International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:11, 2007

A Model-following Adaptive Controller for Linear/Nonlinear Plants using Radial Basis Function Neural Networks

Yuichi Masukake, and Yoshihisa Ishida

***Abstract*—**In this paper, we proposed a method to design a model-following adaptive controller for linear/nonlinear plants. Radial basis function neural networks (RBF-NNs), which are known for their stable learning capability and fast training, are used to identify linear/nonlinear plants. Simulation results show that the proposed method is effective in controlling both linear and nonlinear plants with disturbance in the plant input.

***Keywords*—**Linear/nonlinear plants, neural networks, radial basis function networks.

I. Introduction

IN the study of control systems, a linear operator using well-established techniques in linear algebra generally defines the system performance. However, the motion of real systems tends to be nonlinear by environmental changes.

Recently, a family of RBF-NNs [1] has been applied to adaptive control [2]-[4]. In many nonlinear control systems, the multi-layered perceptron neural networks (MLP-NNs) are commonly used for system identification. However, it is well known that the utility of MLP-NNs normally involves heavy computation, due to their long and iterative training of the weight vector. Moreover, they quite often suffer from numerical instability, due to the network parameters remaining at local minima during the training. Now, it is well known that the RBF-NNs are stable as compared with the conventional MLP-NNs and a number of non-iterative network parameter tuning paradigms have also been proposed.

In this paper, we propose a design method of the model-following adaptive controllers for linear and nonlinear plants using RBF-NNs. The organization of this paper is as follows: In Section 2, the problem formulation is described. RBF-NNs and the structure of the proposed control scheme are described in Section 3. In section 4, simulation results using both linear and nonlinear plants are given. Finally some conclusions are remarked in Section 5.

Manuscript received August 31, 2007.

The authors are with the School of Science and Technology, Meiji University, kanagawa, Japan, (corresponding author to provide phone: +81-44-934-7307; e-mail: ce77077@isc.meiji.ac.jp).

II. Problem Statements

Consider a single-input, single-output discrete system:

$$y_p(k+1) = f\left[y_p(k), y_p(k-1), \cdots, y_p(k-L+1)\right] + \sum_{j=0}^{L-1} b_j u(k-j), \quad (1)$$

where

$y_p(k)$: plant output,
$f[\bullet]$: unknown linear/nonlinear function,
$b_0 \neq 0$,
$u(k)$: plant input.

The purpose of this paper is to identify the linear/nonlinear function $f[\bullet]$ and the parameters $\{b_j\}$ of the linear term, and to implement a stable model-following adaptive controller.

III. Principle of Design

A. Radial Basis Function Network

Fig. 1 illustrates an RBF-NN with N_i inputs, N_h radial basis functions (RBFs), and a single output. In Fig. 1, the RBF-NN can be viewed as a single hidden-layer feed forward neural network.

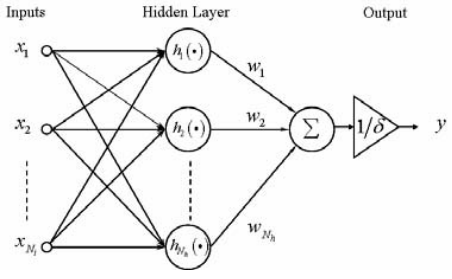

Fig. 1 Radial basis function neural network

Here $\boldsymbol{w} = \left[w_1, w_2, \cdots, w_{N_h} \right]^T$ is called as the weight vector. Each layer neuron computes the following Gaussian response function:

$$h_i = \exp\left(-\frac{\left\| \boldsymbol{x} - \boldsymbol{c}_i \right\|^2}{2\sigma^2} \right). \tag{2}$$

In the above equation, $\boldsymbol{x} = \left[x_1, x_2, \cdots, x_{N_i} \right]^T$ is the input vector, $\boldsymbol{c}_i$ denotes the centroid vector for the *i*-th RBF, $\left\| \cdots \right\|^2$ denotes the L_2 norm, and σ determines the radius.

The output neuron is then given as the linear weighted sum of the output values of RBFs, i.e.,

$$y = \frac{1}{\delta} \sum_{i=1}^{N_h} h_i w_i, \tag{3}$$

where δ is a constant, N_h is the number of RBFs.

B. Process of Design

We consider here the problem of controlling the plant described in Section 2 and the process of design. Let the difference equation of a plant be given as (1). Then, let us assume that the difference equation for the reference model is given by a second-order system,

$$\begin{aligned} y_m(k+1) = &-a_{m1} y_m(k) - a_{m2} y_m(k-1) \\ &+ b_{m0} r(k) + b_{m1} r(k-1), \end{aligned} \tag{4}$$

where

$y_m(k)$: model output,
$r(k)$: reference input.

The error $e(k)$ between the model output and the plant output is defined as

$$e(k) = y_m(k) - y_p(k). \tag{5}$$

Now, let us assume that at the steady state the following condition is satisfied:

$$e(k+1) = e(k). \tag{6}$$

From (6), the following equation is obtained:

$$y_m(k+1) - y_p(k+1) = y_m(k) - y_p(k). \tag{7}$$

By substituting (1) and (4) into (7) and replacing $y_m(k)$ with $y_p(k)$, we have

$$\begin{aligned} &-a_{m1} y_p(k) - a_{m2} y_p(k-1) + b_{m0} r(k) + b_{m1} r(k-1) \\ &\quad - f\left[k\right] - b_o u(k) - \sum_{j=1}^{L-1} b_j u(k-j) \\ &\quad = y_p(k) \\ &\quad - f\left[k-1\right] - b_o u(k-1) - \sum_{j=1}^{L-1} b_j u(k-j-1), \end{aligned} \tag{8}$$

where $f\left[k\right] = f\left[y_p(k), y_p(k-1), \cdots, y_p(k-L+1) \right]$.

Notice that the first term $y_m(k)$ on the right hand side in (7) is not replaced with (4). By so doing, the integral term $u(k-1)$ will be appeared in the plant input. Therefore, a new relation for the plant input $u(k)$ can be derived:

$$\begin{aligned} u(k) = &\, u(k-1) \\ &+ \frac{1}{b_0} \Big\{ -(a_{m1}+1) y_p(k) - a_{m2} y_p(k-1) \\ &+ b_{m0} r(k) + b_{m1} r(k-1) - f\left[k\right] + f\left[k-1\right] \\ &- \sum_{j=1}^{L-1} b_j \left[u(k-j) - u(k-j-1) \right] \Big\}. \end{aligned} \tag{9}$$

In the above, due to the appearance of $u(k-1)$, it is considered that the controllers so designed are not affected by the load disturbance. Although the proposed design method is extremely simple as compared with Narendra's method [5], the same result may be obtained. On the other hand, substituting (9) into (1), we obtain

$$\begin{aligned} y_p(k+1) = &-a_{m1} y_p(k) - a_{m2} y_p(k-1) \\ &+ b_{m0} r(k) + b_{m1} r(k-1). \end{aligned} \tag{10}$$

Then, the error $e(k+1)$ is given by

$$e(k+1) = -a_{m1} e(k) - a_{m2} e(k-1). \tag{11}$$

Therefore, the following condition is satisfied for arbitrary initial value $e(0)$:

$$\lim_{k \to \infty} e(k) = 0. \tag{12}$$

C. Design for Nonlinear Plants

In the system identification of plants, the multi-layer perceptron neural networks (MLP-NNs) with back-propagation algorithm [2]-[4] have been widely used. However, as discussed early, it is well known that employing the MLP-NNs involves rather heavy computation due to the iteratively updating of the weight vectors and the fact that there may always be a danger of the network parameters remaining at local minima. This limits their practical use, since an on-line processing is inevitable for controlling the dynamical systems. In this paper, we thus propose to use RBF-NNs for representing the plant within the system identification.

In the design, the function $f[\bullet]$ in (9) is replaced with the output of the RBF-NN and the input vector $\boldsymbol{x}(k)=\left[y_p(k), y_p(k-1), \cdots, y_p(k-L+1)\right]^T$ (i.e., $N_i = L$), namely

$$f[k]=f\left[\boldsymbol{x}^T(k)\right]=\frac{1}{\delta}\sum_{i=1}^{N_h} h_i(k)w_i, \tag{13}$$

with the response function

$$h_i(k)=\exp\left(-\frac{\|\boldsymbol{x}(k)-\boldsymbol{c}_i\|^2}{2\sigma^2}\right). \tag{14}$$

Then, the difference equation for plants yields

$$y_p(k+1)=\frac{1}{\delta}\sum_{i=1}^{N_h} h_i(k)w_i+\sum_{j=0}^{L-1} b_j u(k-j). \tag{15}$$

Finally, the plant input $u(k)$ is given by (9) with linear/nonlinear function (13).

D. Estimating Feedforward and Weight Parameters [6]

To estimate both the feedforward parameters $b_j\,(j=0,1,\cdots,L-1)$ and the weight parameters $w_i\,(i=1,2,\cdots,N_h)$ of the RBF-NN in (15), we apply the singular value decomposition (SVD) [7], instead of applying the conventional least squares.

E. Adaptive Control Algorithm

From (15),

$$\begin{aligned} y_p(k+1) &= \frac{1}{\delta}\sum_{i=1}^{N_h} h_i(k)w_i+\sum_{j=0}^{L-1} b_j u(k-j) \\ &= \left[h_1(k),\cdots h_{N_h}(k), u(k),\cdots,u(k-L+1)\right]\begin{bmatrix} \frac{w_1}{\delta} \\ \vdots \\ \frac{w_{N_h}}{\delta} \\ b_0 \\ \vdots \\ b_{L-1}\end{bmatrix} \\ &= \boldsymbol{\varphi}^T(k)\boldsymbol{\theta}. \end{aligned} \tag{16}$$

If the centroids $\{\boldsymbol{c}_i\}$ are known, we can obtain the unknown parameter vector $\boldsymbol{\theta}$, by applying the conventional recursive least-squares method to (16), as follows:

$$\begin{aligned} \hat{\boldsymbol{\theta}}(k) &= \hat{\boldsymbol{\theta}}(k-1)+\boldsymbol{\Gamma}(k)e(k) \\ e(k) &= y_p(k)-\boldsymbol{\varphi}^T(k-1)\hat{\boldsymbol{\theta}}(k-1) \\ \boldsymbol{\Gamma}(k) &= \frac{\boldsymbol{P}(k-1)\boldsymbol{\varphi}(k-1)}{\rho(k)+\boldsymbol{\varphi}^T(k-1)\boldsymbol{P}(k-1)\boldsymbol{\varphi}(k-1)} \\ \boldsymbol{P}(k) &= \frac{1}{\rho(k)}\left[\boldsymbol{P}(k-1)-\frac{\boldsymbol{P}(k-1)\boldsymbol{\varphi}(k-1)\boldsymbol{\varphi}^T(k-1)\boldsymbol{P}(k-1)}{\rho(k)+\boldsymbol{\varphi}^T(k-1)\boldsymbol{P}(k-1)\boldsymbol{\varphi}(k-1)}\right] \end{aligned} \tag{17}$$

On the other hand, if the centroids $\{\boldsymbol{c}_i\}$ are unknown, we apply the recursive *K*-means algorithm [8] as follows:

Step 1: Choose a set of centers $\{c_1, c_1, \cdots, c_{N_h}\}$ arbitrarily and give the initial learning rate $\gamma(0)=1$.

Step 2: Compute the minimum Euclidean distance

$$\left.\begin{aligned} l_i(k) &= \|\boldsymbol{x}(k)-\boldsymbol{c}_i(k-1)\| \quad (i=1,2,\cdots,N_h) \\ r &= \arg\left|\min l_i(k)\right| \end{aligned}\right\} \tag{18}$$

Step 3: Adjust the location of these centers as follows

$$\left.\begin{aligned} \boldsymbol{c}_i(k) &= \boldsymbol{c}_i(k-1)+\gamma(k)\left(\boldsymbol{x}(k)-\boldsymbol{c}_i(k-1)\right) \quad (i=r) \\ &= \boldsymbol{c}_i(k-1) \quad (i\neq r) \end{aligned}\right\} \tag{19}$$

Step 4: $k=k+1$, $\gamma(k)=0.998\cdot\gamma(k-1)$ and go to Step 2.

IV. Simulation Study

Consider a system described by

$$y_p(k+1)=f[\bullet]+b_0u(k)+b_1u(k-1), \tag{20}$$

where

$$\left.\begin{aligned} b_0 &= 0.350 \\ b_1 &= 0.076 \end{aligned}\right\}, \tag{21}$$

and,

$$f[\bullet]=\begin{cases} -a_1y_p(k)-a_2y_p(k-1)+\dfrac{y_p(k)}{1+y_p^2(k)}, \\ \quad \text{: in case of nonlinear plant} \\ -a_1y_p(k)-a_2y_p(k-1), \\ \quad \text{: in case of linear plant} \end{cases} \tag{22}$$

where

$$\left.\begin{aligned} a_1 &= -0.581 \\ a_2 &= 0.0067 \end{aligned}\right\}.$$

On the other hand, a reference model is given by

$$y_m(k+1) = -a_{m1}y_m(k) - a_{m2}y_m(k-1) + b_{m0}r(k) + b_{m1}r(k-1), \quad (23)$$

where

$$\left.\begin{aligned} a_{m1} &= -1.213 \\ a_{m2} &= 0.368 \\ b_{m0} &= 0.090 \\ b_{m1} &= 0.065 \end{aligned}\right\}. \quad (24)$$

In simulation experiments, the sampling time is 0.5[s], the initial values $a_1 = 1$, $a_2 = 1$, the number of RBFs N_h is 10 and the other parameters are set arbitrarily. We also assumed that at the sample number 80 a step-wise disturbance with the magnitude 0.1 is added to the plant input $u(k)$. Simulation examples are limited to second-order plants in order to simplify the problem.

A. Linear Plant

For the linear plant, the simulation result is shown in Fig. 2. In Fig. 2, the solid and dotted lines represent model and plant outputs, respectively.

B. Nonlinear Plant

For the nonlinear plant, the simulation result is shown in Fig. 3. Simulation condition is the same as the linear plant.

V. Conclusion

In this paper, we have proposed a new model-following adaptive controller for both linear and nonlinear plants. In the proposed method, RBF-NNs have been used for the system identification of linear and nonlinear plants. Simulation results have revealed that the proposed control scheme is suitable for both linear and nonlinear plants. Moreover, the proposed method control method is extremely simple as compared with Narendra's method.

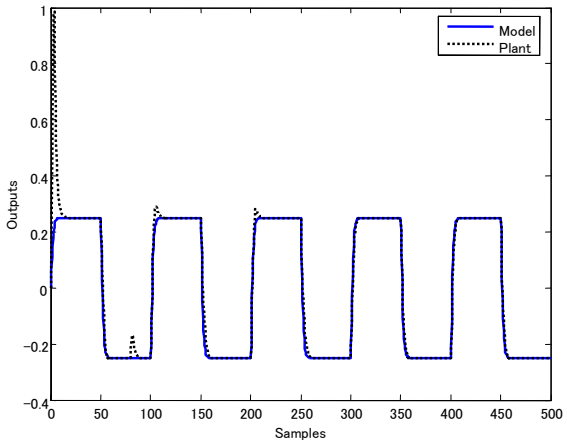

Fig. 2 Simulation result for the linear plant

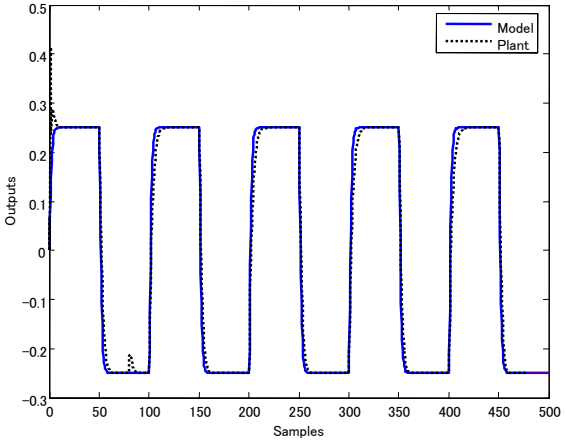

Fig. 3 Simulation result for the nonlinear plant.

Acknowledgement

This research was supported in part by a research grant from SMC Co., Ltd, Japan.

References

[1] P. D. Wasserman, *Advanced Method in Neural Networks* (Book Style), Van Nostrand Reinhold, New York, 1993, pp. 147-176.
[2] R. M. Sanner and J. E. Slotine, "Gaussian Networks For Direct Adaptive Control," IEEE Trans. Neural Networks, Vol. 3, No. 6, pp.837-863, 1992.
[3] R. M. Sanner and J. E. Slotine, "A Stable Adaptive Control Of Robot Manipulators Using Neural Networks," Neural Computation, Vol. 7, pp.753-790, 1995.
[4] D. Sbarbaro, J. P. Segovia, S. Alcozer and J. Gonzales, "Applications Of Radial basis Network Technology To Process Control," IEEE Trans. Control System Technology, Vol. 8, No. 1, pp14-22, 2000.
[5] S. Mukhopadhyay and K. S. Narendra, "Disturbance rejection in nonlinear systems using neural networks," IEEE trans. Neural Networks, vol. 4, no.1, pp. 63-72, 1993.
[6] Y. Ishikawa, Y. Masukake and Y. Ishida, "Control of Chaotic Dynamical Systems using RBF Networks," ENFORMATIKA, Vol. 19, pp. 175-178, 2007.
[7] G. H. Goulb and C. F. Van-Loan, *Matrix Computation* (Book Style), 3rd Edition, The Johns Hopkins Univ. Press, Baltimore and London, 1996.
[8] T. Hachino, K. Hasuka and H. Takata, "On-line Identification of Continuous-time Nonlinear System via RBF Network Model," SICE Conference in Kyusyu, pp. 215-216, 2001(in Japanese).

Yuichi Masukake was born in Chiba, Japan, on October 26, 1983. He received the B. E. degree in Electronics and Communications from Meiji University, Kawasaki, Japan, in 2007. He is currently working toward the M. E. degree at Graduate School of Electrical Engineering, Meiji University. His research interests include digital signal processing and its applications.

Yoshihisa Ishida was born in Tokyo, Japan, on February 24, 1947. He received the B. E., M. E., and Dr. Eng. Degrees in Electrical Engineering, Meiji University, Kawasaki, Japan, in 1970, 1972, and 1978., respectively. In Meiji University, as a Research Assistant and became a Lecturer and an Associate Professor in 1978 and 1981, respectively. He is currently a Professor at the Department of Electronics and Communications, Meiji University. His current research interests include signal processing, speech analysis and recognition, and digital control. He is a member of the IEEE, and the IEICE of Japan.