11.pdf

Nonlinear Model-Predictive Control Based on Quasi-ARX Radial-Basis Function-Neural-Network

Imam Sutrisno
Graduate School of Information Production and Systems
Waseda University
Kitakyushu 808-0135 Japan and
Politeknik Perkapalan Negeri Surabaya
Email: imams3jpg@moegi.waseda.jp

Mohammad Abu Jami'in
Graduate School of Information Production and Systems
Waseda University
Kitakyushu 808-0135 Japan and
Politeknik Perkapalan Negeri Surabaya
Email: mohammad@ruri.waseda.jp

Jinglu Hu
Graduate School of Information Production and Systems
Waseda University
Kitakyushu 808-0135 Japan
Email: jinglu@waseda.jp

Mohammad Hamiruce Marhaban
Department of Electrical and Electronics Engineering
Faculty of Engineering
Universiti Putra Malaysia
43400 Serdang Selangor, Malaysia
Email: mhm@upm.edu.my

Norman Mariun
Centre for Advance Power and Energy Research
Faculty of Engineering
Universiti Putra Malaysia
43400 Serdang Selangor, Malaysia
Email: norman@upm.edu.my

***Abstract*—A nonlinear model-predictive control (NMPC) is demonstrated for nonlinear systems using an improved fuzzy switching law. The proposed moving average filter fuzzy switching law (MAFFSL) is composed of a quasi-ARX radial basis function neural network (RBFNN) prediction model and a fuzzy switching law. An adaptive controller is designed based on a NMPC. a MAFFSL is constructed based on the system switching criterion function which is better than the (ON/OFF) switching law and a RBFNN is used to replace the neural network (NN) in the quasi-ARX black box model which is understood in terms of parameters and is not an absolute black box model, in comparison with NN. The proposed controller performance is verified through numerical simulations to demonstrate the effectiveness of the proposed method.**

Keywords–nonlinear model-predictive control; moving average filter fuzzy switching law ; quasi-ARX radial basis function neural network;

I. INTRODUCTION

Adaptive control has a lot of interest in recent years[1][2][3]. Therefore, many nonlinear black box models have been used to control nonlinear systems. There are problems of the controller designed and the stability of whole systems. Zeng et al.[4] established a neural network (NN) predictive control scheme for studying the coagulation process of waste water treatment in a paper mill, and Wang et al. presented adaptive NN model based predictive control[5].

A quasi-ARX model embodies an ARX-like macro model part, a kernel part and a controller was designed based on the predictive model. The kernel part is an ordinary network model, such as NNs, wavelet networks (WNs), neuro-fuzzy networks (NFNs) and RBFNNs to parameterize the nonlinear coefficients of macro model[6]. However, there are still some aspects needed to be improved in the above control method. First, on the (ON/OFF) hard switching control method which is not very smooth; the second is the assumption of global boundedness also can be relaxed; the third is the parameters of quasi-ARX NN model to be adjusted on-line are highly nonlinear, which deteriorates the adaptability of control system[7].

Motivated by the above aspects, a MAFFSL is constructed based on the system switching criterion function which is better than the (ON/OFF) switching law and a RBFNN is used to replace the NN in the quasi-ARX black box model which is understood in terms of parameters and is not an absolute black box model, in comparison with NN. The simulation includes two parts: the fuzzy switching control, results based on quasi-ARX NN model and the fuzzy switching control based on quasi-ARX RBFNN model. The simulation results show that the proposed control model and method based on the three improvements have better control performance[6].

The rest of the paper is organized as a NMPC is developed to control nonlinear systems based on a quasi-ARX RBFNN prediction model using MAFFSL. In Section 2, a quasi- ARX RBFNN prediction model is introduced. Section 3 describes the NMPC design using MAFFSL. Numerical simulations are carried out to show the effectiveness of the proposed method in Section 4 comparison with previous improved Elman neural network-particle swarm optimization(IENN-PSO), fuzzy switching control, (ON/OFF) switching control and linear control. Finally, Section 5 presents the conclusions.

978-1-4799-6487-1/14 $31.00 © 2014 IEEE
DOI 10.1109/AMS.2014.30

II. Quasi-ARX Radial Basis Function Neural Network Prediction Model

A. ARX-Like Expression

Consider a nonlinear time-invariant dynamical system with single-input-single-output (SISO) whose input-output relation described by:[7]

$$y(t+d) = g(\vartheta(t)) \tag{1}$$

$$\vartheta(t) = [y(t+d-1), \cdots y(t+d-n), u(t), \cdots, u(t-m+1)]^T \tag{2}$$

where $y(t)$ is the output at the time $t (t = 1, 2, ...)$, $u(t)$ is the input, d is the known integer time delay, $\vartheta(t)$ is the regression vector and n, m are the system orders. $g(\cdot)$ is a nonlinear function and at a small region around $\vartheta(t) = 0$, it is C^∞ continuous, then $g(0) = 0$ [6].

Under the continuous condition, the unknown nonlinear function $g(\vartheta(t))$ can be found by Taylor series expansion on a small region around $\vartheta(t) = 0$:

$$y(t+d) = g'(0)\vartheta(t) + \frac{1}{2}\vartheta^T(t)g''(0)\vartheta(t) + \cdots \tag{3}$$

where the prime denotes differentiation with respect to $\vartheta(t)$. Then the following notations are introduced:[7]

$$(g'(0) + \frac{1}{2}\vartheta^T(t)g''(0)+)^T = [a_{1,t} \cdots a_{n,t} \;\; b_{0,t} \cdots b_{m-1,t}]^T \tag{4}$$

where $a_{i,t} = a_i(\vartheta(t))(i = 1, \cdots, n)$ and $b_{j,t} = b_j(\vartheta(t))(j = 0, \cdots, m-1)$ are nonlinear functions of $\vartheta(t)$.

Therefore to get $y(t+d)$ by using the input-output data up to time t in a model. The coefficients $a_{i,t}$ and $b_{j,t}$ need to be mathematically tractable using the input-output data up to time t. It could be found by iteratively replace $y(t+l)$ in the expressions of $a_{i,t}$ and $b_{j,t}$ with functions:

$$y(t+l) \Rightarrow g(\tilde{\vartheta}(t+l)), l = 1, \cdots, d-1 \tag{5}$$

where $\tilde{\vartheta}(t+l)$ is $\vartheta(t+l)$ whose elements $y(t+k), l+1 < k \le d-1$ are replaced by Eq.(5) and define the new expressions of the coefficients by:

$$a_{i,t} = \tilde{a}_{i,t} = \tilde{a}_i(\phi(t)), b_{j,t} = \tilde{b}_{j,t} = \tilde{b}_j(\phi(t)) \tag{6}$$

where $\phi(t)$ is a vector:

$$\phi(t) = [y(t) \cdots y(t-n+1)u(t) \cdots u(t-m-d+2)]^T \tag{7}$$

Two polynomials $A(q^{-1}, \phi(t))$ and $B(q^{-1}, \phi(t))$ based on the coefficients are introduced,

$$A(q^{-1}, \phi(t)) = 1 - a_{1,t}q^{-1} - \cdots - a_{n,t}q^{-n} \tag{8}$$

$$B(q^{-1}, \phi(t)) = b_{0,t} + \cdots + b_{m-1,t}q^{-m+1} \tag{9}$$

where q^{-1} is a backward shift operator, e.g. $q^{-1}u(t) = u(t-1)$. Then the nonlinear system Eq.(1) can be equivalently represented as the following ARX-like expression:[2]

$$A(q^{-1}, \phi(t))y(t+d) = B(q^{-1}, \phi(t))u(t) \tag{10}$$

By the Eq.(10), $y(t+d)$ satisfies the following equation as in[11]:

$$y(t+d) = \alpha(q^{-1}, \phi(t))y(t) + \beta(q^{-1}, \phi(t))u(t) \tag{11}$$

where

$$\begin{aligned} \alpha(q^{-1}, \phi(t)) &= G(q^{-1}, \phi(t)) = \alpha_{0,t} + \alpha_{1,t}q^{-1} \\ &+ \cdots + \alpha_{n-1,t}q^{-n+1} \end{aligned} \tag{12}$$

$$\begin{aligned} \beta(q^{-1}, \phi(t)) &= F(q^{-1}, \phi(t))B(q^{-1}, \phi(t)) \\ &= \beta_{0,t} + \beta_{1,t}q^{-1} + \cdots \\ &+ \beta_{m+d-2,t}q^{-m-d+2} \end{aligned} \tag{13}$$

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

$$F(q^{-1}, \phi(t))A(q^{-1}, \phi(t)) = 1 - G(q^{-1}, \phi(t))q^{-d} \tag{14}$$

B. Quasi-ARX Radial Basis Function Neural Network

As we know, a controller can be derived easily and can share parameters from the identified prediction model, when the prediction model is linear in the input variable $u(t)$[6]. However, the Eq.(16) is a general one which is nonlinear in the variable $u(t)$, because the $\tilde{\theta}^n_\Psi$ are based on $\Psi(t)$ whose elements contain $u(t)$. To solve this problem, an extra variable $x(t)$ is introduced and an unknown nonlinear function $\rho(\xi(t))$ is used to replace the variable $u(t)$ in $\tilde{\theta}^n_\Psi$. The function $\rho(\xi(t))$ exists, where $\xi(t)$ is:

$$\begin{aligned} \xi(t) &= [y(t) \cdots y(t-n_1)x(t+d) \cdots \\ & \quad x(t-n_3+d)u(t-1) \cdots u(t-n_2)]^T \end{aligned} \tag{15}$$

including the extra variable $x(t+d)$ as an element[7]. A typical choice for n_1, n_2 and n_3 in $\xi(t)$ is $n_1 = n+d-1, n_2 = m+2d-2$ and $n_3 = 0$. We can express the Eq.(16) by:

$$\Delta y(t+d) = \psi^T(t)\theta + \Psi^T(t)\theta^n_\xi \tag{16}$$

where $\theta^n_\xi = \tilde{\theta}^n_\Psi$.

The elements of θ^n_ξ are unknown nonlinear function of $\Phi(t)$, which can be parameterized by NN or RBFNN. In this paper the RBFNN used has a local property,

$$\theta^n_\xi = \sum_{j=1}^{M} \mathbf{w}_j R_j(\xi(t), \Omega_j) \tag{17}$$

where M is the number of RBFs, $\mathbf{w}_j = [\omega_{1j}, \omega_{2j}, \cdots, \omega_{Nj}]^T$ the coefficient vector, and $R_j(\xi(t), \Omega_j)$ the RBFs defined by:

$$R_j(\xi(t), \Omega_j) = e^{-\lambda_j \|\xi(t) - Z_j\|^2} \quad j = 1, 2, \cdots, M \tag{18}$$

where $\Omega_j = \lambda_j, Z_j$ is the parameters set of the RBFNN; Z_j is the center vector of RBF and λ_j are the scaling parameters; $\|\cdot\|^2$ denotes the vector two-norm. Then we can express the quasi-ARX RBFNN prediction model for Eq.(18) in a form of:

$$\Delta y(t+d) = \psi^T(t)\theta + \sum_{j=1}^{M} \Psi^T \mathbf{w}_j R_j(\xi(t), \Omega_j) \tag{19}$$

Now, the quasi-ARX RBFNN model is further expressed by

$$\Delta y(t+d) = \psi^T(t)\theta + \Psi^T(t)\mathbf{W}\Upsilon(\xi(t)) = \psi^T(t)\theta + \Xi(t)^T\Theta \tag{20}$$

where $\Theta = [w_{11} \cdots w_{n1} \cdots w_{1M} \cdots w_{nM}]^T$ and $\Xi(t) = \Upsilon(\xi(t)) \bigotimes \Psi(t)$.

Remark 1 the quasi-ARX RBFNN prediction model described by Eq.(21) is an accurate model of the system in d-difference form Eq.(16).

C. Parameter Estimation

According to the parameter property, the model parameters as in Eq.(24) can be divided into three groups: the linear parameter θ of the linear part $\psi^T(t)\theta$, the linear parameter Θ and the nonlinear parameter Ω_j of the nonlinear part $\Psi^T(t)\mathbf{W}\Upsilon(\xi(t))$. The nonlinear parameters Ω_j are determined off-line. Let us denote the estimation of Ω_j by $\hat{\Omega}_j$. In order to determine the centers and widths of the RBFNN, AP clustering method is employed. The center Z_j is the arithmetic mean value of all training data in each cluster. The width λ_j is ϱ times the largest distances between all training data in each cluster. The parameters θ and Θ are estimated by using on-line identification algorithms, respectively[6].

The linear parameter θ of linear part of model is updated as:

$$\hat{\theta}(t) = \hat{\theta}(t-d) + \frac{a(t)\psi(t-d)e_1(t)}{1 + \psi(t-d)^T\psi(t-d)} \tag{21}$$

where $\hat{\theta}(t)$ is the estimate of θ at time instant t, which also denotes the parameter of a linear model used to approximate the system in d-difference form. And

$$a(t) = \begin{cases} 1 & \texttt{if } |e_1(t)| > 2D \\ 0 & \texttt{otherwise} \end{cases} \tag{22}$$

where $e_1(t)$ denotes the error of the linear model, defined by

$$e_1(t) = \Delta y(t) - \psi(t-d)^T\hat{\theta}(t-d) \tag{23}$$

The linear parameter Θ of nonlinear part of the quasi-ARX RBFNN model is updated by a least square (LS) algorithm:

$$\hat{\Theta}(t) = \hat{\Theta}(t-d) + \frac{P(t)\Xi(t-d)e_2(t)}{1 + \Xi(t-d)^T P(t)\Xi(t-d)} \tag{24}$$

where $\hat{\Theta}(t)$ is the estimate of Θ at time instant t. $\hat{\Theta}(0) = \Theta_\circ$ is assigned with an appropriate initial value. $e_2(t)$ is the error of quasi-ARX RBFNN model, defined by

$$e_2(t) = \Delta y(t) - \psi(t-d)^T\hat{\theta}(t-d) - \Xi^T(t-d)\hat{\Theta}(t-d) \tag{25}$$

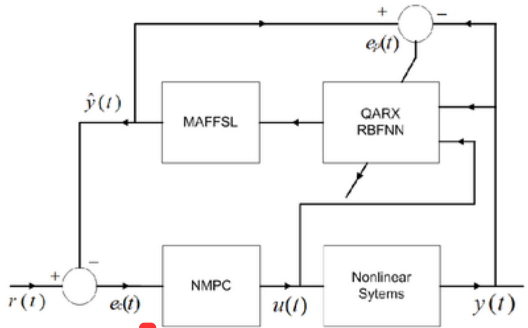

Figure 1. Quasi ARX RBFNN model based NMPC.

and

$$P(t) = \frac{P(t-d) - P^T(t-d)\Xi(t-d)^T\Xi(t-d)P(t-d)}{1 + \Xi(t-d)^T P(t)\Xi(t-d)} \tag{26}$$

no restriction is made on how the parameters $\hat{\Theta}(t)$ are updated except they always lie inside some pre-defined compact region $\hbar$:

$$\hat{\Theta}(t) \in \hbar \forall t \tag{27}$$

III. Nonlinear Model-Predictive Control Design

The purpose of NMPC is to design a control law and an updating law for the primary controller parameters, such that the system output y is to follow an input reference signal and the closed-loop dynamic performance of system follows the predictive model. Figure 1 shows the proposed NMPC scheme. A moving average filter fuzzy switching mechanism is employed to improve the performance of the quasi-ARX RBFNN prediction model. The structure of the proposed NMPC includes the input layer (i layer), the hidden layer (j layer), the context layer (r layer) and the output layer (o layer) with two inputs and one output [8]. The basic function and the signal propagation of each layer are introduced in the following: Layer 1 (input layer): the node input and the node output are represented as:

$$X_i(k) = f_i(\mathtt{net}_i) = \mathtt{net}_i = e_i(k) \tag{28}$$

where $e_i(k)$ and $X_i(k)$ are the input and the output of the input layer, respectively and k represents the k^{th} iteration.

Layer 2 (hidden layer): the node input and the node output are represented as:

$$X_j(k) = S(\mathtt{net}_j) \tag{29}$$

$$\mathtt{net}_j = \sum_i w_{ij}X_i(k) + \sum_r w_{rj}X_r^c(k) \tag{30}$$

where $X_j(k)$ and $\mathtt{net}_j$ are the output and the input of the hidden layer, w_{ij} and w_{rj} are the connective weights of input neurons to hidden neurons and context neurons to hidden neurons, respectively, X_r^c the output of the context layer, and $S(X)$ is sigmoid function, that is, $S(X) = 1/(1+e^{-X})$.

Layer 3 (context layer): the node input and the node output are represented as:

$$X_r^c(k) = \gamma X_r^c(k-1) + X_j(k-1) \tag{31}$$

where $0 \leq \gamma < 1$ is the self-connecting feedback gain. Layer 4 (output layer): the node input and the node output are represented as:

$$Y_o(k) = f(\mathtt{net}_o(k)) = \mathtt{net}_o(k) \tag{32}$$

$$\mathtt{net}_o(k) = \sum_j w_{jo} X_j(k) + w_o Y^c(k) \tag{33}$$

$$Y^c(k) = \zeta Y^c(k-1) + Y_o(k-1) \tag{34}$$

where $Y_o(k)$ the output of the NMPC and also the control effort of the proposed controller, $Y^c(k)$ the output of the output feedback neuron, $0 \leq \zeta < 1$ is the self-connecting feedback gain, w_{jo} and w_o are the connective weights of hidden neurons to output neurons and output feedback neuron to output neuron, respectively.

A. *Learning Algorithm*

The learning algorithm of the NMPC is referred to as the BP learning rule method. To describe using supervised gradient decent method, first the energy function E is defined as:[8]

$$E = \frac{1}{2}(y-\hat{y})^2 = \frac{1}{2}e \tag{35}$$

where y and $\hat{y}$ represents the output of the system and output of the prediction model, respectively, and e denotes the tracking error. Then the learning algorithm is defined as follows: Layer 4: the error term to be propagated is given by:

$$\begin{aligned}\delta_o &= -\frac{\partial E}{\partial Y_o(k)} = -\frac{\partial E}{\partial e}\frac{\partial e}{\partial Y_o(k)} \\ &= -\frac{\partial E}{\partial e}\frac{\partial e}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial Y_o(k)}\end{aligned} \tag{36}$$

$$\begin{aligned}\Delta w_o &= -\eta_1 \frac{\partial E}{\partial w_o} \\ &= -\eta_1 \frac{\partial E}{\partial Y_o(k)}\frac{\partial Y_o(k)}{\partial \mathtt{net}_o(k)}\frac{\partial \mathtt{net}_o(k)}{\partial w_o} \\ &= \eta_1 \delta_o Y^c(k)\end{aligned} \tag{37}$$

$$\begin{aligned}\Delta w_{jo} &= -\eta_2 \frac{\partial E}{\partial w_{jo}} \\ &= -\eta_2 \frac{\partial E}{\partial Y_o(k)}\frac{\partial Y_o(k)}{\partial \mathtt{net}_o(k)}\frac{\partial \mathtt{net}_o(k)}{\partial w_{jo}} \\ &= \eta_2 \delta_o X_j(k)\end{aligned} \tag{38}$$

The connective weights w_o and w_{jo} are updated according to the following equations, correspondingly:

$$w_o(k+1) = w_o(k) + \Delta w_o \tag{39}$$

$$w_{jo}(k+1) = w_{jo}(k) + \Delta w_{jo} \tag{40}$$

where the factors η_1 and η_2 are the learning rate. Layer 3: using chain rule, the update law of w_{rj} is:

$$\begin{aligned}\Delta w_{rj} &= -\eta_3 \frac{\partial E}{\partial w_{rj}} \\ &= -\eta_3 \frac{\partial E}{\partial Y_o(k)}\frac{\partial Y_o(k)}{\partial \mathtt{net}_o(k)}\frac{\partial \mathtt{net}_o(k)}{\partial X_j(k)}\frac{\partial X_j(k)}{\partial w_{rj}} \\ &= \eta_3 \delta_o w_{jo} X_j(k)[1-X_j(k)] X_r^c(k)\end{aligned} \tag{41}$$

$$w_{rj}(k+1) = w_{rj}(k) + \Delta w_{rj} \tag{42}$$

Layer 2: through using chain rule, the update law of w_{ij} is:

$$\begin{aligned}\Delta w_{ij} &= -\eta_4 \frac{\partial E}{\partial w_{ij}} \\ &= -\eta_4 \frac{\partial E}{\partial Y_o(k)}\frac{\partial Y_o(k)}{\partial \mathtt{net}_o(k)}\frac{\partial \mathtt{net}_o(k)}{\partial X_j(k)}\frac{\partial X_j(k)}{\partial w_{ij}} \\ &= \eta_4 \delta_o w_{jo} X_j(k)[1-X_j(k)] X_i(k)\end{aligned} \tag{43}$$

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij} \tag{44}$$

where the factors η_3 and η_4 are the learning rate, $(\eta_1, \eta_2, \eta_3, \eta_4)$ will be optimized by the PSO algorithm.

B. *Moving Average Filter Fuzzy Switching Law*

Consider a similar switching criterion function as in[9]:

$$\begin{aligned}J_i(t) &= \sum_{l=k}^{t} \frac{a_i(l)(\|e_i(l)\|^2 - 4\Delta^2)}{2(1+a_i(l)\Psi(l-k)^T P_i(l-k-1)\Psi(l-k))} \\ &+ c\sum_{l=t-N+1}^{t}(1-a_i(l)\,\|e_i(l)\|^2), \quad i=1,2\end{aligned} \tag{45}$$

where N is an integer and $c \geq 0$ is a predefined constant. The previous work [2] introduce a fuzzy switching parameter $\chi(t)$ based on the criterion function $J_1(t)$ and $J_2(t)$:

$$\chi(t) = \begin{cases} 1 & \mathtt{if}\ x(t) > K \\ x(t) & \mathtt{if}\ k \leq x(t) \leq K \\ 0 & \mathtt{if}\ x(t) < k \end{cases} \tag{46}$$

where $x(t) = J_1(t)(J_1(t)+J_2(t))^{-1}$, K and k are constants which satisfy $k \in (0, 0.5), K \in (0.5, 1)$. In order to make the switching control smooth, the first improvement that can be done is adding the moving average filter in the switching parameter $\chi(t)$, with the preset threshold parameters k and K to improve the accuracy and the adaptation in the controller by reducing the unreasonable switching in the control process. The switching parameter for MAFFSL is defined as:

$$\chi(t) = \begin{cases} 1 & \texttt{if } x(t) > K \\ \frac{1}{\sum_{t=0}^{M-1} \zeta_i} \sum_{t=0}^{M-1} \zeta_i \chi(t-1) & \texttt{if } k \le x(t) \le K \\ 0 & \texttt{if } x(t) < k \end{cases} \tag{47}$$

C. Lyapunov Stability Analysis for the Whole Systems

The following theorem states that the proposed NMPC convergent is based on Lyapunov stability theory.

Theorem Let the weights of proposed NMPC are updated along with Eqs.(37), (38), (41) and (43). Then the convergence of the proposed NMPC Eq.(34) is guaranteed if the learning rate η_i satisfy:

$$\eta_i = \frac{\beta [\sum_{o=1}^{m_y} e_o(k) \frac{\partial \hat{y}_o(k)}{\partial \varpi}]^T [\sum_{o=1}^{m_y} e_o(k) \frac{\partial \hat{y}_o(k)}{\partial \varpi}]}{[\sum_{o=1}^{m_y} \frac{\partial \hat{y}_o(k)}{\partial \varpi}]^T [\sum_{o=1}^{m_y} e_o(k) \frac{\partial \hat{y}_o(k)}{\partial \varpi}]} \tag{48}$$

where i=1,2,3,4 ; the condition are $0 < \beta < 2$ and $\varpi = [w_{11}^o \cdots w_{m_y n_h}^o \; w_{11}^i \cdots w_{n_h n_i}^i \; w_1^j \cdots w_{n_h}^j]^T$, proposed NMPC has n_i inputs, n_h hidden units and m_y output variables.

Proof. Let a Lyapunov function candidate be chosen as $l(k) = \sum_{o=1}^{m_y} e_o^2(k)$, and let $\delta l(k) = l(k+1) - l(k)$ and $\delta e_o(k) \equiv e_o(k+1) - e_o(k)$. Then:

$$\delta l(k) = 2 \sum_{o=1}^{m_y} e_o(k) \delta e_o(k) + \sum_{o=1}^{m_y} (\delta e_o(k))^2 \tag{49}$$

By the method in [10], $\delta e_o(k)$ can be represented by $\delta e_o(k) = [\partial e_o(k)/\partial \varpi]^T \delta \varpi$, while $\delta \varpi = -\eta_i (\partial y_o(k)/\partial \varpi)$ and the Eq.(49) becomes:

$$\begin{aligned} \delta l(k) &= 2 \sum_{o=1}^{m_y} e_o(k) \left[\frac{\partial e_o}{\partial \varpi}\right]^T \delta \varpi + \sum_{o=1}^{m_y} \left(\left[\frac{\partial e_o}{\partial \varpi}\right]^T \delta \varpi \right)^2 \\ &= -2\eta_i \left[\sum_{o=1}^{m_y} e_o(k) \frac{\partial \hat{y}_o(k)}{\partial \varpi} \right]^T \left[\sum_{o=1}^{m_y} e_o(k) \frac{\partial \hat{y}_o(k)}{\partial \varpi} \right] \\ &+ \eta_i^2 \left(\left[\sum_{o=1}^{m_y} \frac{\partial \hat{y}_o(k)}{\partial \varpi} \right]^T \left[\sum_{o=1}^{m_y} e_o(k) \frac{\partial \hat{y}_o(k)}{\partial \varpi} \right] \right)^2 \end{aligned} \tag{50}$$

To ensure this selected learning rate inside the stable region, the learning rate η_i was set as in Eq.(48) and $\delta l(k) < 0$ which shows that the proposed NMPC converged. This completes the proof of the theorem.

IV. Simulation

In order to study the behavior of the proposed control method, a numerical simulation is described in this section. The system is a nonlinear one governed by:

$$y(t) = f[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] \tag{51}$$

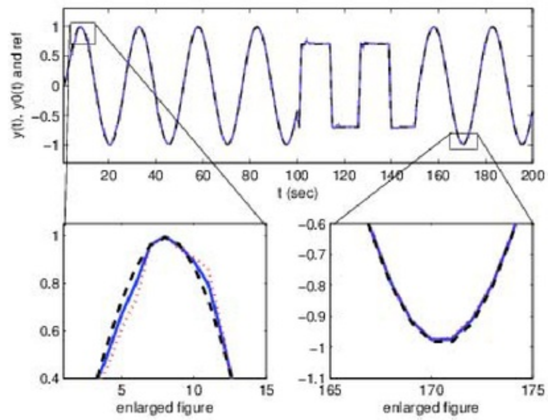

Figure 2. Quasi ARX RBFNN model based NMPC.

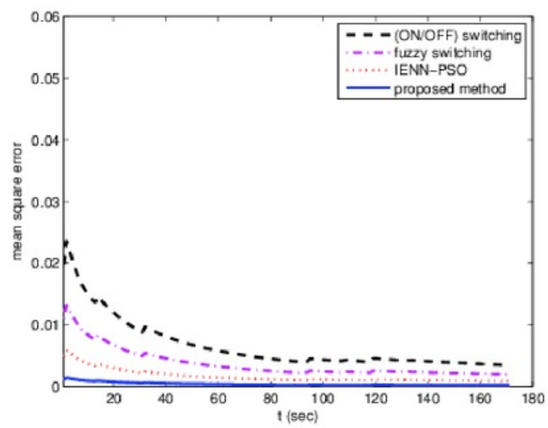

Figure 3. Convergence characteristics of the errors.

where:

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2} \tag{52}$$

The desired output in this example is a piecewise function:

$$y^*(t) = \begin{cases} 0.6y^*(t-1) \\ +r(t-1), & t \in [1,100] \cup [151,200] \\ 0.7\texttt{sign}(0.4493 \\ y^*(t-1) \\ +0.57r(t-1), & t \in [101,150] \end{cases} \tag{53}$$

where $r(t) = \sin(2\pi t/25)$. In the nonlinear part, the NN has one hidden layer and 20 hidden nodes and other parameters are set by m=4, n=3, d=1. Figure 2 shows the performance when the proposed controller is used. In the Fig. 2, the proposed control output is almost coincidence with the

TABLE I
COMPARISON RESULT OF THE ERRORS.

Method	Mean of errors	Variance of errors
Proposed control method	-0.0025	0.0011
IENN-PSO control method	-0.0035	0.0013
Fuzzy switching control method	-0.0044	0.0032
0/1 switching control method	-0.0051	0.0067

TABLE II
CLASSIFICATION RESULT FOR SEVERAL METHODS.

Method	Iteration	CPU Time	Accuracy
Proposed control method	143	0.31	98.08
IENN-PSO control method	153	0.36	97.38
Fuzzy switching control method	194	0.58	95.14
0/1 switching control method	253	1.02	93.87

desired output. It also can be found that the 0/1 switching control results have some wobble at the last half time. The similar conclusion were obtained from convergence characteristic of the errors is shown in Fig. 3. Table I gives the contrast of three methods errors. The error of the proposed control system is smaller than the other methods. Obviously, proposed controller has better performance than other controller. For additional comparison, convergence speed of different methods are given in Fig. 3, and classified in Table II, the proposed control method gets a better accuracy, and also has a faster convergence rate than other methods.

V. CONCLUSION

This study has successfully demonstrated the effectiveness of the NMPC controller based on QARX RBFNN prediction model using MAFFSL method. It can satisfy the stability, response and performance requirement with only one model used. For parameterizing the coefficients of the macro-model, a RBFNN is used in the kernel part to replace NN, thus nonlinear parameters of the proposed quasi-ARX RBFNN prediction model using MAFFSL method can be determined by a priori knowledge,then the prediction model only remains linear parameters to be adjusted on-line. Simulations are given to show the effectiveness of the proposed method on control stability, accuracy, response and robustness.

ACKNOWLEDGMENT

This research has been supported by Directorate General of Higher Education, Ministry of National Education Indonesia and Shipbuilding Institute of Polytechnic Surabaya (SHIPS).

REFERENCES

[1] Wang, S. W., Yu, D. L. Gomm, J. B., Page, G. F. and Douglas, S. S.: Adaptive neural network model based predictive control for air- fuel ratio of SI engines, Engineering Application of Artificial Intelligent, Vol. 19, pp. 189–200, SICE Publishing (2006)

[2] L. Wang, Y. Cheng and J. Hu: A Quasi-ARX Neural Network with Switching Mechanism to Adaptive Control of Nonlinear Systems, SICE Journal of Control, Measurement, and System Integration, vol. 3, no. 4, pp. 246-252 July (2010)

[3] I. Sutrisno, M.A. Jami'in and J. Hu: Modified fuzzy adaptive controller applied to nonlinear systems modeled under quasi-ARX neural network, Journal Artificial Life and Robotics, Springer, Japan, vol. 19, no. 1, pp. 22-26, February (2014)

[4] Zeng, G. M., Qin, X. S., He, L., Huang, G. H., Liu, H. L. and Lin, Y. P.: A neural network predictive control system for paper mill wastewater treatment, Engineering Application of Artificial Intelligent, Vol. 16 pp. 121–129 (2003)

[5] Yuan-Hai Chang: Predictive control based on recurrent neural network and application to plastic injection molding processes, IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society, Nopember (2007)

[6] L. Wang: Study on adaptive control of nonlinear dynamical systems based on quasi-ARX models, DSpace at Waseda University (2013)

[7] L. Wang, Y. Cheng and J. Hu: Stabilizing Switching Control for Nonlinear System Based on Quasi-ARX Model, IEEJ Trans. On Electrical and Electronic Engineering, vol. 7, no. 4, pp. 390-396 July (2012)

[8] F.-J. Lin, L.-T.Teng and H.Chu: Modified Elman neural network controller with improved particle swarm optimisation for linear synchronous motor drive, IET Elect. Power Appl., Vol. 2, No. 3, pp. 201–214 (2008)

[9] L. Chen and K.S. Narendra, Nonlinear adaptive control using neural networks and multiple models. Automatica, vol.37, no.8, pp.12451255, (2001)

[10] Ku, C.C., Lee, K.Y.: Diagonal recurrent neural networks for dynamical system control, IEEE Transactions on Neural Networks 6 (1),pp. 144-156 (1995)

[11] J. Hu and K. Hirasawa: A method for applying neural networks to control of nonlinear systems, Neural Information Processing: Research and Development, J. C. Rajapakse and L. Wang, Eds., pp. 351-369, Springer, 5 (2004)

11.pdf

ORIGINALITY REPORT

73%

SIMILARITY INDEX

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

★Wang, Lan. "Study on adaptive control of nonlinear dynamical systems based on quansi-ARX models", DSpace at Waseda University, 2013. 45%

Publications

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	OFF		