The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

Quasi-ARX Neural Network and Its Application to Adaptive Control of Nonlinear Systems

Lan Wang, Yu Cheng and Jinglu Hu

Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japans
(Tel : 81-93-692-5271; Fax : 81-093-692-5271)
(E-mail: wanglan@fuji.waseda.jp, chengyu0930@fuji.waseda.jp, jinglu@waseda.jp)

Abstract: This paper introduces a quasi-ARX neural network and discusses its application to adaptive control of nonlinear systems. A switching mechanism is employed to improve the performance of control system. An adaptive switching control of nonlinear system is established and some stability analysis of control system is shown. Simulations are given to show the effectiveness of the proposed method both on stability and accuracy.

Keywords: Nonlinear system, Quasi-ARX neural network, Adaptive control, Switching mechanism.

I. INTRODUCTION

Neural networks have been used to identify and control nonlinear dynamical systems because of its ability to approximate arbitrary map to any desired accuracy[1-4]. Some researchers have used neural networks directly to identify and control nonlinear systems[2, 5]. Many nonlinear ARX models based directly on neural networks also have been proposed. However, from a user's point of view, there are three major criticisms on those neural network models. One is that their parameters do not have useful interpretations. The second is that they do not have a friendly interface for controller design and system analysis[4]. The third one is that initialization and overfitting will lead to instability.

To solve these criticisms, a quasi-ARX neural network model has been proposed which embodied a macro-model part and a kernel part[3, 4]. The macro-model part is a user-friendly interface constructed using already known knowledge and the characteristic of network structure. In this paper, we will limit our discussion to a quasi-ARX approach. The linear ARX model has a various useful linearity properties which will solve the former criticisms. The kernel part is an ordinary neural network, which is used to parameterize the coefficients of macro-model. The quasi-ARX neural network is different from a nonlinear ARX model based directly on neural networks because of the linear characteristics and it also can be used to identify and control nonlinear systems accurately because of the nonlinear characteristics. In Ref.[4], we proposed an off-line control scheme based on the quasi-ARX neural network. We will discuss the adaptive control of nonlinear system in this paper.

Quasi-ARX neural network has two parts: the linear part and the nonlinear part. If the linear part would be used to ensure the nonlinear control stability and the nonlinear part would be utilized to improve the control accuracy, both stability and universal approximation capability will be realized. Motivated by the discussion, a switching law is established into the model based on the characteristic of quasi-ARX neural network structure in this paper. An adaptive switching control law is proposed for nonlinear dynamical systems and control system stability is proved.

The paper is organized as follows: In section 2, the considered system is given. In section 3, the quasi- ARX neural network model is introduced. Section 4 gives the parameters identification methods and the switching mechanism. Section 5 describes adaptive control system and analyzes the stability. Then, numerical simulations are carried out to show the effectiveness of the proposed modeling in Section 6. At last Section 7 gives some conclusions.

II. PROBLEM DESCRIPTION

Consider a single-input-single-output (SISO) black-box nonlinear system

$$y(t) = g(\varphi(t)) + v(t), \tag{1}$$

where $\varphi(t) = [y(t-1), ...y(t-n), u(t-d), ..., u(t-m-d+1)]^T$. $y(t)$ denotes the output at time t ($t = 1, 2, ...$), $u(t)$ the input, d the known integer time delay (Let d=1 in this paper, and other conditions can be got by the similar method.), $\varphi(t)$ the regression vector, and $v(t)$ the system disturbance. $g(\cdot)$ is a nonlinear function.

Assumption 1 (i) $g(\cdot)$ is a continuous function, and at $\varphi(t) = 0$ it is C^∞ continuous. (ii) the system is controllable, where a reasonable unknown controller may be expressed by $u(t) = \rho(\xi(t))$, where $\xi(t) = [y(t)\ ...\ y(t-n)\ u(t-1)\ ...\ u(t-m)\ y^*(t+1)\ ...\ y^*(t+1-l)]^T$ ($y^*(t)$ denotes reference output). (iii) the system has a globally uniformly asymptotically stable zero dynamics.

III. QUASI-ARX NEURAL NETWORK MODEL

1. Regression Form Representation

Through Taylor expansion of function $g(\cdot)$ around the region $\varphi(t) = 0$

$$y(t) = g(0)+g'(0)\varphi(t)+\frac{1}{2}\varphi^T(t)g''(0)\varphi(t)+ ... +v(t). \tag{2}$$

Let

$$\theta(\varphi(t)) \quad = \quad \left(g'(0) + \frac{1}{2}\varphi^T(t)g''(0) + \cdots\right)^T$$

$$= [a_{1,t} \dots a_{n,t}\ b_{0,t} \dots b_{m-1,t}]^T$$

where the coefficients $a_{i,t} = a_i(\varphi(t))$ and $b_{i,t} = b_i(\varphi(t))$ are nonlinear functions of $\varphi(t)$. g(0)= 0 is assumed for simplicity. A regression form of the system (1) is described by (3):

$$y(t) = \varphi^T(t)\theta(\varphi(t)) + v(t). \tag{3}$$

Now, two polynomials $A(q^{-1}, \varphi(t))$ and $B(q^{-1}, \varphi(t))$ based on the coefficients $a_{i,t}$ and $b_{i,t}$ are defined by

$$\begin{aligned} A(q^{-1}, \varphi(t)) &= 1 - a_{1,t}q^{-1} - \dots - a_{n,t}q^{-n} \\ B(q^{-1}, \varphi(t)) &= b_{0,t} + \dots + b_{m-1,t}q^{-m+1} \end{aligned}$$

where q^{-1} is the backward shift operator, e.g. $q^{-1}u(t) = u(t-1)$.

A similar-linear ARX model is developed

$$A(q^{-1}, \varphi(t))y(t) = B(q^{-1}, \varphi(t))u(t-1) + v(t). \tag{4}$$

2. Weighted One-Step-Ahead Predicate

The theorem for a d-step prediction has been proved in Ref.[4]. When d=1, one-step predictor is given :

$$\hat{y}(t+1) = \alpha(q^{-1}, \phi(t))y(t) + \beta(q^{-1}, \phi(t))u(t) \tag{5}$$

where $\phi(t) = [y(t), ...y(t-n+1), u(t), ...u(t-m+1)]^T$. $\alpha(q^{-1}, \phi(t)) = \alpha_{0,t} + \dots + \alpha_{n-1,t}q^{-n-1}$ and $\beta(q^{-1}, \phi(t)) = \beta_{0,t} + \dots + \beta_{m-1,t}q^{-m+1}$.

The prediction model (5) is a general one that is nonlinear in the variable $u(t)$, because the coefficient $\alpha_{i,t}$ and $\beta_{i,t}$ are function of $\phi(t)$. With Assumption 1 (ii), the unknown $\rho(\cdot)$ replaces variable $u(t)$ in the coefficients $a_{i,t}$ and $b_{i,t}$

$$\begin{aligned} \alpha_{i,t} &= \alpha_i(\mathbf{x}(\varphi(t))) \simeq \alpha_i(\phi_\rho(t)) \triangleq \alpha_i(\xi(t)) \\ \beta_{i,t} &= \beta_i(\mathbf{x}(\varphi(t))) \simeq \beta_i(\phi_\rho(t)) \triangleq \beta_i(\xi(t)) \end{aligned}$$

where $\phi_\rho(t)$ is $\phi(t)$ whose element $u(t)$ is replaced by $\rho(\xi(t))$, that is, $\phi_\rho(t) = [y(t) \dots y(t-n+1)\ \rho(\xi(t))\ u(t-1) \dots u(t-m+1)]^T$. $\xi(t)$ has a form of

$$\begin{aligned} \xi(t) = [&y(t) \dots y(t-n+1)\ u(t-1) \dots \\ &u(t-m+1)\ y^*(t+1)]. \end{aligned}$$

We have a predictor expressed by:

$$\hat{y}(t+1) = \alpha(q^{-1}, \xi(t))y(t) + \beta(q^{-1}, \xi(t))u(t) \tag{6}$$

We finally express the predictor by

$$\hat{y}(t+1) = \Psi^T\Theta_\xi \tag{7}$$

where $\Psi(t) = [y(t) \dots y(t-n+1)\ u(t) \dots u(t-m+1)]$.

3. Incorporation of Neural Networks

Parameterizing Θ_ξ with a MIMO neural network, the quasi-ARX prediction model is expressed by

$$\hat{y}(t+1) = \Psi(t)^T(t)\mathcal{N}(\xi(t), \Omega) \tag{8}$$

where $\mathcal{N}(\cdot, \cdot)$ is a 3-layer neural network with n input nodes, M sigmoid hidden nodes and $n+1$ *linear* output nodes[1].

[1] The number of input node is $n = \dim(\xi(t)) = n_y + n_u$, the number of output node is equal to $\dim(\Phi(t)) = n+1$

Let us express the 3-layer neural network by

$$\mathcal{N}(\xi(t), \Omega) = W_2\Gamma(W_1\xi(t) + B) + \Theta \tag{9}$$

where $\Omega = \{W_1, W_2, B, \Theta\}$, $W_1 \in \mathcal{R}^{M\times n}$, $W_2 \in \mathcal{R}^{(n+1)\times M}$ are the weight matrices of the first and second layers, $B \in \mathcal{R}^{M\times 1}$ is the bias vector of hidden nodes, $\Theta_2 \in \mathcal{R}^{(n+1)\times 1}$ is the bias vector of output nodes, and Γ is the diagonal nonlinear operator with identical sigmoid elements σ (i.e., $\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}}$). Then the quasi-ARX prediction model (9) is expressed in a form of

$$\hat{y}(t+1) = \Psi^T(t)\Theta + \Psi^T(t) \cdot W_2\Gamma(W_1\xi(t) + B). \tag{10}$$

The quasi-ARX neural networks prediction model consists of two parts: the first term of the right side of (10) is a linear ARX prediction model part, while the second term is a nonlinear part. Therefore, in the quasi-ARX prediction model the bias of output nodes Θ describes a linear approximation of the object system.

Assumption 2 (i) The linear parameters Θ lies in a compact region $\mathcal{B}$. (ii) The nonlinear term $\Psi^T(t) \cdot W_2\Gamma(W_1\xi(t) + B)$ is globally bounded, i.e. $\Psi^T(t) \cdot W_2\Gamma(W_1\xi(t) + B) \le \Delta$.

IV. PARAMETERS ESTIMATION AND SWITCHING CRITERION FUNCTION

1. Parameter Estimation

For the quasi-ARX model, the linear part parameter Θ is updated as:

$$\hat{\Theta}(t+1) = \hat{\Theta}(t) + \frac{a(t)\Psi(t)e_1(t)}{1 + \Psi(t)^T\Psi(t)} \tag{11}$$

$$a(t) = \begin{cases} 1 & \text{if } \ |e_1(t)| > 2\Delta \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where ε is a small positive constant. $e_1(t) = y(t+1) - \Psi(t)^T\hat{\Theta}(t)$.

For the quasi-ARX model, the nonlinear part error: $e_2(t) = y(t+1) - \Psi(t)^T\hat{\Theta}(t) - \Psi^T(t) \cdot W_2(t)\Gamma(W_1(t)\xi(t) + B(t))$. is chosen

The parameters Ω of the part combined with neural networks is adjusted by BP algorithm.

2. Switching Criterion Function

When the overfitting in the neural network part happens, nonlinear part will be turned off.

Now give the switching criterion function as in [7]:

$$\begin{aligned} J_i(t) = &\sum_{l=k}^{(t)} \frac{a_i(l)(\| e_i(l) \|^2 - 4\Delta^2)}{2(1 + a_i(l)\Psi(l-k)^T P_i(l-k-1)\Psi(l-k))} \\ &+ c * \sum_{l=t-N+1}^{t} (1 - a_i(l) \| e_i(l) \|^2) \end{aligned} \tag{13}$$

where N is an integer and $c \ge 0$ is a predefined constant.

By comparing $J_1(t)$ and $J_2(t)$, decides when the nonlinear part is abandoned. If $J_1(t) > J_2(t)$ the nonlinear part is chosen, else only use linear part to identify.

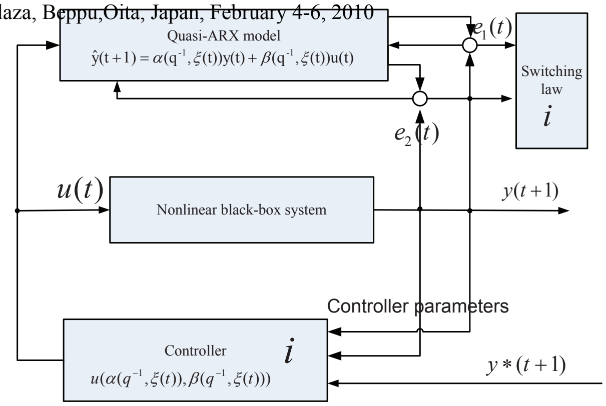

Fig. 1 A Switching control to nonlinear system based on linear model and quasi-ARX model

VI. CONTROLLER DESIGN AND STABILITY ANALYSIS

Consider a minimum variance control with the criterion function as follows:

$$M(t+1) = \left[\frac{1}{2}(y(t+1) - y^*(t+1))^2 + \frac{\lambda}{2}u(t)^2\right] \quad (14)$$

where λ is weighting factor for the control input.

The controller can obtain by solving:

$$\frac{\partial M(t+1)}{\partial u(t)} = 0 \quad (15)$$

Therefore, the proposed models can derive controller by solving (15):

$$u(t) = \frac{\beta_{0,t}}{\beta_{0,t}^2 + \lambda}((\beta_{0,t} - \beta(q^{-1},\xi(t))q)u(t-1) + y^*(t+1) - \alpha(q^{-1},\xi(t))y(t)). \quad (16)$$

where

$$[\beta(q^{-1},\xi(t)); \alpha(q^{-1},\xi(t))] = \quad (17)$$
$$\begin{cases} \Theta(t) & \text{if } J_1(t) < J_2(t) \\ W_2(t)\Gamma(W_1(t)\xi(t) + B(t)) + \Theta(t) & \text{otherwise} \end{cases}$$

The proposed controller has three distinctive features:
(1) it is linear for the variables synthesized in control systems;
(2) its parameters have explicit meanings;
(3) it is one controller which have a switching algorithm.

Figure 1 shows the controller for unknown nonlinear systems. We can see that the identified model and controller model share their parameters.

Theorem For the system (1) with adaptive controller (13), all the input and output signals in the closed-loop system are bounded. Moreover, the tracking error of the system can converge on zero when a properly neural network is determined.

Proof: (i) Subtracting Θ_0 from both sides of (11), and gives:

$$\tilde{\Theta}(t+1) = \tilde{\Theta}(t) - \frac{a(t)\Psi(t)(\Psi(t)^T\tilde{\Theta}(t) - \omega(t))}{1 + \Psi(t)^T\Psi(t)} \quad (18)$$

where $\omega(t) = y(t+1) - \Psi(t)^T\Theta_1(t)$.

Consider the following functional:

$$V(t) = \|\tilde{\Theta}(t)\|^2. \quad (19)$$

Then, noting that a(t)=0 or 1, and combined with (11) and (12), we can get as in [6]:

$$V(t+1) \le V(t) + \frac{2a(t)\Delta^2}{1 + \Psi(t)^T\Psi(t)} - \frac{1}{2}\frac{a_1(t)e_1(t)^2}{1 + \Psi(t)^T\Psi(t)}. \quad (20)$$

In view of (20), {V(t)} is a nonincreasing sequence bounded below by zero. Moreover,

$$\lim_{N\to\infty}\sum_{t=1}^{N}\frac{a(t)(e_1(t)^2 - 4\Delta)}{2(1 + \Psi(t)^T\Psi(t))} < \infty, \quad (21)$$

and

$$\lim_{N\to\infty}\frac{a(t)(e_1(t)^2 - 4\Delta)}{2(1 + \Psi(t)^T\Psi(t))} \to 0. \quad (22)$$

Along with (iii) of Assumptions 1 similar to Ref.[7], $e_1(t)$ is bounded.

By (13) and (22), the second term of $J_1(t)$ is always bounded. $J_2(t)$ has two cases:

(i) $J_2(t)$ is bounded. so the model error $e(t)$ is bounded and satisfies (22).

(ii) $J_2(t)$ is unbounded. Since (1)$J_1(t)$ is bounded. So there exists a constant t_0 such that $J_1(t) < J_2(t),\ \forall t > t_0$. The model also has bounded error $e(t)$.

From above inequalities, the input and output of the closed-loop switching control system are bounded.

The linear model is always bounded. If a proper nonlinear model is chosen and the accurate parameters is adjusted, the nonlinear model error $e_2(t)$ can converge on zero. It also exists a constant T_0 satisfies $J_2(t) < J_1(t),\ \forall t > T_0$. Then the tracking error of model can converge on zero.

VII. CONTROL SIMULATIONS

Now consider a nonlinear system:

$$\begin{aligned} y(t) &= \frac{\exp(-y^2(t-2)) * y(t-1)}{1 + u^2(t-3) + y^2(t-2)} \\ &+ \frac{(0.5 * (u^2(t-2) + y^2(t-3))) * y(t-2)}{1 + u^2(t-2) + y^2(t-1)} \\ &+ \frac{\sin(u(t-1) * y(t-3)) * y(t-3)}{1 + u^2(t-1) + y^2(t-3)} \\ &+ \frac{\sin(u(t-1) * y(t-2)) * y(t-4)}{1 + u^2(t-2) + y^2(t-2)} \\ &+ u(t-1) \end{aligned} \quad (23)$$

The desired output in this example is a piecewise function.

$$y^*(t) = \begin{cases} 0.4493y^*(t-1) + 0.57r(t-1) \\ \qquad t \in [1,100] \cup [151,200] \\ 0.7\text{sign}(0.4493y^*(t-1) \\ \qquad +0.57r(t-1)) \\ \qquad t \in [101,150] \end{cases} \quad (24)$$

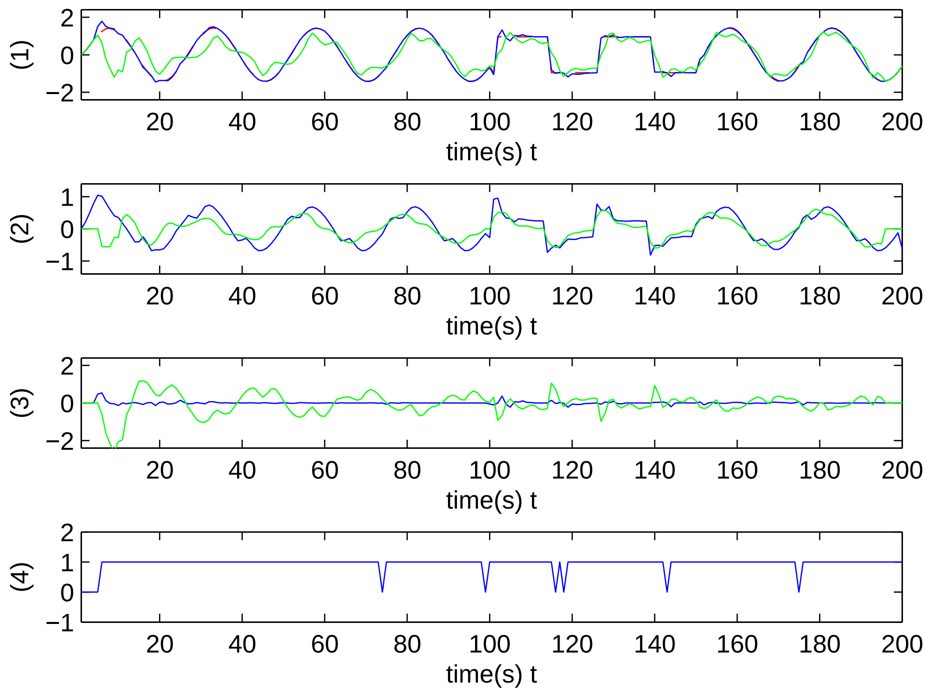

Fig. 2 switching control results of example 1

where $r(t) = 1.2 * \sin(2\pi t/25)$. At the nonlinear part, a neural network with one hidden layer and 20 hidden nodes as in Ref.[4] is used and other parameters satisfy m=4, n=3, c=1.2 and N=3. The quasi-ARX model can be trained off-line by the hierarchical training algorithm as in Ref.[4]. This model is used on-line as an identifier which is adjusted by BP algorithm. The linear part, m=4, n=3 which is adopted on-line by above section mentioned algorithm.

Figure 2 gives the results of example. In Fig.2(1), the red line is desired output, blue line proposed method control output and green pine linear control output. The Fig.2(2) gives the control input where blue and green denotes the proposed method control and linear control input. The errors are shown in Fig.2(3). The mean of linear control is -0.0364 and the variance is 0.2930. The mean of the proposed method control is 0.0035 and the variance is 0.0053. Therefore, our method is better than linear control. The switching sequence is presented which 1 is model with nonlinear part and 0 is model without nonlinear part in Fig.2(4).

VIII. CONCLUSION

In this paper, a new framework is established to adaptive control nonlinear system based on quasi-ARX neural network, and a switching algorithm is designed. Different from some relative researches which established more than two prediction models and made switching among so many corresponding controllers[7, 8], the proposed method is simpler and control-easier because of the compact and efficient structure of control system. Simulations are given to show the effectiveness of the proposed method both on stability and accuracy.

REFERENCES

[1] Hornik MK and White H(1989), Multilayer feedforward networks are universal approximators. Neural networks 2(1):359–366.

[2] Narendra K and Parthasarathy K(1990), Identification and control of dynamical systems using neural networks. IEEE Trans. on Neural Networks 1(1):4–27.

[3] Hu JL, Hirasawa K and Kumamaru K(1999), Adaptive predictor for control of nonlinear systems based on neurofuzzy models. Proc. of European Control Conference (ECC'99) (Karlsruhe) 8 1999.

[4] Hu JL and Hirasawa K (2004), A method for applying neural networks to control of nonlinear systems. Neural Information Processing: Research and Development. Springer 5:351–369.

[5] Chen F and Khalil H (1995), Adaptive control of a class of nonlinear discrete time systems using neural networks. IEEE Trans. on Automatic Control 40(5):791–801.

[6] Goodwin GC and Sin KS(1984), Adaptive Filtering Prediction and Control. Prentice-Hall, Inc..

[7] Chen L and Narendra KS(2001), Nonlinear adaptive control using neural networks and multiple models. Automatica 37:1245–1255.

[8] Fu Y and Chai T(2007), Nonlinear multivariable adaptive control using multiple models and neural networks. Automatica 43:1101–1110.