Correspondence

A Novel Neural Approximate Inverse Control for Unknown Nonlinear Discrete Dynamical Systems

Hua Deng and Han-Xiong Li

***Abstract*—A novel neural approximate inverse control is proposed for general unknown single-input-single-output (SISO) and multi-input-multi-output (MIMO) nonlinear discrete dynamical systems. Based on an innovative input/output (I/O) approximation of neural network nonlinear models, the neural inverse control law can be derived directly and its implementation for an unknown process is straightforward. Only a general identification technique is involved in both model development and control design without extra training (online or offline) for the neural nonlinear inverse controller. With less approximation made on controller development, the control will be more robust to large variations in the operating region. The robustness of the stability and the performance of a closed-loop system can be rigorously established even if the nonlinear plant is of not well defined relative degree. Extensive simulations demonstrate the performance of the proposed neural inverse control.**

***Index Terms*—Input/output approximation, inverse control, MIMO systems, neural networks (NNs), SISO systems, unknown discrete-time systems.**

I. INTRODUCTION

Nonlinear inverse control using neural networks (NNs) for unknown discrete-time nonlinear dynamical systems have received much attention in recent years [1]–[7], with particular interest in two design methods: neural inverse control (offline design) [1]–[5] and neural adaptive inverse control (online design) [6], [7]. Generally speaking, identification and controller designs are two major steps in inverse control development using NNs for unknown nonlinear discrete systems [2]. In the identification stage, a neural network NN_1 is first used to approximate the considered system. Static back-propagation can be used to train NN_1 [8]. In the controller design stage, another network NN_2 is trained with the help of NN_1 to determine the desired control signal in terms of the measured inputs and outputs as well as the given reference. However, static back-propagation cannot be used for training NN_2 since the controller is in the feedback loop of the dynamical system [2], [4]. Thus, dynamic gradient (DG) methods such as back-propagation through time [9] and real-time recurrent learning [10] need to be used. It is well-known that all DG methods are computationally intensive as compared to static back-propagation for computing the control signal $u(k)$ [4]. In neural adaptive inverse control, since the weights of NNs are adjusted online, special consideration must be given to the stability and convergence of the adaptation because they are not ensured [3], [6]. Gradient methods with small learning steps may not cause instability but will slow the training.

For offline design of neural inverse control, the main concern is to find a simple, effective and engineering-preferred procedure, which may lead to a satisfactory neural inverse controller. To avoid the intensive computation of using DG methods, various numerical inverse methods such as Newton's method are proposed to determine the inverse control. The Newton's method, which is an iterative optimization algorithm and introduced for neural network internal model control [11], requires the discrete time interval to be long enough for searching a suitable input. Moreover, necessary and sufficient convergence conditions cannot be ensured for such iterative optimization algorithms [12]. It is worth noting that a method of using approximate models is proposed in [4], in which an affine model is assumed and then two NNs are used to approximate general (nonaffine) nonlinear discrete SISO systems. Because the control signal occurs linearly, it can be computed directly from the approximate models. However, since the output of the original system is not linear to the input, the method requires the control signal $|u(k)|$ to be small enough to maintain reasonable accuracy of the affine model. Moreover, stable design methodologies based on such models remain to be developed [4]. An improvement is suggested in [13] to remove the requirement of a small $|u(k)|$ by assuming a special affine model for the original system. Since the special model is only valid under the requirement that $\Delta u(k) \rightarrow 0$, the applications are limited only to sufficiently slow trajectories. Because of the special affine model assumption, general identification techniques cannot be used. A special training procedure is needed to develop an NN controller, particularly for the MIMO case, in which a more complicated training procedure will be involved [13]. Therefore, the design and computational complexity still exist. Moreover, it requires the availability of outputs $y(k+d-1), y(k+d-2), \ldots, y(k+1)$ for delay $d > 1$, which may cause difficult implementation of the controller because usually only $y(k), y(k-1), \ldots y(k-n+1)$ are available at instant k. Besides, all the above alternative methods require that the relative degree [2] of a plant be well defined. If the relative degree of a plant is not well defined, it is shown in previous work [2] that desired control performance is difficult to achieve even though intensive computation has been made on training of the NN inverse controller.

Unlike the previously mentioned approaches, a novel NN approximate inverse control (AIC) is proposed without preassuming a special affine model for both SISO and MIMO general unknown nonlinear discrete systems. First, an NN model with general form is established by using general identification techniques. Then, a simple and innovative input/output (I/O) approximation is derived directly from the NN model and the neural inverse control law can be determined straightforwardly based on the I/O approximation. Since less approximation is assumed on the control development, large $|u(k)|$ and $|\Delta u(k)|$ are allowed, which results in a large working and stability region. Besides, the outputs $y(k+d-1), y(k+d-2), \ldots, y(k+1)$ are not required for delay $d > 1$ at instant k. With the proposed neural control law, the robustness of the stability and the performance of a closed-loop system can be rigorously established even if the nonlinear plant is of not well defined relative degree. In the proposed method, only general training algorithms, such as those available in neural network Toolbox in MATLAB, are used for both SISO and MIMO nonlinear discrete systems. Thus, the method proposed can facilitate the design of an NN controller and is easy for control engineers to use since no special training procedure is involved. Extensive simulations demonstrate the performance of the proposed NN control approach.

Manuscript received February 20, 2004; revised May 12, 2004. This work was supported in part by the RGC Hong Kong SAR Project CityU 1129/03E and the City University of Hong Kong SRG Project 7001477. This paper was recommended by Associate Editor W.-Y. Wang.

The authors are with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, P.R. China. (e-mail: mehxli@cityu.edu.hk).

Digital Object Identifier 10.1109/TSMCB.2004.836472

II. NN I/O Models for Nonlinear Discrete Systems

A. Input/Output Representation

State equations are widely used to represent dynamical systems. Consider a system described by following state equations:

$$\begin{aligned} x(k+1) &= f_s\left[x(k), u(k)\right] \\ y(k) &= h_s\left[x(k)\right] \end{aligned} \tag{1}$$

where $x(k) \in R^n$, $u(k) \in R^p$, and $y(k) \in R^p$ are the state, input, and output of the system, respectively, $f_s : R^n \times R^p \to R^n$, $h_s : R^n \to R^p$, and $f_s, h_s \in C^\infty$. Without loss of generality, $f_s(0,0) = 0$ and $h_s(0) = 0$ are assumed.

The relative degree [2] plays an important role in deriving an I/O representation of the system (1). Suppose that the relative degree for system (1) is d. A general I/O representation of the system (1) is as follows [2].

For SISO case

$$y(k+d) = f_0\left[w_k, u(k)\right] \tag{2}$$

where $w_k = [y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-n+1)]$

Similarly, for MIMO case

$$y(k+d) = F_0\left[v_k, u(k)\right] \tag{3}$$

where

$$\begin{aligned} y(k) &= \left[y_1(k)\; y_2(k), \ldots, y_p(k)\right]^T \\ u(k) &= \left[u_1(k),\; u_2(k),\; \ldots,\; u_p(k)\right]^T \\ v_k &= [y(k), y(k-1), \ldots, y(k-n+1) \\ &\qquad u(k-1), \ldots, u(k-n+1)] \\ y(k+d) &= \begin{bmatrix} y_1(k+d_1) \\ y_2(k+d_2) \\ \cdots \\ y_p(k+d_p) \end{bmatrix} \\ F_0\left[v_k, u(k)\right] &= \begin{bmatrix} f_{01}\left[v_k, u(k)\right] \\ f_{02}\left[v_k, u(k)\right] \\ \cdots \\ f_{0p}\left[v_k, u(k)\right] \end{bmatrix}. \end{aligned}$$

The models (2) and (3) are general representations of SISO and MIMO nonlinear discrete dynamical systems, respectively. An advantage to use the above models is that the availability of outputs $y(k+d-1), y(k+d-2), \ldots, y(k+1)$ for delay $d > 1$ is not required since they are usually not available at instant k. A different model requiring the outputs $y(k+d-1), y(k+d-2), \ldots, y(k+1)$ at instant k was used in [13] because of the pre-assumption of a special affine model. This requirement will cause difficult implementation of the controller. Besides, only a special case of nonlinear MIMO discrete systems was considered in [13], where $u_j(k)$ only affects $y_j(k+d_j)$ at instant k.

B. Neural I/O Models

Various NNs have been proposed. Among them, the dynamic multilayer NN has been widely used in modeling and control of nonlinear discrete dynamical systems [1]–[8], [14]–[17] due to its general approximation abilities [18], [19]. In this study, this type of NN is used and trained to represent the dynamic behavior of unknown nonlinear discrete systems as shown in Fig. 1.

For a general discrete-time nonlinear dynamical system, the nonlinear autoregressive moving average (NARMA) model is an exact representation of its I/O behavior over the range in which the system operates [4]. An NN as shown in Fig. 1 is used to approximate the SISO I/O representation (2) and is often called NN NARMA model as (4)

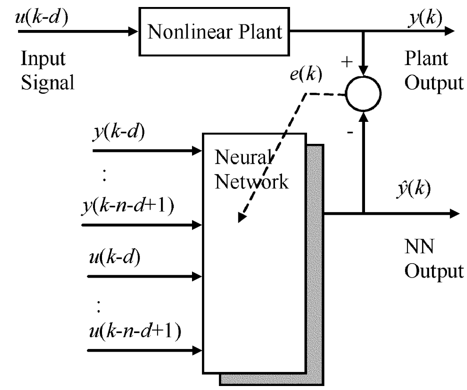

Fig. 1. NN model in input-output identification.

$$y(k+d) = f\left[w_k, u(k)\right] + \xi(k). \tag{4}$$

In (4), the weight vector of the neural network is omitted for simplicity. $\xi(k)$ is an approximation error and $|\xi(k)| \le \bar{\xi}$ with $\bar{\xi}$ being a small positive number. This NN model can be trained by the static back-propagation [8] to approximate the underlying SISO process in terms of the past inputs and outputs of the system. Generally, it is difficult to design a controller that results in a desired output for (4) since its output is nonlinearly dependent on the input. This is why, in nonlinear inverse control using NNs, another NN need to be trained with the DG methods to determine the desired control signal [2], [4].

In order to guarantee the universal approximation, both a suitable NN structure and sufficient learning time are needed for adaptive or non-adaptive NN nonlinear inverse control. It is well-known that finding the best NN (approximator) structure is a difficult and unsolved problem [20]. It is believed that a fairly large NN can deal with relatively complex approximation problems. Thus, in this study, a basic NN structure with two hidden layers is fixed as in [3], [4], and [8]. Using the same training data set, repetitive training experiments may be performed by adding hidden neurons in successive trials. Based on the learning performance, the number of hidden neurons can be roughly determined. There is no doubt that most of the training time is spent on neural inverse controller design because of the DG methods used. However, in the proposed AIC, such time consuming algorithms are totally avoided and thus the controller design is greatly simplified.

III. Novel Neural AIC

A novel neural AIC method is proposed here using a newly developed I/O approximation, which not only avoids complex control development and intensive computation, but also overcomes the shortcomings of other existing methods. A unique and rigorous stability proof will be given and its superior performance will be demonstrated in later simulations.

A. Novel Neural I/O Approximation

SISO Case: For the neural network NARMA model (4), the Taylor expansion of the neural network $f[w_k, u(k)]$ with respect to $u(k)$ around $u(k-1)$ can result in

$$\begin{aligned} y(k+d) &= f\left[w_k, u(k)\right] + \xi(k) \\ &= f\left[w_k, u(k-1)\right] + f_{1k}\Delta u(k) + R + \xi(k) \end{aligned} \tag{5}$$

where , $f_{1k} = (\partial f[w_k, u(k)]/\partial u(k))|_{u(k)=u(k-1)} \Delta u(k) = u(k) - u(k-1)$.

The remainder $R = f_{2k}[w_k, \zeta]\Delta u^2(k)/2$ is bounded by

$$|R| \leq \frac{r_1 \Delta u^2(k)}{2} \tag{6}$$

where $f_{2k}[w_k, \zeta] = (\partial^2 f[w_k, u(k)]/\partial u^2(k))|_{u(k)=\zeta}$, ζ is a point between $u(k)$ and $u(k-1)$, and $0 \leq |f_{2\mathrm{k}}[w_k, \zeta]| \leq r_1$ with r_1 being a finite positive number.

The NN I/O approximate model for SISO case is derived by neglecting the remainder R and the approximation error $\xi(k)$ as follows:

$$\mathbf{NSM}: \hat{y}(k+d) = f[w_k, u(k-1)] + f_{1\mathrm{k}}\Delta u(k). \tag{7}$$

Using (7), the NN inverse control law can be determined directly since the increment $\Delta u(k)$ of the control signal appears linearly. The model (7) is an approximation of the system (5) described by an NN. The magnitude of the total approximation error depends on the magnitude of the remainder $R(= f_{2k}[w_k, \zeta]\Delta u^2(k)/2)$ and the magnitude of the NN approximation error $\xi(k)$. The error $\xi(k)$ can be made arbitrarily small by increasing the layers and neurons of the neural network. Thus, according to (6), $|\Delta u(k)|$ should not be too large in order to limit the approximation error of the model (7) for a computed $u(k)$. In practice, this is reasonable because the output of a physical system (actuator) cannot change too fast within a small time interval due to the "inertia" of the system [21].

MIMO Case: A neural network $F[.]$ can be used to approximate the MIMO NARMA model (3) as

$$y(k+d) = F[v_k, u(k)] + \xi_p(k) \tag{8}$$

where $\xi_p(k)$ is an approximation error vector, $\|\xi_p(k)\| \leq \bar{\xi}_p$ with $\bar{\xi}_p$ as a small positive number and the weight vector of the NN is omitted for simplicity.

Similarly to SISO case, (8) can be equivalently expressed in (9)

$$y(k+d) = F[v_k, u(k-1)] + F_d\Delta u(k) + R_p + \xi_p(k) \tag{9}$$

where

$$\Delta u(k) = \begin{bmatrix} u_1(k) - u_1(k-1) \\ u_2(k) - u_2(k-1) \\ \cdots \\ u_p(k) - u_p(k-1) \end{bmatrix}$$

$$F_d = \left.\frac{\partial F[v_k, u(k)]}{\partial u(k)}\right|_{u(k)=u(k-1)}$$

$$F_{dd}[v_k, \zeta] = \left.\frac{\partial^2 F[v_k, u(k)]}{\partial u^2(k)}\right|_{u(k)=\zeta}$$

$$R_p = \frac{[\Delta u(k)]^{\mathrm{T}} F_{dd}[v_k, \zeta]\Delta u(k)}{2}$$

$\zeta = [\zeta_1\ \zeta_2,\ \ldots,\ \zeta_{\mathrm{p}}]^{\mathrm{T}}$ with ζ_j being a point between $u_j(k)$ and $u_j(k-1)$.

Letting $0 \leq \|F_{dd}[v_k, \zeta]\| \leq r_p$, thus

$$\|R_p\| \leq \frac{r_p \|\Delta u(k)\|^2}{2} \tag{10}$$

where r_p is a finite positive number.

Then, an NN input-output approximation model can be derived from (9) for MIMO case by neglecting R_p and $\xi_p(k)$ as follows:

$$\mathbf{NMM}: \hat{y}(k+d) = F[v_k, u(k-1)] + F_d\Delta u(k). \tag{11}$$

B. Control Development and Stability Proof

SISO Case: Based on the NN input-output approximation model derived in Section III-A, the NN control law can be determined straightforwardly from the NN approximation model (7) for SISO system as follows:

$$u(k) = u(k-1) + \Delta u(k) \tag{12}$$

where

$$\Delta u(k) = \frac{r(k+d) - f[w_k, u(k-1)]}{f_{1k}^2 + \alpha} f_{1k}, \quad \text{for } |\Delta u(k)| \leq \delta$$
$$\Delta u(k) = \delta \operatorname{sign}(\Delta u(k)), \quad \text{for } |\Delta u(k)| > \delta$$

with α and δ as given finite positive numbers.

Define control error as $e(k) = r(k+d-1) - y(k+d-1)$, thus

$$\begin{aligned} e(k+1) &= r(k+d) - y(k+d) \\ &= r(k+d) - f[w_k, u(k-1)] \\ &\quad - f_{1\mathrm{k}}\Delta u(k) - R - \xi(k). \end{aligned} \tag{13}$$

The robustness of the stability and the performance for NN control law (12) are given in Theorem 1.

Theorem 1: (SISO Case): For given $|r(k) - r(k-1)| \leq \Delta r$, using the NN control law (12), the solution of error system (13) is uniformly ultimately bounded (UUB) [20] for all k with ultimate bound $\lim_{k\to\infty} |e(k)| \leq (k_2/(1-k_1))$.

where $k_1 = (1 - s_0(k) + s_0(k)\alpha/(\beta^2 + \alpha))\mu_0(k)$, $k_2 = k_1\Delta r + r_1(\delta^2/2) + \bar{\xi}$

Δr is a given positive number;
$0 < s_0(k) \leq 1$;
$0 \leq \beta \leq |f_{1\mathrm{k}}|$; $0 \leq \mu_0(k) < 1$;
$\bar{\xi}$ and r_1 are defined in (4) and (6), respectively.

Proof: Define a variable $s_0(k)$ where $0 < s_0(k) \leq 1$ for all k. The control law (12) is equivalently expressed as

$$\Delta u(k) = \frac{r(k+d) - f[w_k, u(k-1)]}{f_{1k}^2 + \alpha} s_0(k) f_{1k} \tag{14}$$

where

$$s_0(k) = 1, \quad \text{if } |\Delta u(k)| \leq \delta \text{ and}$$
$$0 < s_0(k) < 1, \quad \text{if } |\Delta u(k)| > \delta.$$

Using (14), one has

$$\begin{aligned} r(k+d) &- \hat{y}(k+d) \\ &= r(k+d) - f[w_k, u(k-1)] - f_{1\mathrm{k}}\Delta u(k) \\ &= r(k+d) - f[w_k, u(k-1)] \\ &\quad - \frac{r(k+d) - f[w_k, u(k-1)]}{f_{1k}^2 + \alpha} s_0(k) f_{1k}^2 \\ &= \left(1 - s_0(k) + \frac{s_0(k)\alpha}{f_{1k}^2 + \alpha}\right) \\ &\quad \times (r(k+d) - f[w_k, u(k-1)]). \end{aligned} \tag{15}$$

Thus, (13) can be rewritten as

$$\begin{aligned} e(k+1) &= r(k+d) - y(k+d) \\ &= r(k+d) - \hat{y}(k+d) - R - \xi(k) \\ &= \left(1 - s_0(k) + \frac{s_0(k)\alpha}{f_{1k}^2 + \alpha}\right) \\ &\quad \times (r(k+d) - f[w_k, u(k-1)]) - R - \xi(k). \end{aligned} \tag{16}$$

Using the following fact (detailed derivation in Appendix A):

$$|r(k+d) - f[w_k, u(k-1)]| < |r(k+d) - y(k+d-1)| \tag{17}$$

we obtain

$$|r(k+d) - f[w_k, u(k-1)]| = \mu_0(k)\,|r(k+d) - y(k+d-1)| \tag{18}$$

with $0 \leq \mu_0(k) < 1$ for all k.

Using (18), (16) becomes

$$\begin{aligned}|e(k+1)| &\le \left(1-s_0(k)+\frac{s_0(k)\alpha}{f_{1k}^2+\alpha}\right)\\ &\quad\times |r(k+d)-f\left[w_k,u(k-1)\right]|+|R|+|\xi(k)|\\ &= \left(1-s_0(k)+\frac{s_0(k)\alpha}{f_{1k}^2+\alpha}\right)\mu_0(k)\\ &\quad\times |r(k+d)-y(k+d-1)|+|R|+|\xi(k)|\\ &\le \left(1-s_0(k)+\frac{s_0(k)\alpha}{\beta^2+\alpha}\right)\mu_0(k)\\ &\quad\times |r(k+d)-r(k+d-1)+e(k)|+|R|+|\xi(k)|\\ &\le \left(1-s_0(k)+\frac{s_0(k)\alpha}{\beta^2+\alpha}\right)\mu_0(k)\,|e(k)|\\ &\quad+\left(1-s_0(k)+\frac{s_0(k)\alpha}{\beta^2+\alpha}\right)\mu_0(k)\Delta r+r_1\frac{\delta^2}{2}+\bar{\xi}\\ &= k_1\,|e(k)|+k_2. \end{aligned}\tag{19}$$

Choosing a Lyapunov function as $V_k = |e(k)|$, from (19), one has

$$V_{k+1} - V_k = |e(k+1)| - |e(k)| = -(1-k_1)V_k + k_2.$$

Since $0 \le k_1 < 1$ and k_2 is bounded, according to the lemma 13.1 in [20], one concludes that, using the NN control law (12), the solutions of error system (13) are UUB for all k with ultimate bound $\lim_{k\to\infty} |e(k)| \le (k_2/(1-k_1))$. #

Remark 1: The inverse control directly from the approximate model NSM (7) requires $f_{1k} \ne 0$ to get the control incremental $\Delta u(k)$. It is worth noting that if the relative degree of the original system is not well defined, even if the system is invertible, the control law cannot be determined directly from (7), since f_{1k} might be 0 or sufficiently close to 0 at some points, which is one of the similar problems in the previous methods [4], [11], [13]. However, through a slight modification in the proposed method (12), i.e., choosing a finite positive number α, the singularity problem can be completely avoided.

In the neural control law (12), two parameters, α and δ, need to be chosen. It can be easily seen from (19) that both two parameters affect the transient response and steady state performance. A simple method to tune these parameters is summarized as follows.

- Initially, a small α (0.25 is used for our case) is given with δ left unbounded. Usually, the control performance is good if the system is of well-defined relative degree. Better control performance may be achieved by increasing or decreasing α.
- If the control performance is inferior for the initial given α, one may conclude that the system is of not well defined relative degree. In this case, decreasing α may lead to a better control performance. However, the response around the singular point may change abruptly due to the singularity in the control law. Thus, a finite value of δ is needed to limit control increment. By choosing a very small α (0.001 is used in our example) and varying δ, a satisfactory control performance can always be achieved.

MIMO Case: Based on the NN approximation model (11), the NN control law for MIMO case can be determined straightforwardly as follows:

$$u_j(k) = u_j(k-1) + \Delta u_j(k) \quad j = 1,2,\ldots,p \tag{20}$$

where

$$\begin{aligned}\Delta u_j(k) &= F_{dj}^T\left(F_dF_d^T+\alpha\right)^{-1}\\ &\qquad\times\left(r(k+d)-F\left[v_k,u(k-1)\right]\right), &&\text{when } |\Delta u_j(k)|\le\delta_j\\ \Delta u_j(k) &= \delta_j \operatorname{sign}\left[\Delta u_j(k)\right], &&\text{when } |\Delta u_j(k)|>\delta_j\end{aligned}$$

$$r(k+d) = \begin{bmatrix} r(k+d_1)\\ r(k+d_2)\\ \cdots\\ r(k+d_p)\end{bmatrix}$$

$$F[v_k,u(k-1)] = \begin{bmatrix} F_1[v_k,u(k-1)]\\ F_2[v_k,u(k-1)]\\ \cdots\\ F_p[v_k,u(k-1)]\end{bmatrix}$$

$$F_d = \begin{bmatrix} F_{d1}\\ F_{d2}\\ \cdots\\ F_{dp}\end{bmatrix} \text{ is defined in (9)}$$

$\alpha = \operatorname{diag}(\alpha_1,\alpha_2,\ldots,\alpha_p)$ with α_j as a given small positive number,

$\delta = [\delta_1\ \delta_2\ \cdots\ \delta_p]^T$ with δ_j as a given positive number.

Define control error $e(k) = r(k+d-1) - y(k+d-1)$, thus

$$\begin{aligned}e(k+1) &= r(k+d) - y(k+d)\\ &= r(k+d) - \hat{y}(k+d) - R_p - \xi_p(k)\\ &= r(k+d) - F\left[v_k,u(k-1)\right]\\ &\quad - F_d\Delta u(k) - R_p - \xi_p(k).\end{aligned}\tag{21}$$

Theorem 2: (MIMO Case): For given $|r_j(k) - r_j(k-1)| \le \Delta r_j$, $j = 1,2,\ldots,p$, using the NN control law (20), the solutions of error system (21) are UUB for all k with ultimate bound $\lim_{k\to\infty} \|e(k)\| \le (k_2/(1-k_1))$.

where $k_1 = (1 - \underline{s}(k) + \underline{s}(k)\|\alpha(\underline{\lambda}I_p+\alpha)^{-1}\|)\mu(k)$ and $k_2 = k_1\|\Delta r\| + r_p\|\delta\|^2/2 + \bar{\xi}_p$;

$\Delta r = [\Delta r_1\ \Delta r_2\ \cdots\ \Delta r_p]^T$ with Δr_j as a given positive number

$0 < s_j(k) \le 1$;

$\underline{s}(k) = \min\{s_1(k), s_2(k),\ldots,s_p(k)\}$;

$0 \le \mu(k) < 1$; $\bar{\xi}_p$ and r_p are defined in (8) and (10), respectively,

$\underline{\lambda}$ is the minimum eigenvalue of $F_dF_d^T$.

Proof: The control law (20) is equivalent to the following form:

$$\Delta u(k) = s(k)F_d^T\left(F_dF_d^T+\alpha I_p\right)^{-1}F_r \tag{22}$$

where $s(k) = \operatorname{diag}[s_1(k), s_2(k),\ldots,s_p(k)]$ and

$$\begin{aligned}&s_j(k) = 1, &&\text{for } |\Delta u_j(k)| \le \delta_j\\ &0 \le s_j(k) < 1, &&\text{for} |\Delta u_j(k)| > \delta_j.\\ &F_r = r(k+d) - F\left[v_k,u(k-1)\right].\end{aligned}$$

Using (22), (21) becomes

$$\begin{aligned}e(k+1) &= F_r - F_ds(k)F_d^T\left(F_dF_d^T+\alpha\right)^{-1}F_r - R_p - \xi_p(k)\\ &= \left(F_d\left(I_p-s(k)\right)F_d^T+\alpha\right)\left(F_dF_d^T+\alpha\right)^{-1}F_r\\ &\quad - R_p - \xi_p(k).\end{aligned}$$

Since $(F_d(I_p-s(k))F_d^T \le (1-\underline{s}(k))F_dF_d^T$ and $(F_dF_d^T+\alpha)^{-1} \le (\underline{\lambda}I_p+\alpha)^{-1}$, then

$$\begin{aligned}&\left(F_d\left(I_p-s(k)\right)F_d^T+\alpha\right)\left(F_dF_d^T+\alpha\right)^{-1}\\ &\le \left((1-\underline{s}(k))F_dF_d^T+\alpha\right)\left(F_dF_d^T+\alpha\right)^{-1}\\ &= (1-\underline{s}(k))\left(F_dF_d^T+\alpha\right)\left(F_dF_d^T+\alpha\right)^{-1}\\ &\quad+\underline{s}(k)\alpha\left(F_dF_d^T+\alpha\right)^{-1}\\ &\le (1-\underline{s}(k))I_p + \underline{s}(k)\alpha(\underline{\lambda}I_p+\alpha)^{-1}.\end{aligned}$$

Thus

$$\begin{aligned}\|e(k+1)\| \leq& \left\|\left(F_d\left(I_p - s(k)\right)F_d^T + \alpha\right)\left(F_dF_d^T + \alpha\right)^{-1}\right\| \\ &\times \|F_r\| + \|R_p\| + \|\xi_p(k)\| \\ \leq& \left\|\left(1 - \underline{s}(k)\right)I_p + \underline{s}(k)\alpha(\underline{\lambda}I_p + \alpha)^{-1}\right\| \|F_r\| \\ &+ \|R_p\| + \bar{\xi}_p. \end{aligned} \tag{23}$$

Using the following fact (detailed derivation in Appendix B:)

$$\begin{aligned}\|F_r\| &= \|r(k+d) - F\left[v_k, u(k-1)\right]\| \\ &< \|r(k+d) - y(k+d-1)\|\end{aligned} \tag{24}$$

one has

$$\|F_r\| = \mu(k)\,\|r(k+d) - y(k+d-1)\| \tag{25}$$

where $0 \leq \mu(k) < 1$ for all k.

Using (25), (23) becomes

$$\begin{aligned}\|e(k+1)\| \leq& \left(1 - \underline{s}(k) + \underline{s}(k)\left\|\alpha(\underline{\lambda}I_p + \alpha)^{-1}\right\|\right)\mu(k) \\ &\times \|(r(k+d) - y(k+d-1)\| + \|R_p\| + \bar{\xi}_p \\ \leq& \left(1 - \underline{s}(k) + \underline{s}(k)\left\|\alpha(\underline{\lambda}I_p + \alpha)^{-1}\right\|\right)\mu(k) \\ &\times (\|r(k+d) - r(k+d-1)\| \\ &+ \|r(k+d-1) - y(k+d-1)\|) + \|R_p\| + \bar{\xi}_p \\ \leq& \left(1 - \underline{s}(k) + \underline{s}(k)\left\|\alpha(\underline{\lambda}I_p + \alpha)^{-1}\right\|\right)\mu(k) \\ &\times (\|\Delta r\| + \|e(k)\|) + \|R_p\| + \bar{\xi}_p \\ =& k_1\,\|e(k)\| + k_2 \end{aligned} \tag{26}$$

Since $0 \leq k_1 < 1$ and k_2 is bounded, similarly to the proof of theorem 1, one concludes that, using the NN control law (20), the solutions of error system (21) are UUB for all k with ultimate bound $\lim_{k\to\infty}\|e(k)\| \leq (k_2/(1-k_1))$. #

C. Summary

The development of the proposed NN approximate inverse control is shown in Fig. 2 with key points below.

- An NN NARMA model is first developed using general identification techniques to approximate the original unknown nonlinear plant.
- From the identified NN NARMA model, a novel NN I/O approximation model is derived directly in which the increment of the control signal appears linearly.
- Then, the inverse control law is easily obtained from the NN approximate model.
- The computation work is small since no DG methods are involved and no further training is required for the neural nonlinear controller.

It is worth noting that the resulting controller can tolerate large system variations, i.e., we do not require that control signal $|u(k)|$ be small and $\Delta \mathrm{u(k)} \to 0$. However, the two conditions are required in [4] and [13], respectively. Recall that the widely used linear model is actually built by linearizing a nonlinear system around the origin (equilibrium point). Usually, the linearized model cannot tolerate large system variations (e.g. large input $u(k)$ and input increment $\Delta u(k)$) for the nonlinear system [4] because of too much approximation included in the linear model. However, the NN approximation model (7) can be considered as a dynamic linearization of NN NARMA model (4) around $u(k-1)$ so that less approximation is made on control development. Thus, large system variations can be tolerated by the proposed method. This result has been shown in the proof of the stability and performance of a closed-loop system and will be further confirmed in later simulations.

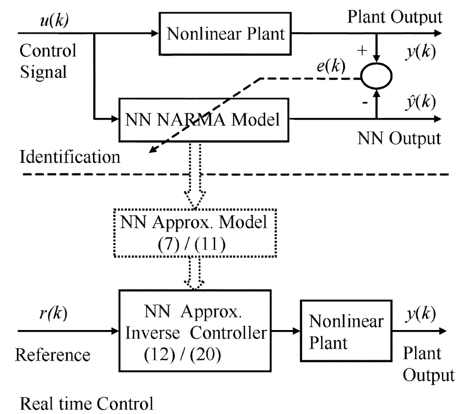

Fig. 2. Development of neural AIC.

Besides, it is interesting to note that the proposed AIC may provide some benefit to neural adaptive inverse control [6]. After model adaptation using NNs in adaptive inverse control, the proposed method can be used to derive an inverse controller based on the identified model. If the control performance is satisfactory, no further training is required and thus a lot of computation work is avoided.

IV. Simulations

When the process is unknown either in the SISO or the MIMO case, the neural network will be used for I/O identification. The developed neural AIC as indicated in Fig. 2 will be simulated here. The following common properties are employed in the simulations:

1) multilayer time delay neural network is used with structure $N^3_{i+1,11,11,m}$ (i.e., i inputs plus a bias, ten nodes plus a bias input in the first and second hidden layers, m output nodes and all bias inputs are unity);
2) $\tanh(.)$ function is chosen as the activation function for nonoutput neurons and the linear activation function is used for the output neurons;
3) Levenberg–Marquart method [22] is used in training, and the weight & bias initialization is made by the Nguyen-Widrow method [23]

They all are available in Neural Network toolbox in MATLAB.

A. Neural AIC of Unknown SISO Process

A first-order SISO system is described in the following difference equation [4]:

$$y(k+1) = \sin\left[y(k)\right] + u(k)\left(5 + \cos\left[(y(k)u(k)\right]\right). \tag{27}$$

The objectives of the simulations include the following:

1) to test the approximation performance of the NN input-output approximation model under the following test signals with different amplitudes.
   - Signal 1: $u_1(k) = 0.5\sin(2\pi k/50) + 0.5\sin(2\pi k/25)$ with $|u_1(k)| \leq 1$
   - Signal 2: $u_2(k) = 0.75\sin(2\pi k/50) + 0.75\sin(2\pi k/25)$ with $|u_2(k)| \leq 1.5$
2) to track the following two trajectories with different magnitudes using the proposed NN control law.
   - Trajectory 1: $r_1(k) = 2\sin(2\pi k/50) + 2\sin(2\pi k/100)$;
   - Trajectory 2: $r_2(k) = 2.75\sin(2\pi k/50) + 2.75\sin(2\pi k/100)$.

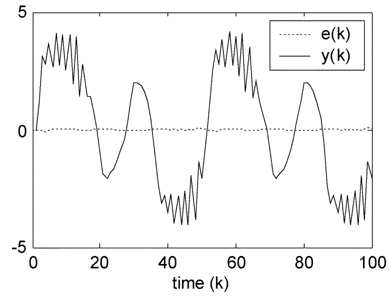

Fig. 3. Model test under input signal $u_1(k)$.

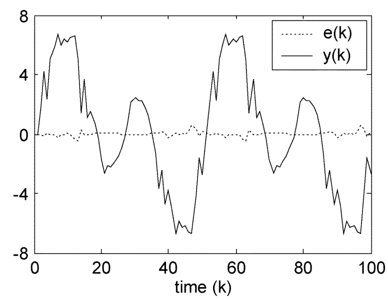

Fig. 4. Model test under input signal $u_2(k)$.

Step 1) Identification

- Neural structure: $N^3_{3,11,11,1}$;
- Training sequence: a random input sequence with 1000 values uniformly distributed in the interval [−1.5, 1.5] is used;
- Training error: mean square error (MSE) of 1.54×10^{-4} is obtained after 840 training epochs of the training sequences;
- NN NARMA model: $y(k+1) = f[y(k), u(k)]$
- NN Approx. model: (7) from the above identified NN NARMA model;

Step 2) Real time control

- NN control law: (12) from model NSM (7) with $\alpha = 0.25$ and $\delta = \infty$ ($\Delta u(k)$ is not bounded).

For the test signal 1, the output of NN approximation model (7) is very close to that of the system with quite small approximation error $e(k)$ as shown in Fig. 3. As the amplitude is increased in the test signal 2, the model NSM (7) still shows good approximation though some error $e(k)$ appears due to larger $|u(k)|$ and $|\Delta u(k)|$ as in Fig. 4.

The tracking errors of the SISO system under the proposed NN control law (12) are shown in Fig. 5. Since the magnitude of the second trajectory is larger than that of the first one, the required $|u(k)|$ and $|\Delta u(k)|$ for the second trajectory are larger than those for the first one. The proposed NN control method demonstrates very good control performance for both cases. The tracking error for trajectory 1 tracking (indicated by solid line in Fig. 5) is obtained as $|e(k)| < 0.08$. However, the larger error $|e(k)| < 0.3$ is found for trajectory 1 tracking using the previous NN control law [4].

B. Neural AIC of Unknown MIMO Process

Consider a MIMO nonlinear discrete system in [13] as follows:

$$\begin{aligned}
y_1(\mathrm{k}+1) =\ & 0.2\cos\left[2\left(y_1(k)+y_1(k-1)\right)\right.\\
&\left. +0.2\left(y_2(k)+y_2(k-1)\right)\right]\\
&+0.2\sin\left[0.2\left(y_1(k)+y_1(k-1)\right)+u_1(k)\right.\\
&\left. +u_1(k-1)+u_2(k-1)+0.1\right]\\
&+\frac{u_1(k)+u_1(k-1)+u_2(k-1)}{1+\cos\left(y_1(k)+0.2y_2(k)\right)}-0.2,\\
&\text{for } k>0 \text{ with } y_1(k)=0 \text{ for } k\le 0\\
y_2(k+1) =\ & 0.2\sin\left[0.2\left(y_1(k)+y_1(k-1)\right)\right.\\
&\left. +2\left(y_2(k)+y_2(k-1)\right)\right]\\
&+0.2\sin\left[0.2\left(y_2(k)+y_2(k-1)\right)+u_2(k)\right.\\
&\left. +u_2(k-1)+u_1(k-1)+0.1\right]\\
&+\frac{u_2(k)+u_2(k-1)+u_1(k-1)}{1+\sin\left(0.2y_1(k)+y_2(k)\right)},\\
&\text{for } k>0 \text{ with } y_2(k)=0 \text{ for } k\le 0
\end{aligned} \tag{28}$$

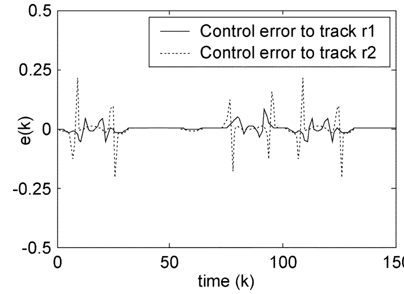

Fig. 5. Performance of NN AIC for the SISO plant.

The objectives of the simulations include the following:

1) to test the approximation performance of the NN I/O approximation model for MIMO nonlinear discrete systems under the following test inputs.
   Input 1: $u_1(k) = 0.125 + 0.375\sin(\pi k/50)$;
   Input 2: $u_2(k) = 0.4 + 0.475\cos(\pi k/50)$.
2) to track the following two sets of trajectories with different magnitudes using the proposed NN control law.
   Trajectory 1: $r_1(k) = 0.3+0.05[\sin(\pi k/50)+\sin(\pi k/100) + \sin(\pi k/150)]$, $r_2(k) = 0.6 + 0.1[\sin(\pi k/50) + \sin(\pi k/100)+ \sin(\pi k/150)]$.
   Trajectory 2: $r_1(k) = 0.3 + 0.2[\sin(\pi k/50) + \sin(\pi k/100) + \sin(\pi k/150)]$, $r_2(k) = 0.6 + 0.2[\sin(\pi k/50) + \sin(\pi k/100)+ \sin(\pi k/150)]$.

Step 1) Identification

- Neural structure: $N^3_{9,11,11,2}$;
- Training sequence: two input sequences each with 1000 random values uniformly distributed in the intervals [−0.4 0.6] and [−0.1 1];
- Training error: MSE of 1.82×10^{-6} is obtained after 210 training epochs of the training sequences.
- NN NARMA model: $y(k+1) = f[y(k), y(k-1), u(k), u(k-1)]$
- NN Approx. model: (11) from the above-identified NN NARMA model, where: $y(k) = [y_1(k)\ y_2(k)]^T$ and $u(k) = [u_1(k)\ u_2(k)]^T$.

Step 2) Real time control

- NN control law: (20) from model NMM (11) with $\alpha_1 = 0.5, \alpha_2 = 0.25$, and $\delta_1 = \delta_2 = \infty$ ($\Delta u_i(k)$ is not bounded).

The simulations in Fig. 6 show that the outputs ($y_1(k)$ and $y_2(k)$) of the NN approximation model (11) are very close to that of the system under the two test input signals, with very small approximation errors ($e_1(k)$ and $e_2(k)$). Similarly as in simulation IV-A, the required $|u(k)|$ and $|\Delta u(k)|$ for tracking the second trajectories are larger due to the increased magnitude. However, the proposed NN control method shows good tracking performance for both cases, and is not affected much by the increasing magnitude of the trajectory as shown in Fig. 7 and Fig. 8.

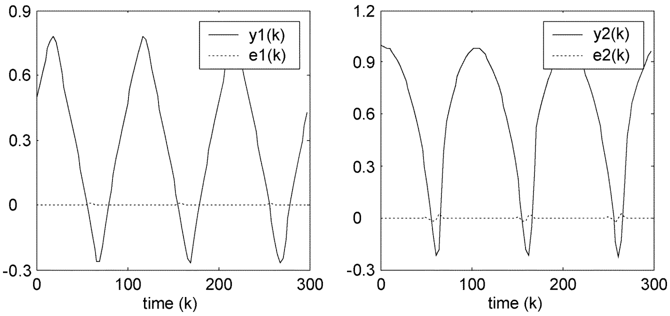

Fig. 6. Model test under input signals $u_1(k)$ and $u_2(k)$.

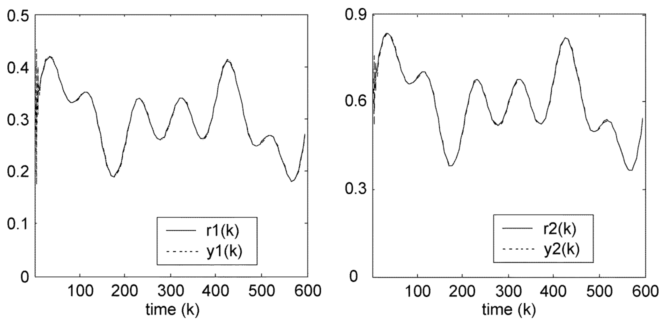

Fig. 7. Tracking performance of the first trajectories using NN AIC.

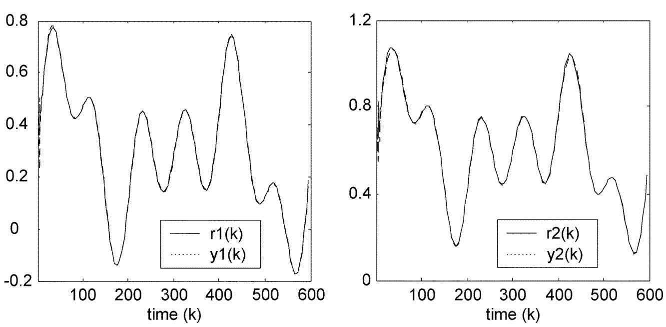

Fig. 8. Tracking performance of the second trajectories using NN AIC.

Comparing the control errors shown in Fig. 7 with those shown in [13], we can find that the proposed NN inverse control law gives much better control performance.

C. Neural AIC of a System With Not Well Defined Relative Degree

Consider an unknown SISO system with not well defined relative degree as described in the following difference equations:

$$\begin{aligned} x_1(k+1) &= \frac{0.25x_1(k)}{x_1^2(k)+x_2^2(k)+1} \\ &\quad - \frac{x_1^2(k)+x_2^2(k)}{x_1^2(k)+x_2^2(k)+1}\tanh\left(u^3(k)\right) \\ x_2(k+1) &= \frac{0.25x_2(k)}{x_1^2(k)+x_2^2(k)+1} + \tanh\left(u^3(k)\right) \\ y(k) &= x_1(k)+x_2(k). \end{aligned} \tag{29}$$

The objective of the simulation is to track a trajectory $r(k) = 0.3\sin(2\pi k/50)$ using the proposed NN control law.

The previous control methods [4], [11], [13] cannot be applied to this example since there is a singular point in their control laws. However it is feasible to apply the proposed NN approximate inverse control method, with critical design features shown below.

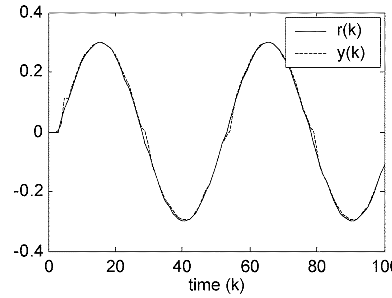

Fig. 9. Tracking control of a plant with not well defined relative degree.

Neural structure: $N^3_{5,11,11,1}$;
Training sequence: a random input sequence with 1000 values uniformly distributed in the interval [−1.5, 1.5] is used;
Training error: MSE of 2.16×10^{-5} is obtained after 660 training epochs of the training sequences.
NN NARMA model: $y(k+1) = f[y(k), y(k-1), u(k), u(k-1)]$
NN Approx. model: (7) from the above identified NN NARMA model.
NN control law: (12) from the NN approximation model (7) with $\delta = 0.26$ and $\alpha = 0.001$.

By using the proposed neural control law, the tracking performance is satisfactory, as shown in Fig. 9, even though the relative degree of the plant is not well defined.

V. Conclusions

For unknown nonlinear discrete systems, neural networks are widely used to model them and form an NN NARMA model. However, a good NARMA model in hand does not mean that a good controller can be easily obtained for general nonlinear discrete dynamical systems. With a novel I/O approximation proposed for the general NN NARMA model, a robust inverse control law is derived directly from the approximation and its implementation in neural network is straightforward without further training (online or offline) for the inverse controller. Only a general identification technique is involved for both SISO and MIMO systems and thus the design of the proposed neural inverse controller is simple. Since less approximation is made on model development than other methods, the proposed control is more accurate and robust because of larger tolerance of complexity. Extensive simulations demonstrate the robust and superior performance of the proposed NN approximate inverse control and its feasibility for most unknown nonlinear discrete dynamical systems with mild assumptions.

Appendix A
Derivation of Inequality (17)

The NN approximation model (7) is rewritten as follows:

$$\hat{y}(k+d) = f\left[w_k, u(k-1)\right] + f_{1k}\Delta u(k). \tag{A1}$$

The Taylor expansion of $f[w_k, u(k-1)]$ around w_{k-1} is as follows:

$$\begin{aligned} f\left[w_k, u(k-1)\right] &= f\left[w_{k-1}, u(k-1)\right] + R_1 \\ &= y(k+d-1) + R_1 - \xi(k-1). \end{aligned} \tag{A2}$$

The remainder $R_1 = f_w[\zeta, u(k-1)][w_k - w_k - 1]^T$ with $f_w[\zeta, u(k-1)] = (\partial f[w_k, u(k-1)]/\partial w_k)|_{w_k=\zeta}$ and ζ as a vector between w_k and w_{k-1}.

Since $0 < 1 - s_0(k) + (s_0(k)\alpha/(f_{1k}^2 + \alpha)) \leq 1$, from (15), one has $\mathrm{sign}[r(k+d) - \hat{y}(k+d)] = \mathrm{sign}[r(k+d) - f[w_k, u(k-1)]]$

and $|r(k+d) - \hat{y}(k+d)| \le |r(k+d) - f[w_k, u(k-1)]|$, which implies that

$$\hat{y}(k+d) \in [f[w_k, u(k-1)], r(k+d)). \tag{A3}$$

From (A1) and (A2), one has

$$\begin{aligned}\text{sign}\,&[f[w_k, u(k-1)] - y(k+d-1)]\\ &= \text{sign}\,[R_1 - \xi(k-1)]\\ &= \text{sign}\,[\hat{y}(k+d) - f_{1\text{k}}\Delta u(k) - y(k+d-1)]\end{aligned} \tag{A4}$$

and

$$\begin{aligned}0 &= |\hat{y}(k+d) - f_{1\text{k}}\Delta u(k) - f[w_k, u(k-1)]|\\ &\le |f[w_k, u(k-1)] - y(k+d-1)|.\end{aligned} \tag{A5}$$

Combining (A4) and (A5) yields

$$f[w_k, u(k-1)] \in (y(k+d-1), \hat{y}(k+d) - f_{1\text{k}}\Delta u(k)] \tag{A6}$$

for any $\Delta u(k)$.

Since the sign of $R_1 - \zeta(k-1)$ does not change when $\Delta u(k) \to 0$, from (A4) and (A5), one has

$$f[w_k, u(k-1)] \in (y(k+d-1), \hat{y}(k+d)] \tag{A7}$$

whenever $\Delta u(k) \to 0$.

On the other hand

$$\begin{aligned}0 &= |\hat{y}(k+d) - f_{1\text{k}}\Delta u(k) - f[w_k, u(k-1)]|\\ &\le |\hat{y}(k+d) - f[w_k, u(k-1)]|\end{aligned} \tag{A8}$$

for any $\Delta u(k)$.

Combining (A6), (A7), and (A8), one concludes that, for any $\Delta u(k)$

$$f[w_k, u(k-1)] \in (y(k+d-1), \hat{y}(k+d)]. \tag{A9}$$

Using (A3) and (A9) yields

$$f[w_k, u(k-1)] \in (y(k+d-1), r(k+d)).$$

Therefore, $|r(k+d) - f[w_k, u(k-1)]| < |r(k+d) - y(k+d-1)|$.

APPENDIX B
DERIVATION OF INEQUALITY (24)

Replacing (22) into the model NMM (11)

$$\begin{aligned}\hat{y}(k+d) = F[v_k, u(k-1)] + F_d s(k) F_d^T \left(F_d F_d^T + \alpha\right)^{-1}\\ \times (r(k+d) - F[v_k, u(k-1)]).\end{aligned} \tag{B1}$$

Since $\|F_d s(k) F_d^T (F_d F_d^T + a)^{-1}\| < 1$, from (B1), one has

$$\|\hat{y}(k+d) - F[v_k, u(k-1)]\| < \|(r(k+d) - F[v_k, u(k-1)])\|. \tag{B2}$$

On the other hand, according to model NMM (11)

$$\begin{aligned}r&(k+d) - \hat{y}(k+d)\\ &= r(k+d) - F[v_k, u(k-1)] - F_d\Delta u(k)\\ &= F_r - F_d s(k) F_d^T \left(F_d F_d^T + \alpha\right)^{-1} F_r\\ &= \left(F_d (I_p - s(k)) F_d^T + \alpha\right)\left(F_d F_d^T + \alpha\right)^{-1} F_r\end{aligned}$$

Since $\|(F_d(I_p - s(k))F_d^T + \alpha)(F_d F_d^T + \alpha)^{-1}\| \le 1$, one has

$$\|r(k+d) - \hat{y}(k+d)\| \le \|Fr\| = \|r(k+d) - F[v_k, u(k-1)]\| \tag{B3}$$

$r(k+d)$, $\hat{y}(k+d)$ and $F[v_k, u(k-1)]$ can be considered as three points in a p-dimensional space. Combining (B2) and (B3), one concludes that, $F[v_k, u(k-1)]$ is a point at somewhere between $r(k+d)$ and $\hat{y}(k+d)$ in the p-dimensional space under the constraints (B2) and (B3).

Similarly to the result (A9) in Appendix A, one can easily derive that

$$F_j[v_k, u(k-1)] \in (y_j(k+d-1), \hat{y}_j(k+d)] \tag{B4}$$

(B4) shows that $F[v_k, u(k-1)]$ is a point between $\hat{y}(k+d))$ and $y(k+d-1)$ at somewhere of the straight line connecting the points $\hat{y}(k+d))$ and $y(k+d-1)$ in the p-dimensional space. Thus, one has

$$\|\hat{y}(k+d) - F[v_k, u(k-1)]\| \le \|\hat{y}(k+d)) - y(k+d-1)\| \tag{B5}$$

and

$$\|F[v_k, u(k-1)] - y(k+d-1)\| \le \|\hat{y}(k+d)) - y(k+d-1)\|. \tag{B6}$$

Considering the space relations among the four points $r(k+d)$, $\hat{y}(k+d)$, $F[v_k, u(k-1)]$ and $y(k+d-1)$ in the p-dimensional space and combining (B2), (B3) and (B5), (B6) yields

$$\|r(k+d) - F[v_k, u(k-1)]\| < \|r(k+d)) - y(k+d-1)\|.$$

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers and the associate editor for their valuable comments.

REFERENCES

[1] C. Alippi, C. de Russis, and V Piuri, “A neural-network based control solution to air-fuel ratio control for automotive fuel-injection systems,” *IEEE Trans. Syst., Man, Cybern. C*, vol. 33, pp. 259–268, May 2003.
[2] J. B. D. Cabrera and K. S. Narendra, “Issues in the application of neural networks for tracking based on inverse control,” *IEEE Trans. Automat. Contr.*, vol. 44, pp. 2007–2027, Nov. 1999.
[3] S. Mukhopadhyay and K. S. Narendra, “Adaptive control of nonlinear multivariable systems using neural networks,” *Neural Netw.*, vol. 7, no. 5, pp. 737–752, 1994.
[4] K. S. Narendra and S. Mukhopadhyay, “Adaptive control using neural networks and approximate models,” *IEEE Trans. Neural Networks*, vol. 8, pp. 475–485, May 1997.
[5] I. Rivals and L. Personnaz, “Nonlinear internal model control using neural networks: application to processes with delay and design issues,” *IEEE Trans. Neural Networks*, vol. 11, pp. 80–90, Jan. 2000.
[6] G. L. Plett, “Adaptive inverse control of linear and nonlinear system using dynamic neural networks,” *IEEE Trans. Neural Networks*, vol. 14, pp. 360–376, Mar. 2003.
[7] B. Widrow and E. Walach, *Adaptive Inverse Control*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[8] K. S. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, Mar. 1990.
[9] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proc. IEEE*, vol. 78, pp. 1550–1560, Oct. 1990.
[10] R. J. Williams and D. Zipser, “Experimental analysis of the real-time recurrent learning algorithm,” *Connection Sci.*, vol. 1, no. 1, pp. 87–111, 1989.
[11] E. P. Nahas, M. A. Henson, and D. E. Seborg, “Nonlinear internal model control strategy for neural network models,” *Comput. Chem. Eng.*, vol. 16, no. 12, pp. 1039–1057, 1992.
[12] R. Boukezzoula, S. Galichet, and L. Foulloy, “Nonlinear internal model control: application of inverse model based fuzzy control,” *IEEE Trans. Fuzzy Systems*, vol. 11, pp. 814–829, Dec. 2003.
[13] O. Adetona, E. Garcia, and L. H. Keel, “A new method for the control of discrete nonlinear dynamic systems using neural networks,” *IEEE Trans. Neural Networks*, vol. 11, pp. 102–112, Jan. 2000.
[14] W. T. Miller, R. S. Sutton, and P. J. Werbos, *Neural Networks for Control*. Cambridge, MA: M.I.T. Press, 1992.

[15] L. Jin, P. N. Nikiforuk, and M. M. Gupta, "Fast neural learning and control of discrete-time nonlinear systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 478–488, Mar. 1995.

[16] G. Lightbody and G. W. Irwin, "Nonlinear control structures based on embedded neural system models," *IEEE Trans. Neural Networks*, vol. 8, pp. 553–567, May 1997.

[17] J.-Y. Choi, H. F. Van Landingham, and S. Bingulac, "A constructive approach for nonlinear system identification using multilayer perceptrons," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, pp. 307–312, Apr. 1996.

[18] K. Funahashi, "On the approximate realization of continuous mapping by neural network," *Neural Netw.*, vol. 2, pp. 183–192, 1989.

[19] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feed-forward networks are universal approximators," *Neural Netw.*, vol. 2, pp. 359–366, 1989.

[20] J. T. Spooner, M. Maggiore, R. Ordonez, and K. M. Passino, *Stable Adaptive Control and Estimation for Nonlinear Systems*. New York: Wiley, 2002.

[21] S. S. Ge, T. H. Lee, G. Y. Li, and J. Zhang, "Adaptive NN control for a class of discrete-time nonlinear systems," *Int. J. Control*, vol. 76, no. 4, pp. 334–354, 2003.

[22] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*. New York: McGraw-Hil, 1996.

[23] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *Proc. Int. Joint Conf. Neural Networks*, vol. 3, 1990, pp. 21–26.