \quad (25)$$

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

$$\begin{aligned}\dot{e}_3 &= \dot{x}_3 - \dot{x}_{3d} = f_3(\bar{x}_3) + g_3(\bar{x}_3)u - \dot{x}_{3d} \\ &= g_3(\bar{x}_3)[g_3(\bar{x}_3)^{-1} f_3(\bar{x}_3) + u - g_3(\bar{x}_3)^{-1}\dot{x}_{3d}]\end{aligned} \quad (26)$$

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:

$$u = -e_2 - \theta_3^{\mathrm{T}}\xi_3(\bar{x}_3) + \delta_3^{\mathrm{T}} n_3(\bar{x}_3)\dot{x}_{3d} - k_3 e_3 \quad (27)$$

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

$$\dot{\theta}_3 = \Gamma_{31}[e_3\xi_3(\bar{x}_3) - \sigma_3\theta_3]$$
$$\dot{\delta}_3 = \Gamma_{32}[-e_3 n_3(\bar{x}_3)\dot{x}_{3d} - \gamma_3\delta_3] \quad (28)$$

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

3 Simulation

In order to show the effectiveness and apply the above approach a simulation is presented for the pump model:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\phi_a x_1 - f_a\left(x_1, x_2\right)x_2 - a_a x_3$$
$$\dot{x}_3 = \chi_a\left(x_1\right)\left[x_2 - q_a\left(x_3\right)u\right]$$
$$y = x_1$$

where x_1, x_2, x_3 and y are states and output of the system respectively. The initial conditions are $x_0 = [x_{10},\ x_{20},\ x_{30}]^T = [0.3,\ 0.2,\ 0.1]^T$ and the desired output signal of the system is $y_d = (atan\left(10*\left(t-10\right)\right)/\pi) + 0.5$. These selections are not based on any experiments in the lab.

We make the assumption that all the basis function of the NNs [12] have the form $G(\overline{x}_i) = \exp[-\frac{(\overline{x}_i - u_i)^T(\overline{x}_i - u_i)}{v_i^2}]$ (as described in [6]) where $u_i = \left[u_{i1},\ u_{i2},\ \ldots,\ u_{ij}\right]^T$ are the centers of the receptive field and v_i are the widths of the Gaussian function.

The Neural Networks $\theta_1^{\mathrm{T}}\xi_1(x_1)$ and $\delta_1^{\mathrm{T}}\eta_1(x_1)$ have 5 nodes with centres u_j evenly spaced in [-6, 6] and widths $v_j = 1$, $\theta_2^{\mathrm{T}}\xi_2(\overline{x}_2)$ and $\delta_2^{\mathrm{T}}\eta_2(\overline{x}_2)$ have 25 nodes with centres u_j evenly spaced in [-6, 6] x [-6, 6] and widths $v_j=1$ and $\theta_3^{\mathrm{T}}\xi_3(\overline{x}_3)$, $\delta_3^{\mathrm{T}}\eta_3(\overline{x}_3)$ 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 $k_1 = k_2 = 3.5,\ \Gamma_1 = \Gamma_2 = diag\left\{2\right\},\ \sigma_1 = \sigma_2 = \gamma_1 = \gamma_2 = 0.2$. 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. 3-8 show the simulation results of applying the controller for tracking the desired signal y_d. From figure 3 we can see that good tracking performance is obtained. Figure 4 shows the trajectory of the controller. Figure 5 shows the phase plane of the system. Figure 6 shows the error e_1, Figure 7 shows the error e_2 and finally Figure 8 shows the error e_3.

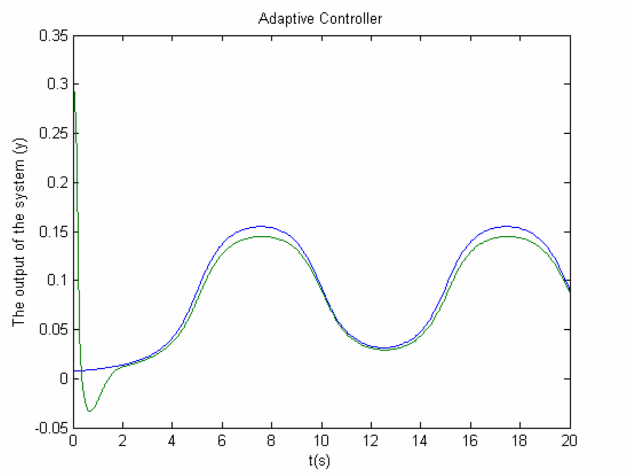

Fig. 3: The output of the system under adaptive controller.

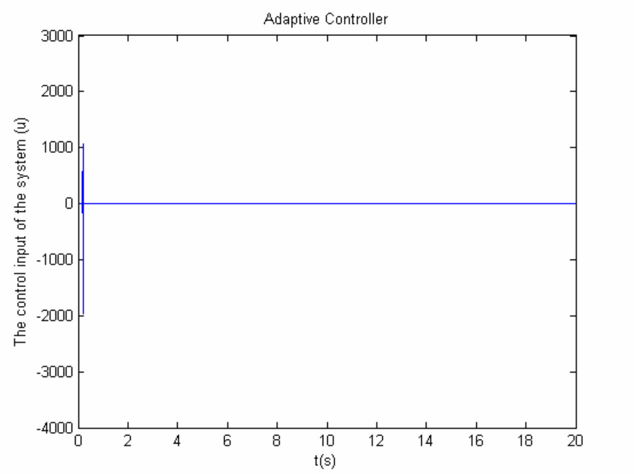

Fig. 4: The trajectory of the adaptive controller.

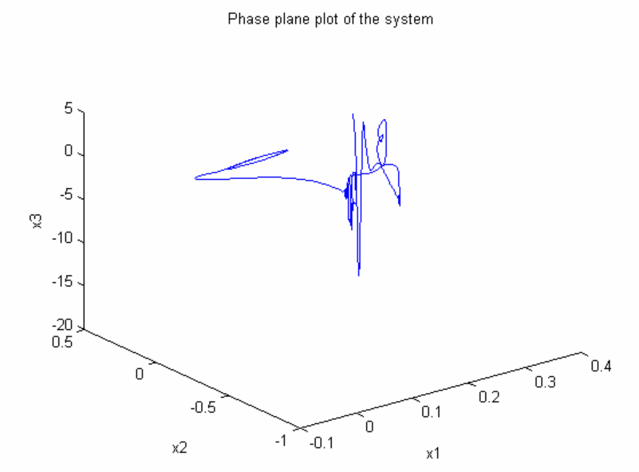

Fig. 5: The phase plane plot of the system.

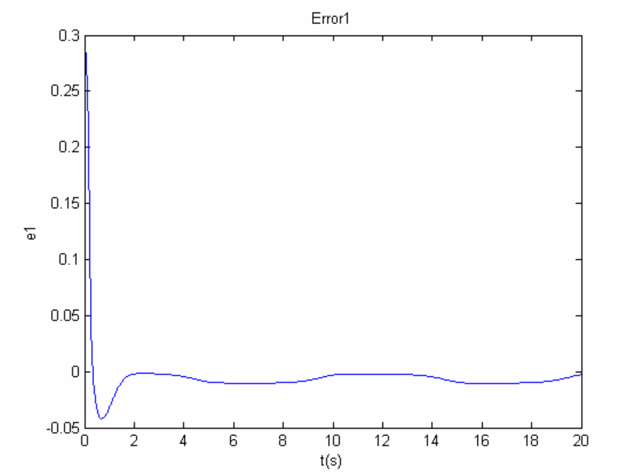

Fig. 6: Error e_1.

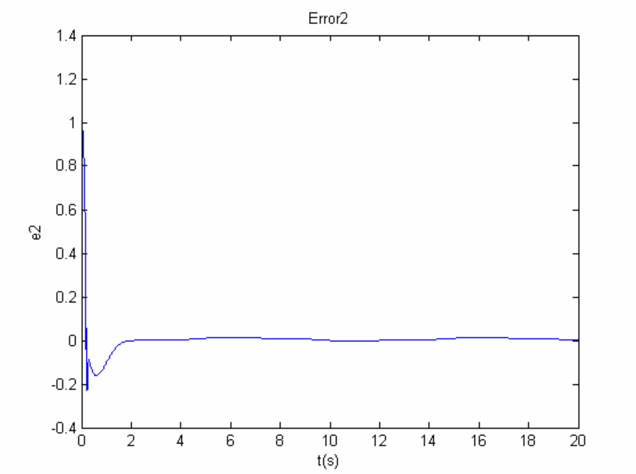

Fig. 7: Error e_2.

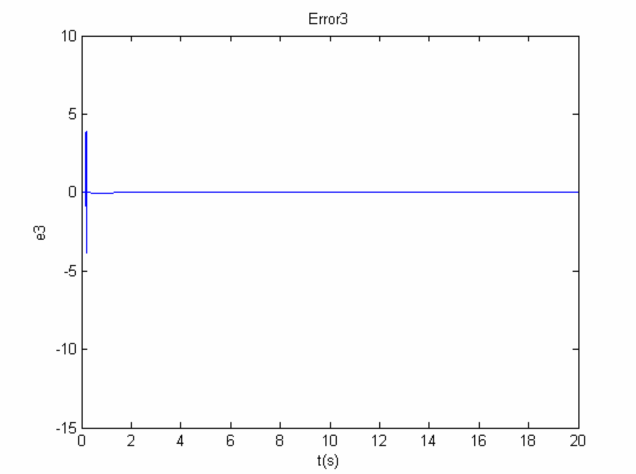

Fig. 8: Error e_3.

4 Conclusion

In this paper, we apply a backstepping controller scheme to control the output of the pump model to reach a specific pressure behavior without knowing the dynamics. The tracking error is bounded and is established on the basis of the Lyapunov approach. Simulation results show the effectiveness of this algorithm in controlling the mechanical pump. Future research will be focused on implementing this algorithm in the real experimental model.

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] 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.

[5] 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.

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

[7] 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.

[8] 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.

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Kyriakos G. Vamvoudakis, Manolis A. Christodoulou, "Adaptive Backstepping Control for MAPK Cascade Models Using RBF Neural Networks," Proc. Mediterranean Conf. Control & Automation, Athens, Greece, June 2007.

[17] P. V. Kokotovic, R. Sepulchre and M. Jankovic, Constructive Nonlinear Control. New York: Springer Verlag, 1997.

[18] M. Krstic, "Feedback Linearizability and Explicit Integrator Forwarding Controllers for Classes of Feedforward Systems," *IEEE Trans. on Automatic Control*, vol. 49, pp. 1668-1682, 2004.

[19] A. R. Teel, "Feedback stabilization: Nonlinear solutions to inherently nonlinear problems," Ph.D., Univ. California, Berkeley, CA, 1992.

[20] E. Canuto, S. Malan, Wilber Acuna, "Pump modelling", Technical Report, Politecnico di Torino, 2009.