15th Triennial World Congress, Barcelona, Spain

A QUASI-ARX MODEL INCORPORATING NEURAL NETWORK FOR CONTROL OF NONLINEAR SYSTEMS

Jinglu Hu * **Kotaro Hirasawa** * **Kousuke Kumamaru** **

* *Department of Electrical and Electronic Systems Engineering, Kyushu University, Hakozaki 6-10-1, Higashiku, Fukuoka 812-8581, Japan*
TEL: (+81)92-642-3956, FAX: (+81)92-642-3962
E-mail: jinglu@cig.ees.kyushu-u.ac.jp
** *Department of Control Engineering and Science, Kyushu Institute of Technology, Kawazu 680-4, Iizuka 812, Japan*

Abstract: *Neural networks have been known as flexible nonlinear black-box models and have attracted much interest in control community. This paper introduces a new neural-network based prediction model for control of nonlinear systems. Distinctive features of the new model to the conventional neural-network based ones are that it has not only meaningful interpretation on part of its parameters but also is linear for the input variables. The former feature makes the parameter estimation easier and the latter allows deriving a nonlinear controller directly from the identified prediction model. The modeling and the parameter estimation are described in detail. The usefulness of the new model is demonstrated by applying it to control of two simulated nonlinear black-box systems.*

Keywords: Neural network, nonlinear model, nonlinear control, system modeling, parameter estimation.

1. INTRODUCTION

Neural networks have recently attracted much interest in system control community because they learn any nonlinear mapping (Narendra and Parthasarathy, 1990). Many approaches have been proposed to apply neural networks to prediction and control of general nonlinear systems (MillerIII *et al.*, 1990; Sjöberg *et al.*, 1995; Zhang and Wang, 2001). In most of these approaches, neural networks are used directly as nonlinear models (Narendra and Parthasarathy, 1990; Chen and Chen, 1993). However, there are two major criticisms on using neural network models; one is that they do not have useful interpretations in their parameters, especially for multilayer perceptron (MLP) network (Benitez *et al.*, 1997); the other is that they do not have structures favorable to certain applications such as controller design and system analysis (Narendra and Mukhopadhyay, 1997; Chen and Chen, 1993). In this paper, a new neural-network based model is introduced; the motivation is intended to introduce a desirable model structure and to make part of model parameters meaningful.

In many areas of linear control and some areas of nonlinear control, the theory and design methodology are well studied. In such areas neural networks can play a supportive role to synthesize and tune feedback controllers autonomously instead of replacing conventional controllers. This approach, combining the existing results in the literature of control theory and neural networks, is more acceptable by engineers and practitioners (Zhang and Wang, 2001). Following this consideration, a quasi-ARX prediction model is developed with incorporating MLP network. The model has some linear properties similar to linear ARX model, which are useful for control design and system analysis.

It has been shown that a general nonlinear system can be represented as ARX like regression form by using mathematical transformations such as Taylor expansion (Hu *et al.*, 2001). Such an ARX macro-model has "state dependent coefficients". The "state dependent coefficients" are then parameterized by using a multi-input and multi-output (MIMO) MLP network. The model obtained in this way is called quasi-ARX model, which has useful interpretation in part of its parameters. Moreover, by introducing a virtual input variable, the quasi-ARX model is transformed into one linear in the input variables, which is a useful linearity for controller design.

There are many estimation algorithms available for neural-network based models (Sjöberg and Ljung, 1995; Ljung, 1999). However, it has been found that these methods are not efficient for the quasi-ARX prediction model. To solve this problem, a hierarchical estimation algorithm is introduced. The algorithm consists of two learning loops, corresponding to the meaningful part and the meaningless part of model parameters. Numerical simulation results show that that the proposed dual learning algorithm is efficient to solve local minimum problem and can solve overfitting problem to some extent (Hu *et al.*, 2001).

The paper is organized as follows: Section 2 describes the system to deal with. Section 3 proposes the quasi-ARX prediction model. Section 4 introduces a dual loop learning algorithm for parameter estimation. Section 5 applies the model to control of two numerical systems. Finally, Section 6 presents Conclusions.

2. SYSTEM DESCRIPTION

Let us consider a single input single output (SISO) nonlinear time-invariant system whose input-output relation described by

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

$$\varphi(t) = [y(t-1) \ldots y(t-n_y)\, u(t-1) \ldots u(t-n_u-d+1)]^T \tag{2}$$

where $y(t)$ is the output at time t ($t = 1, 2, \ldots$), $u(t)$ the input, $\varphi(t)$ the regression vector with known order (n_y, n_u), d the known integer time delay, $e(t)$ the disturbance, and $g(\cdot)$ the unknown nonlinear function. We further introduce the following assumptions:

- Assumption 1, the elements of the regression vector $\varphi(t)$ are bounded;
- Assumption 2, $g(\cdot)$ is a continuous function, but at a small region around $\varphi(t) = 0$, it is C^∞ continuous.

3. QUASI-ARX PREDICTION MODEL

Our aim is to develop a neural-network based prediction model similar both in the form and in the properties to a linear ARX prediction model.

3.1 *ARX Like Macro-Model*

Introduce two polynomials $A(q^{-1}, \phi(t))$ and $B(q^{-1}, \phi(t))$, defined by

$$A(q^{-1}, \phi(t)) = 1 - a_{1,t}q^{-1} - \ldots - a_{n_y,t}q^{-n_y}$$
$$B(q^{-1}, \phi(t)) = b_{0,t} + b_{1,t}q^{-1} - \ldots - b_{n_u-1,t}q^{-n_u+1}$$

where coefficients $a_{i,t}$ and $b_{i,t}$ are nonlinear functions of a regression vector $\phi(t) = [y(t) \ldots y(t-n_y+1)\, u(t) \ldots u(t-n_u-d+2)]^T$. By using Taylor expansion and other mathematical transformations, it is easy to show that the system (1) can be represented by an ARX macro-model

$$A(q^{-1}, \phi(t-d)) * y(t, \phi(t-d)) = g(0) + B(q^{-1}, \phi(t-d))q^{-d}u(t) + e(t) \tag{3}$$

where $y(t, \phi(t-d))$ is the model output. '$*$' is a new multiplication operator, for which $A(q^{-1}, \phi(t))$ and $B(q^{-1}, \phi(t))$ are commutable, e.g., $q^{-1} * A(q^{-1}, \phi(t)) = A(q^{-1}, \phi(t)) * q^{-1}$, see (Hu *et al.*, 2000; Hu *et al.*, 2001) for more details

THEOREM *For a system described by* (3), *the one-step-ahead prediction,* $y^o(t+d|t, \phi(t))$, of $y(t)$ *satisfies*

$$y^o(t+d|t, \phi(t)) = y_\phi + \alpha(q^{-1}, \phi(t))y(t) + \beta(q^{-1}, \phi(t))u(t) \tag{4}$$

where

$$y^o(t+d|t, \phi(t)) = y(t+d) - F(q^{-1}, \phi(t))e(t+d),$$
$$y_\phi = F(q^{-1}, \phi(t))g(0),$$
$$\alpha(q^{-1}, \phi(t)) = G(q^{-1}, \phi(t))),$$
$$= \alpha_{0,t} + \alpha_{1,t}q^{-1} + \ldots + \alpha_{n_y-1,t}q^{-(n_y-1)},$$
$$\beta(q^{-1}, \phi(t)) = F(q^{-1}, \phi(t)) * B(q^{-1}, \phi(t)),$$
$$= \beta_{0,t} + \beta_{1,t}q^{-1} + \ldots + \beta_{n_u+d-2,t}q^{-n_u-d+2},$$

and $G(q^{-1}, \phi(t))$, $F(q^{-1}, \phi(t))$ *are unique polynomials satisfying*

$$F(q^{-1},\phi(t)) * A(q^{-1},\phi(t))$$
$$= 1 - G(q^{-1},\phi(t))q^{-d}$$

Proof: See (Hu *et al.*, 2001).

3.2 *Linearity for the Input Variable* $u(t)$

Based on linear control system theory, it is easy to derive a control law if the prediction model used is linear in the control variable $u(t)$. However, a general nonlinear prediction model is nonlinear for the variable. To solve this problem, a *virtual variable* $x(t)$ and an assumption are introduced.

- *Assumption 3*:
  The variable $u(t)$ can be synthesized by a unknown function defined by $u(t) = \rho(\xi(t))$, where $\xi(t) = [y(t)\ ...\ y(t-n_1+1)\ x(t+d)\ ...\ x(t-n_3+d+1)\ u(t-1)\ ...\ u(t-n_2)]^T$.

Remark: It is clear that if a system is controllable, the assumption 3 is satisfied. $\rho(\cdot)$ can be seen as a unknown control law, and $x(t)$ is considered as reference signal. It is reasonable to choose $n_1 = n_y$, $n_2 = n_u + d - 2$, $n_3 = 1$, which gives $\xi(t) = [y(t)\ ...\ y(t-n_y+1)\ x(t+d)\ u(t-1)\ ...\ u(t-d+2)]^T$.

By replacing the variable $u(t)$ in $\phi(t)$ with $\rho(\cdot)$, a prediction model linear in the variable $u(t)$ is obtained

$$y^o(t+d|t,\xi(t))$$
$$= y_\xi + \alpha(q^{-1},\xi(t))y(t) + \beta(q^{-1},\xi(t))u(t) \quad (5)$$

where y_ξ is y_ϕ whose variable $u(t)$ is replaced by $\rho(\cdot)$. By introducing $\Phi(t) = [1\, y(t) ... y(t-n_y+1)\, u(t) ... u(t-n_u-d+2)]^T$, and $\Theta_\xi = [y_\xi\, \alpha_{0,t} \cdots \alpha_{n_y-1,t}\, \beta_{0,t} \cdots \beta_{n_u+d-2}]^T$, the ARX-like macro-model (5) is expressed by

$$y^o(t+d|t,\xi(t)) = \Phi^T(t)\Theta_\xi. \quad (6)$$

3.3 *Incorporation of Neural Network*

The macro-model (6) is not feasible at this stage because the elements of Θ_ξ are unknown nonlinear function of $\xi(t)$, which must be parameterized. In contrast with our previous work (Hu *et al.*, 2001), in the new method we will parameterize the elements of Θ_ξ using MLP network. A significant advantage of using neural network is that it can be used to deal with higher dimensional problems.

Parameterizing Θ_ξ with a multi-input multi-output (MIMO) MLP network, the quasi-ARX prediction model is expressed by

$$\mathcal{M} : y^o(t+d|t,\xi(t)) = \Phi^T(t)\mathcal{N}(\xi(t),\Omega) \quad (7)$$

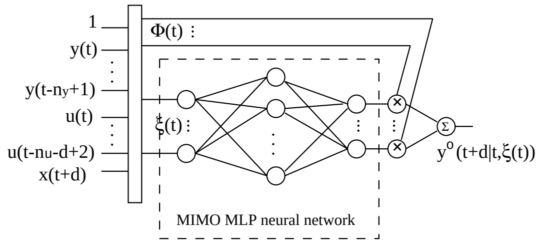

Fig. 1. The quasi-ARX prediction model incorporating MLP network.

where $\mathcal{N}(\cdot,\cdot)$ is a 3-layer MLP network with n input nodes, M sigmoidal hidden nodes and $n+1$ *linear* output nodes [1]. Figure 1 shows the quasi-ARX prediction model incorporating MLP network.

Let us express the 3-layer MLP network by

$$\mathcal{N}(\xi(t),\Omega) = W_2\Gamma(W_1\xi(t)+B)+\theta \quad (8)$$

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, $\theta \in \mathcal{R}^{(n+1)\times 1}$ is the bias vector of output nodes, and Γ is the diagonal nonlinear operator with identical sigmoidal elements σ (i.e., $\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}}$). Then the quasi-ARX prediction model (7) is expressed in a form of

$$\mathcal{M}: \quad y^o(t+d|t,\xi(t)) =$$
$$\Phi^T(t)\cdot W_2\Gamma(W_1\xi(t)+B) + \Phi^T(t)\theta. \quad (9)$$

The quasi-ARX prediction model consists of two parts: the second term of the right side of (9) is a linear ARX prediction model part, while the first 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. This makes θ be distinctive to other parameters. This feature allows us to use a dual loop leaning algorithm for the estimation.

4. HIERARCHICAL ALGORITHM

In the quasi-ARX prediction model, parameter vector θ describes a linear approximation. Therefore, it is natural to estimate θ in a different way to other parameters. Let $z_L(t) = y^o(t+d|t,\xi(t)) - \Phi^T(t)W_2\Gamma(W_1\xi(t)+B)$ and $z_N(t) = y^o(t+d|t,\xi(t)) - \Phi^T(t)\theta$, then the model (9) may be decomposed into the following two submodels

$$\mathcal{SM}1: \ z_L(t) = \Phi^T(t)\theta \quad (10)$$
$$\mathcal{SM}2: \ z_N(t) = \Phi^T(t)W_2\Gamma(W_1\xi(t)+B) \quad (11)$$

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

where $z_L(t)$ is regarded as the output of linear submodel (10), and $z_N(t)$ the output of nonlinear submodel (11). In this way, a hierarchical training algorithm can be introduced for the parameter estimation, described by

- *Step 1*: set $\theta = 0$, and *small* initial values to W_1, W_2, and B.
- *Step 2*: calculate $z_L(t)$, then estimate θ for submodel $\mathcal{SM}1$ by using a recursive least square (RLS) algorithm as described in (Ljung and Söderström, 1983).
- *Step 3*: calculate $z_N(t)$, then estimate W_1, W_2 and B for submodel $\mathcal{SM}2$. This is realized by using the well-known backpropagation (BP) algorithm, *but the BP is only performed for a few epochs L.*
- *Step 4*: stop if pre-specified condition is met, otherwise go to Step 2 and repeat the estimation of θ, and W_1, W_2, B.

The hierarchical algorithm consists of two loops. One is RLS estimation of θ, which does not have drawbacks such as overfitting and getting stuck at local minimum. The other loop is BP estimation of W_1, W_2, and B, which suffers problems of overfitting and local minimum. The hierarchical learning can reduce the risk of getting stuck local minima. Moreover, reducing the BP estimation step L in each iteration, the role of BP estimation is reduced. This is found to have role to solve the overfitting problem to some extent. For small noisy data sets, the step L should be small, see (Hu and Hirasawa, 2001).

5. NUMERICAL SIMULATIONS

The quasi-ARX prediction model is applied to control two simulated nonlinear systems in order to show its usefulness.

5.1 *Numerical Systems to Be Controlled*

Example 1: The system to be controlled is assumed to have a dead zone, shown in Fig.2. The linear part of the system is described by

$$G(q^{-1}) = \frac{0.7q^{-1} - 0.68q^{-2}}{1 - 1.72q^{-1} + 0.74q^{-2}}, \qquad (12)$$

while the nonlinear element is a dead zone described by

$$z(t) = \begin{cases} u(t) - 1.75 & \text{if } u(t) > 2 \\ 0.0625 \times \text{sign}(u(t)) \times u^2(t) & \text{if } |u(t)| \leq 2 \\ u(t) + 1.75 & \text{if } u(t) < -2 \end{cases}$$

Example 2: The system is a nonlinear one governed by

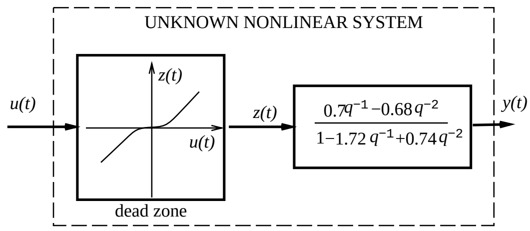

Fig. 2. A unknown nonlinear system with dead zone in the input.

$$y(t) = f[y(t-1), y(t-2), \\ y(t-3), u(t-1), u(t-2)] \quad (13)$$

where

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2}.$$

The above two systems are treated as unknown nonlinear ones. No prior information concerning system nonlinearity is used in the following simulations.

Table 1. Specifications of quasi-ARX models used for *Example 1* and *Example 2*

	n_y	n_u	d	M	MLP	NOP
Example 1	2	2	1	15	$N_{4\times15\times5}$	155
Example 2	3	2	1	20	$N_{5\times20\times6}$	246

5.2 *Identification of Prediction Models*

The quasi-ARX prediction models used for *Example 1* and *Example 2* are described by (9), and their specification are shown in Tab.1, where NOP denotes number of parameters.

To identify the models, from each system we record 1000 input-output data sets by exciting the system with a kind of random input sequence. Figure 3 shows the first 300 data sets taken from *Example 1*. Note that there is a virtual input variable in the models, which is corresponding to the reference signal in a control system. A kind of random sequence shown in Fig.3(c) is used for this input variable in order to obtain quasi-ARX prediction models that are able to deal with any reference signals in the control systems.

The hierarchical algorithm described in Section 4 is used to estimate model parameters. For each example, 200 iterations are carried out, where BP estimation step L is 500. When a noise signal is used as disturbance, one should decrease the value of L and increase the total iterations. Figure 4 shows the root mean square (RMS) error for the estimations; solid line is the result of *Example 1*, and dashed line the result of *Example 2*.

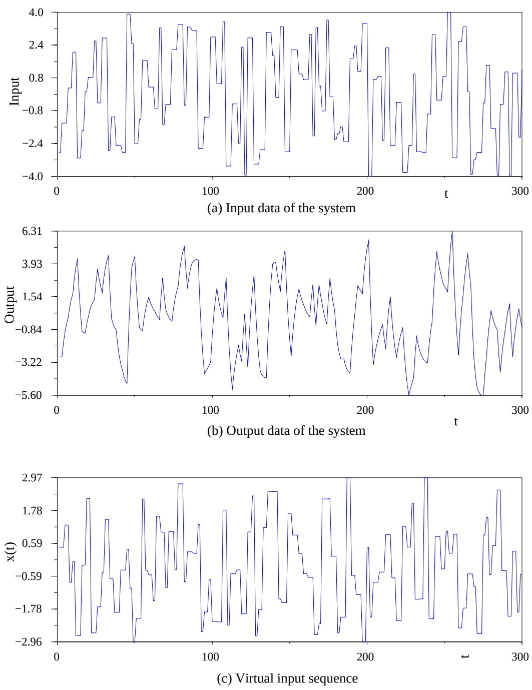

Fig. 3. (a) and (b) Input-output data taken from the system in *Example 1*; (c) A kind of random input sequence used for the virtual input variable of the quasi-ARX prediction model.

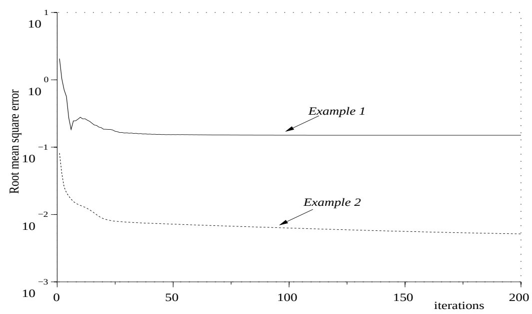

Fig. 4. Root mean square errors for estimation of the quasi-ARX prediction models for *Example1* (solid line) and *Example 2* (dashed line).

5.3 *Control of the Systems*

The systems are then controlled by applying the identified quasi-ARX prediction models

$$y^o(t+1|t,\xi(t)) = \Phi^T(t)\hat{\Theta}_\xi \tag{14}$$

where $\hat{\Theta}_\xi = [\hat{y}_\xi\ \hat{\alpha}_{0,t} \ldots \hat{\alpha}_{n_y-1,t}\ \hat{\beta}_{0,t} \ldots \hat{\beta}_{n_u+d-2,t}]^T$.

5.3.1. *Control Systems* Let us consider a minimum prediction error control by minimizing a criterion function defined by

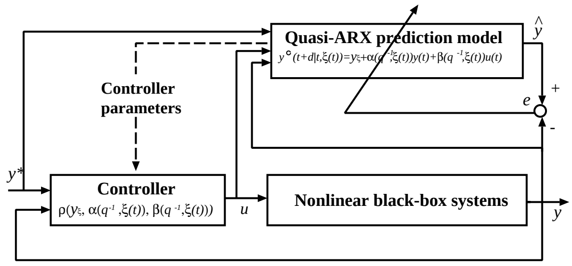

Fig. 5. Quasi-ARX model based control system for unknown nonlinear systems.

$$J(t+d) = \\ \frac{1}{2}\left[(y(t+d) - y^*(t+d))^2 + \lambda u(t)^2\right] \tag{15}$$

where $y^*(t)$ is reference signal, and $\lambda = 0.001$ is weighting factor for the control input. Using the identified quasi-ARX models (14) as prediction models, we can easily obtain controllers by the minimization of (15)

$$u(t) = \frac{\hat{\beta}_{0,t}}{\hat{\beta}_{0,t}^2 + \lambda}\left\{\left[(\hat{\beta}_{0,t} - \hat{\beta}(q^{-1},\xi(t)))q\right]u(t-1)\right. \\ \left. + y^*(t+1) - \hat{\alpha}(q^{-1},\xi(t))y(t) - \hat{y}_\xi\right\}, \tag{16}$$

due to the linearity of the quasi-ARX model for the variable $u(t)$. In the control law (16), as an element of $\xi(t)$, the virtual input variable of prediction model $x(t+d)$ is replaced by using reference signal $y^*(t)$. Figure 5 shows the quasi-ARX model based control system for unknown nonlinear systems.

5.3.2. *Control Results* The aim of the control system is to track the reference signals $y^*(t)$. When identifying the quasi-ARX prediction models, random sequences have been used for the virtual input variable $x(t)$. The control laws (16) are able to deal with any reference signals.

The two systems in *Example 1* and *Example 2* are rather nonlinear. Linear ARX prediction model has been found to be not able to control the systems well. Because of page limitation, we only show the results using the quasi-ARX prediction model. Figure 6 and 7 show the control results for *Example 1* and *Example 2* by using the proposed quasi-ARX prediction model, respectively. Although the reference signals have suddenly changes at $t = 100$ and $t = 150$, the controlled systems can track the reference signal quite well; the RMS control errors are 0.158 and 0.050, respectively.

6. CONCLUSIONS

A new neural-network based prediction model is proposed for control of black-box nonlinear

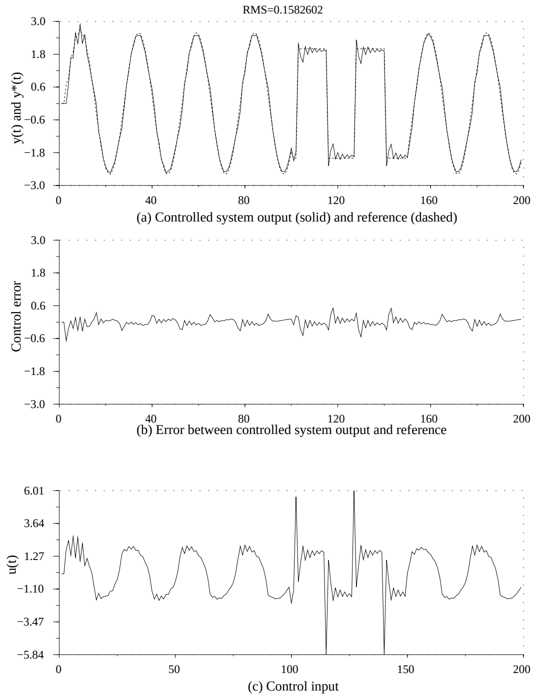

Fig. 6. Control results for *Example 1*.

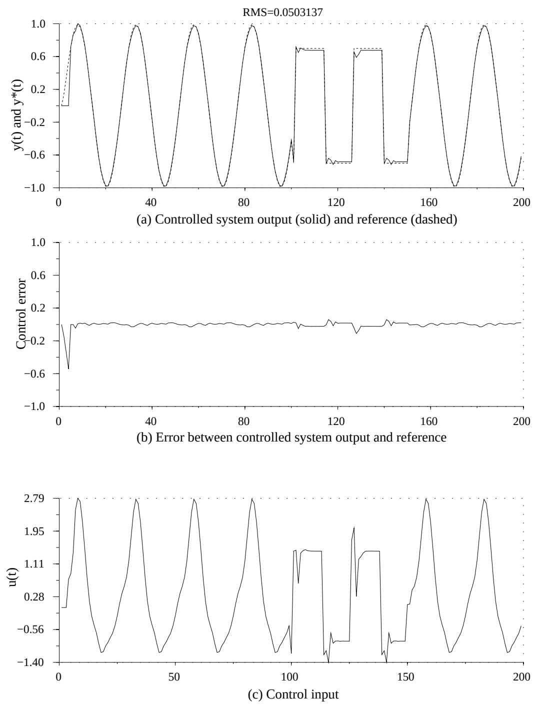

Fig. 7. Control results for *Example 2*.

systems. In the new model, MLP network is not used directly as models, but is embedded in an ARX macro-model. One of distinctive features of the new quasi-ARX prediction model is that it is linear for the input variables, which must be synthesized in a control system. This feature is very useful for controller design. The usefulness of the new model has been confirmed by applying it to control of two simulated nonlinear systems.

7. REFERENCES

Benitez, J.M., J.L. Castro and I. Requena (1997). Are artificial neural networks black boxes?. *IEEE Trans on Neural Networks* **8**(5), 1156–1164.

Chen, C.L. and W.C. Chen (1993). Self-organizing neural control system design for dynamic processes. *INT. J. SYSTEMS SCI.* **24**(8), 1487–1507.

Hu, J. and K. Hirasawa (2001). A hierarchical method for training embedded sigmoidal neural networks. In: *Lecture Notes in Computer Science 2130 (Proc. of ICANN'01)*. pp. 937–942.

Hu, J., K. Kumamaru and K. Hirasawa (2000). Quasi-ARMAX modeling approaches to identification and prediction of nonlinear systems. In: *Proc. of the 12th IFAC Symp. on Identification (Santa Barbara)*. (CDROM: FrMD3-2).

Hu, J., K. Kumamaru and K. Hirasawa (2001). A quasi-ARMAX approach to modeling of nonlinear systems. *International Journal of Control* **74**(18), 1754–1766.

Ljung, L. (1999). *System Identification: Theory for the User, Second Edition*. Prentice-Hall PTR, Upper Saddle River, N.J.

Ljung, L. and T. Söderström (1983). *Theory and Practice of Recursive Identification*. MIT press, Cambridge, Mass.

MillerIII, W.T., R.S. Sutton and P.J. Werbos (1990). *Neural Networks for Control*. The MIT Press, Massachusetts.

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

Narendra, K.S. and S. Mukhopadhyay (1997). Adaptive control using neural networks and approximate models. *IEEE Trans. on Neural Networks* **8**(3), 475–485.

Sjöberg, J. and L. Ljung (1995). Overtraining, regularization and searching for a minimum, with application to neural networks. *INT. J. Control* **62**(6), 1391–1407.

Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Deglon, P.Y. Glorennec, H. Hjalmarsson and A. Juditsky (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica* **31**(12), 1691–1724.

Zhang, Y. and J. Wang (2001). Recurrent neural networks for nonlinear output regulation. *Automatica* **37**(8), 1161–1173.