Artif Life Robotics (2014) 19:22–26
DOI 10.1007/s10015-013-0137-6

SPECIAL FEATURE: ORIGINAL ARTICLE

Modified fuzzy adaptive controller applied to nonlinear systems modeled under quasi-ARX neural network

Imam Sutrisno · Mohammad Abu Jami'in · Jinglu Hu

Received: 6 October 2013 / Accepted: 25 November 2013 / Published online: 14 December 2013

Abstract In this article, a fuzzy adaptive controller approach is presented for nonlinear systems. The proposed quasi-ARX neural network based on Lyapunov learning algorithm is used to update its weight for prediction model as well as to modify fuzzy adaptive controller. The improving performances of the Lyapunov learning algorithm are stable in the learning process of the controller and able to increase the accuracy of the controller as well as fast convergence of error. The simulations are intended to show the effectiveness of the proposed method.

Keywords Fuzzy adaptive controller · Lyapunov learning algorithm · Quasi-ARX neural network

1 Introduction

In the past decades, there has been much interest in the stabilizing adaptive control of dynamical systems. However, the stabilization of adaptive control in dynamical systems is a major challenge because the plants are always nonlinear. Hence, the performance of linear control models cannot satisfy the requirements. For this reason, some nonlinear prediction models have been developed for nonlinear systems to meet the system consideration. Until now, neural networks (NNs), wavelet networks (WNs) and radial basis function networks (RBFNs) [1] have been directly used to identify and control nonlinear dynamical systems because of their abilities to approximate arbitrary mapping to any desired accuracy. However, it still has some difficulties in parameter identification, controller design and stability guarantee when using these control systems. To simplify the identification for controlling, in our previous work a quasi-ARX neural network (QARXNN) model with a switching mechanism was studied for nonlinear system adaptive control [2] [3]. It can satisfy the stability and performance requirements by only using one model.

Lyapunov learning algorithm is applied as algorithm in QARXNN with multilayer perceptron (MLP) kernel to update its weight [4]. Lyapunov based on the controller can make the closed-loop system globally stable [5]. Lyapunov method is used to estimate the asymptotic stable region in controller design by genetic algorithm [6].

In this article, an efficient controller design is proposed for a nonlinear system modeled under QARXNN using Lyapunov learning algorithm. First, Lyapunov function is applied as algorithm in QARXNN with MLP kernel to update its weight for the prediction model. Then, a fuzzy switching algorithm is designed between the linear and nonlinear controllers from the modified QARXNN prediction. This fuzzy switching mechanism is different from the other fuzzy control because it is only used in control model and dependent on a switching criterion function. Last, the control system stability is proved. The proposed method has only a few number of learning, uniform distribution and the bounded modeling error characteristic.

This article is organized as follows. Section 2 briefly describes the problem to be solved. Section 3 discusses the

I. Sutrisno (✉) · M. A. Jami'in · J. Hu (✉)
Graduate School of Information, Production and Systems, Waseda University, Kitakyushu-shi, Fukuoka 808-0135, Japan
e-mail: imams3jpg@moegi.waseda.jp

J. Hu
e-mail: jinglu@waseda.jp

I. Sutrisno · M. A. Jami'in
Politeknik Perkapalan Negeri Surabaya (PPNS), Surabaya, Indonesia

controller design in detail. Section 4 provides numerical simulations to demonstrate the effectiveness of the proposed method, and Sect. 5 presents the conclusions.

2 Statement of the problem

Consider a single-input single-output (SISO) time invariant whose input–output relationship is described by

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

$$\phi(t) = [y(t-1)\cdots y(t-n_y)u(t-1)\cdots u(t-n_u)]^T \tag{2}$$

where $u(t) \in R, y(t) \in R, e(t) \in R$ are the system output, the system input and stochastic noise of zero mean at $t = 1, 2, 3, \ldots$, respectively. $g(\cdot): R^{n=n_u+n_y} \rightarrow R$ is an unknown continuous function (black-box), and $\phi(t) \in R^n$ is the regression vector composed of delays of the input–output data. The number of input variables n is equal to the sum of n_u and n_y. The noise of the system $e(t)$ is added to the unknown function input output of the system.

Assumption 1 (i) The input and output of training data are bounded, while $g(\phi(t))$ is the unknown continuous function. (ii) The system has a global uniformly asymptotically stable zero dynamics.

3 Implementation of an efficient controller

3.1 Lyapunov function

The learning algorithm based on Lyapunov function is described as follows:

Step 1. Setting $\theta = 0$, small initial values of W_1 and W_2, $k = 1$.

Step 2. Calculating z_l, then the θ can be estimated by SM1 model using least square error (LSE) method.

Step 3. Calculating an error, $z_n = \aleph(\Omega^*, \xi(t))$,

$$e(k) = \aleph(\Omega^*, \xi(t)) - \aleph(\Omega, \xi(t)) \tag{3}$$

where k = the sequence of learning number, $e(k) = [e_1 e_2 \cdots e_{n+1}]$, $\aleph(\Omega^*, \xi(t)) = [\aleph_1^* \aleph_2^* \cdots \aleph_{n+1}^*]$, $\aleph(\Omega, \xi(t)) = [\aleph_1 \aleph_2 \cdots \aleph_{n+1}]$

Step 4. Choosing Lyapunov function candidate, whose function is stated as $V(k) = f(e(k))$, where $V(k) = 0$ only if $e(k) = 0$ and $V(k) > 0$ only if $e(k) \neq 0$.

Step 5. Updating the weights of MLP neural network from the output layer to the input layer based on $\Delta V(k) = V(k) - V(k-1) < 0$. According to the Lyapunov theory, if $V(k) > 0$ and $V(k) < 0$, the error output will converge to zero as time goes to infinity.

$$\lim_{k\rightarrow\infty} e(k) = 0 \tag{4}$$

Step 6. Stop if pre-specified condition is met, otherwise go to step 2. Set $k = k + 1$.

The weight matrices in the first and second layers can be calculated based on Lyapunov function candidate expressed as

$$V(k) = \beta^k e^2(k) \tag{5}$$

where β is the positive constant value and $\beta > 1$, k is the k sequence of the learning number.

3.2 Proposed controller

The controller design includes two stages; the first stage is identification of QARXNN prediction model and the next stage is the derivation of the control law.

The identified QARXNN prediction model from previous work [2] is described by

$$\hat{y}(t+d \mid t, \xi(t)) = \Psi^{\mathrm{T}}(t)\hat{\theta} + \chi(t)\Psi^{\mathrm{T}}(t).\hat{W}_2\Gamma(\hat{W}_1\xi(t) + \hat{B}) \tag{6}$$

where $\widehat{\theta}, \hat{W}_1, \hat{W}_2$ and $\hat{B}$ are used for controller design.

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

$$M(t+1) = \left[\frac{1}{2}(y(t+d) - y^*(t+d))^2 + \frac{\lambda}{2}u(t)^2\right] \tag{7}$$

where λ is weighting factor for the control input.

Therefore, the controllers can be achieved by solving

$$\frac{\partial M(t+1)}{\partial u_i} = 0 \quad i = 1, 2 \tag{8}$$

By solving (8), two controllers can be derived based on QARXNN prediction model as follows:

$$C_1: u_l(t) = \frac{\hat{b}_0^1}{\hat{b}_0^1\hat{b}_0^1 + \lambda}\left((\hat{b}_0^1 - \hat{b}^1(q^{-1})q\right)u(t-1) + y^*(t+1) - \hat{a}^1(q^{-1})y(t)) \tag{9}$$

$$C_2: u_n(t) = \frac{\hat{\beta}_{0,t}}{\hat{\beta}_{0,t}^2 + \lambda}\left((\hat{\beta}_{0,t} - \hat{\beta}(q^{-1})q\right)u(t-1) + y^*(t+1) - \hat{\alpha}\left(q^{-1}, \xi(t)\right) y(t)) \tag{10}$$

where

$$\hat{a}(q^{-1}) = \hat{a}_0 + \hat{a}_1 q^{-1} + \cdots + \hat{a}_{n-1}q^{-n+1};$$
$$\hat{b}(q^{-1}) = \hat{b}_0 + \hat{b}_1 q^{-1} + \cdots + \hat{b}_{m+d-2}q^{-m-d+2};$$
$$\hat{\alpha}(q^{-1}, \xi(t)) = \hat{\alpha}_{0,t} + \hat{\alpha}_{1,t}q^{-1} + \cdots + \hat{\alpha}_{n-1,t}q^{-n+1};$$
$$\hat{\beta}(q^{-1}, \xi(t)) = \hat{\beta}_{0,t} + \hat{\beta}_{1,t}q^{-1} + \cdots + \hat{\beta}_{m+d-2}q^{-m-d+2};$$

The coefficients can be obtained as follows:

$$\left[\hat{\alpha_0}\cdots\hat{\alpha}_{n-1}\hat{\beta_0}\cdots\hat{\beta}_{m+d-2}\right]=\hat{\theta}$$

$$\left[\hat{\alpha_{0,t}}\cdots\hat{\alpha}_{n-1,t}\hat{\beta_{0,t}}\cdots\hat{\beta}_{m+d-2,t}\right]=\hat{\theta}+\hat{W}_2\Gamma(\hat{W}_1\xi(t)+\hat{B})$$

Most switching control based on two or more controllers has been proposed in [1, 3]. An integer switching law was introduced into control model as the function ξ (t) in the prediction model. It means that the linear and nonlinear controllers are alternately used in the system. Therefore the switching criterion function can be expressed as:

$$J_i(t)=\sum_{l=k}^{t}\frac{a_i(l)(\|e_i(l)\|^2-4\Delta^2)}{2(1+a_i(l)\Psi(l-k)^{\mathrm{T}}P_i(l-k-1)\Psi(l-k))}+c*\sum_{l=t-N+1}^{t}(1-a_i(l)\|e_i(l)\|^2),\quad i=1,2 \tag{11}$$

where N is an integer, and $c\geq 0$ is a predefined constant. Now, the expression of switching law $\chi(t)$ based on the switching criterion function can be explained as

$$\chi(t)=\begin{cases}1, & J_1(t)>J_2(t);\\ 0, & \texttt{otherwise}\end{cases} \tag{12}$$

To choose the controller in linear and nonlinear part, the $J_1(t)$ and $J_2(t)$ is compared. If $J_1(t)>J_2(t)$, the nonlinear part is added to the controller, or else only the linear part is used to control. However, the jumping control will reduce the precision and adaptability of the control system. Motivated by the fuzzy function theory, this paper introduces a fuzzy membership function $v(t)$ based on the criterion function $J_1(t)$ and $J_2(t)$. Now, a fuzzy switching controller [7] is obtained based on the fuzzy membership functions $v(t)$,

$$C: u(t)=(1-v(t))u_l(t)+v(t)u_n(t) \tag{13}$$

Firstly, the switching law $\xi(t)$ is calculated from input and output signals and model errors; then it will be used in the model identification. The fuzzy switching law $v(t)$ is calculated from input and output signals and model errors; then it will be used in the control model.

3.3 Stability

The stability analysis of the proposed nonlinear controller system can be described as follows:

Theorem 1 *For the system (1) with adaptive fuzzy switching 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 First, similar to [7],

$$\lim_{N\to\infty}\sum_{t=1}^{N}\frac{a(t)^2(e_1(t)^2-4\Delta)}{2(1+\Psi(t-d)^{\mathrm{T}}\Psi(t-d))}<\infty, \tag{14}$$

and

$$\lim_{N\to\infty}\frac{a(t)^2(e_1(t)^2-4\Delta)}{2(1+\Psi(t-d)^{\mathrm{T}}\Psi(t-d))}\to 0 \tag{15}$$

Along with **Assumption 1** (iii) $e_1(t)$ is bounded.

By (11) and (14), 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 Eq. (15); (ii) $J_2(t)$ is unbounded.

Since (1) $J_1(t)$ is bounded, there is a constant t_0 such that $v(t)=1$, $\forall t>t_0$. The model also has bounded error $e(t)$. From the above inequalities, the input and output of the closed-loop switching control system are bounded. The linear part is always bounded.

If a proper nonlinear part is chosen and the accurate parameters are adjusted, the model error $e_2(t)$ can converge on zero. On the other hand, a constant T_0 satisfies $v(t)=0$, $\forall t>T_0$. Then, the tracking error of the model can converge on zero.

4 Numerical simulations

In this proposed design, the system is a nonlinear one governed by

$$y(t)=g[y(t-1),y(t-2),y(t-3),u(t-1),u(t-2)]+v(t) \tag{16}$$

where $g(.)$ is the nonlinear function with a disturbance

$$g[x_1,x_2,x_3,x_4,x_5]=\frac{x_1x_2x_3x_5(x_3-1)+x_4}{1+x_2^2+x_3^2} \tag{17}$$

The desired output in this example is a piecewise function:

$$y^*(t)=\begin{cases}0.6y^*\ (t-1)+r(t-1)\\ \quad t\in[1,100]\cup[151,200]\\ 0.7\,\mathtt{sign}\ (0.4493y^*(t-1)+0.57r(t-1)\\ \quad t\in[101,150]\end{cases} \tag{18}$$

where $r(t)=\sin(2\pi t/25)$.

In Fig. 1, the black dotted line is the desired output, while the red solid line denotes the proposed method output $y(t)$. In Fig. 1a, the magenta dashed line shows the linear control output $y_0(t)$. Obviously, the control output of the proposed method is nearly consistent with the desired output along of the time that is better than the linear. In Fig. 1b, blue dashed line shows the 0/1 switching control output $y_1(t)$. Clearly, the proposed control output is almost coincident with the desired output. It can also be found that the 0/1 switching control results have some wobble at the

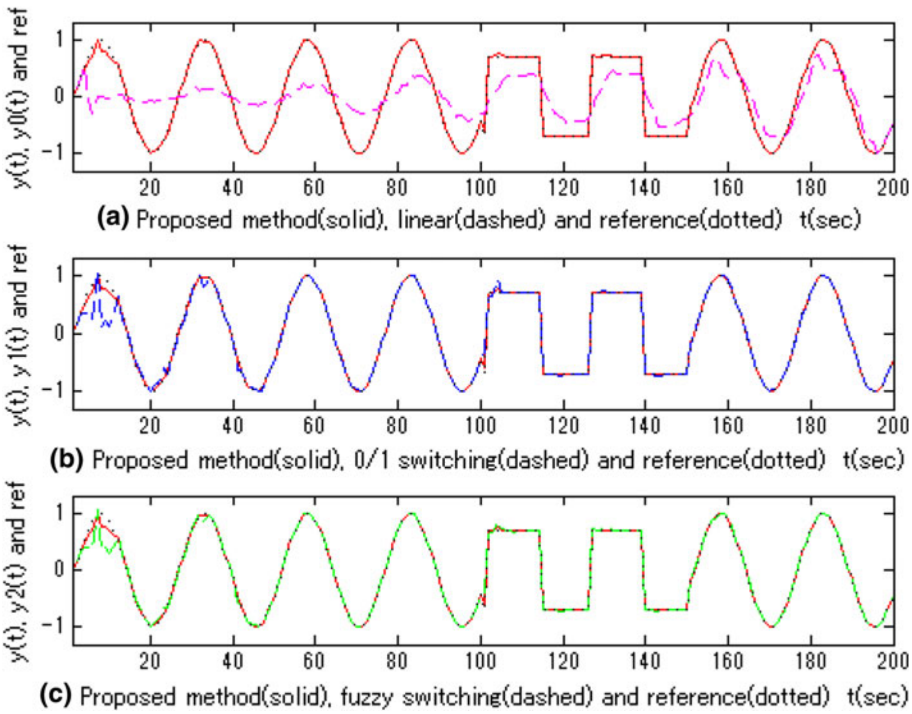

Fig. 1 **a** Simulation result of the proposed method compared with linear control and **b** simulation result of the proposed method compared with 0/1 switching. **c** Simulation result of the proposed method compared with fuzzy switching reference [7]

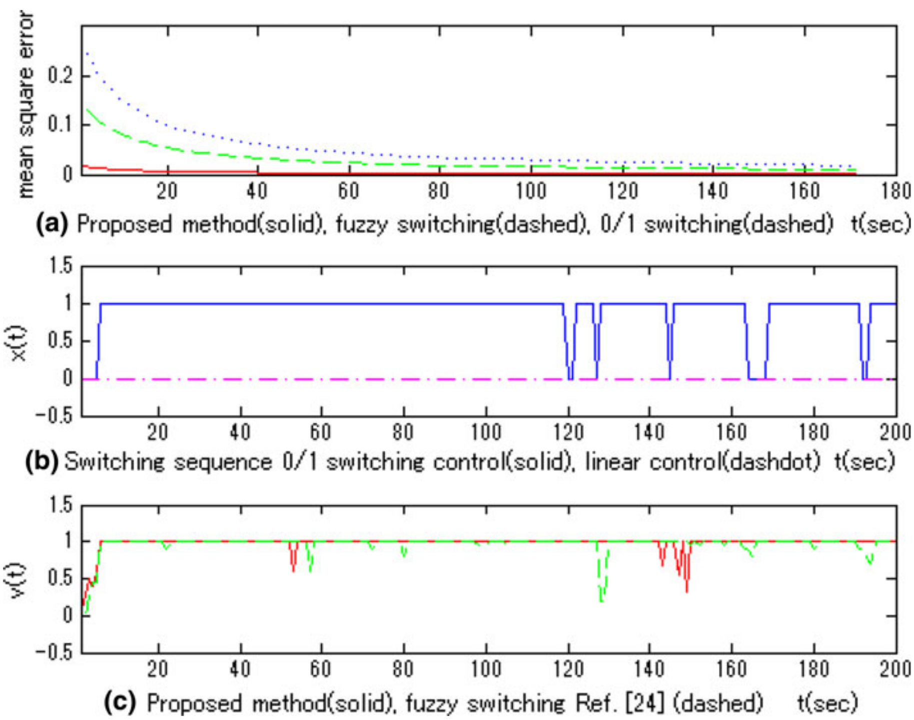

Fig. 2 **a** Convergence characteristics of the errors. **b** Switching sequence. **c** Fuzzy switching sequence

last half time. In Fig. 1c, the green dashed line shows the fuzzy switching control output $y_2(t)$. From the figure, it is shown that the proposed control method can do better than fuzzy switching controller.

A similar conclusion can also be obtained from the convergence characteristic of the error shown in Fig. 2a. The switching sequence is presented in Fig. 2b and fuzzy switching sequence is shown in Fig. 2c.

Table 1 gives the errors of the three methods; the proposed control method has a better accuracy and the error of the proposed method is smaller than the other methods.

Table 1 Comparison results of the errors

Method	RMSE	Variances	Accuracy
Proposed control	0.0109	0.023	95.78
Fuzzy switching control	0.0147	0.047	94.55
0/1 switching control	0.0201	0.082	91.41
Linear control	0.0240	0.105	81.23

5 Conclusions

This study has successfully demonstrated the effectiveness of the proposed fuzzy adaptive controller using Lyapunov learning algorithm based on QARXNN prediction model. First, the principles of QARXNN prediction model was derived. Second, the network structure and theoretical base of the proposed method have been adopted to adapt the Lyapunov learning algorithm to replace the traditional trial-and-error method.

Finally, the control performance of the proposed method based on QARXNN prediction model has been confirmed by experimental result. The main contributions of this study are: (1) the successful development of an improved fuzzy switching controller; (2) the successful adoption of a Lyapunov learning algorithm; (3) the successful application of the fuzzy switching controller based on QARXNN prediction model to control nonlinear system with robust control performance.

Acknowledgements This research has been supported by DIKTI and PPNS.

References

1. Wang L, Cheng Y, Hu J (2012) Stabilizing switching control for nonlinear system based on quasi-ARX RBFN model. IEEJ Trans Electr Electron Eng 7(4):390–396
2. Wang L, Cheng Y, Hu J (2010) A quasi-ARX neural network with switching mechanism to adaptive control of nonlinear systems. SICE J Control Meas Syst Integr 3(4):246–252
3. Sutrisno I, Jami'in MA, Hu J (2012) Neural predictive controller of nonlinear systems based on quasi-ARX neural network. In: Proceedings of the 2012 International Conference on Automation and Computing (ICAC 18th 2012) (Loughborough UK), pp 83–88
4. Jami'in MA, Sutrisno I, Hu J (2012) Lyapunov learning algorithm for quasi-ARX neural network to identification of nonlinear dynamical system. In: Proceedings of The 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC2012) (Seoul), pp 3141–3146
5. Meza C, Biel D, Jeltsema D, Scherpen JMA (2012) Lyapunov-based control schema for single-phase grid-connected PV central inverters. IEEE Trans Control Syst Technol 20(2):520–529
6. Loop BP, Sudhoff SD, Zak SH, Zivi EL (2010) Estimating regions of asymptotic stability of power electronics systems using genetic algorithms. IEEE Trans Control Syst Technol 18(5):1011–1022
7. Wang L, Cheng Y, Hu J (2010) Nonlinear adaptive control using a fuzzy switching mechanism based on improved quasi-ARX neural network. In: Proceedings of The 2010 International Joint Conference on Neural Networks (IJCNN2010) (Barcelona), pp 1–7