⊙ 研究論文

Intelligent Predictive Control of Time – Varying Dynamic Systems with Unknown Structures Using Neural Networks

S. J. Oh*

신경회로망에 의한 미지의 구조를 가진 時變動的시스템의 知能的 豫測制御

오 세 준

Key words : Intelligent, Predictive control, Time – varying, Dynamic system, Neural network, Feedforward control, Inverse dynamics

Abstract

A neural predictive tracking system for the control of structure-unknown dynamic systems is presented. The control system comprises a neural network modelling mechanism for the forward and inverse dynamics of a plant to be controlled, a feedforward controller, a feedback controller, and an error prediction mechanism. The feedforward controller, a neural network model of the inverse dynamics, generates feedforward control signal to the plant. The feedback control signal is produced by the error prediction mechanism. The error predictor adopts the neural network models of the forward and inverse dynamics. Simulation results are presented to demonstrate the applicability of the proposed scheme to predictive tracking control problems.

Ⅰ. INTRODUCTION

In conventional control theory, most of the work on the identification and control of dynamic systems has been based on the assumption ; 1) the plant to be controlled is linear 2) the plant model has a known form, but unknown parameters. The parameter estimation process in the approaches causes a growing computational complexity with the number of unknown parameters[1]. Because of this problem, it is difficult to obtain a practical control system to achieve high performance in the control of unknown dynamic systems.

Recently, there has been much interest in applying neural networks to the identification

* Member of KSME, Korea Maritime University

and control of dynamic systems. This is because neural networks can easily be applied to model the plant forward dynamics and inverse dynamics without *a priori* knowledge of the plant and these neural network models can be immediately used for control problems[2,3].

A number of approaches have been proposed and used in many applications over the years [4-5]. The work of Narendra et al.[4] has shown that it is feasible to model the forward dynamics of general linear and nonlinear plants (including multivariable systems) by using the so called tapped delay neural networks (TDNN). Although some theoretical questions, such as optimum network size and local minima still remain, neural networks have great promise in modelling nonlinear systems[5].

On the other hand, neural networks have been employed to identify or extract inverse dynamics models of unknown plants through learning[6,7]. There are two main approaches, i.e. direct and indirect inverse learnings, to the neural - network - based identification of the inverse dynamics of plants[8]. If a perfect inverse dynamics model is available, the controller could simply be made equal to the inverse dynamics of the plant to be controlled. This idea has been adopted to the control problem for a robot manipulator to determine suitable control inputs for each joint, so that the manipulator can execute a certain commanded motion[9,10]. In the approach of 9), the controller utilised the CMAC network as the robot dynamics model.

This paper presents a predictive tracking control scheme by using neural network[2] for dynamic systems with unknown structures. The scheme is based on a feedforward and feedback controllers for controlling unknown plants without a priori knowledge of their dynamics. The proposed scheme is described in Section Ⅱ and the simulation results obtained are given in Section Ⅲ.

Ⅱ. CONTROL ARCHITECTURE

1. Modelling Forward and Inverse Dynamics

The forward and inverse dynamics models are realised by neural networks which are trained off - line. In the training procedure, the neural network Ψ for the forward dynamics model of the plant to be controlled is positioned across the plant, taking as input the plant input (control signal). The neural network Φ for the inverse dynamics model of the plant is trained with direct inverse learning scheme, i.e. the network is placed across the plant and takes as input the actual plant output. It is assumed that Ψ and Φ are sufficiently complex to be able to approximate any input - output and output - input mappings of dynamic systems, respectively. The sequential type recurrent backpropagation networks described in 2,6) are used to implement Ψ and Φ.

The pattern training method[2] has been employed whereby the weight adjustment process is carried out after presentation of each input - output data pair (training pattern) to the network. The training stops when the mean squared error has fallen below a preset threshold. In training the inverse network, the network learns to produce the approximation of $z^{-d}\mathbf{u}$ (with plant input $\mathbf{u}$) by adjusting its trainable weights based on the gradient of the error between $z^{-d}\mathbf{u}$ and the network output with respect to these weights, so as to drive the error to zero. The delay term d is generally chosen to be between 1 and 3 for systems without time delays[3]. Figure1 shows the neural network architecture used for the forward and

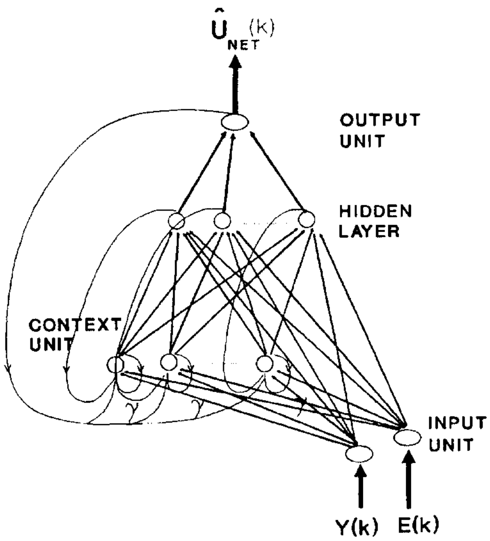

Fig. 1 Architecture of neural network used

inverse dynamics modellings in this work[2].

2. Feedback Error Prediction Mechanism

A feedback error prediction mechanism(predictor) is used to achieve an advanced control performance by incorporating in the control law. The error predictor utilises two neural networks representing the forward and inverse dynamics of the plant as depicted in Figure 2. The forward dynamics modelling network Ψ and the inverse dynamics modelling network Φ are obtained from off-line training, respectively.

The current predicted feedback error $\hat{E}(k+1)$ is computed from a forward and inverse models of the plant, as follows :

$$\hat{y}(k+1)=\Psi\{\Phi\{y_d(k+1), \mathbf{y}_d, \mathbf{u}_f\}+K \cdot E(k)\} \tag{1}$$

$$\Delta\hat{E}(k+1)=y_d(k+1)-\hat{y}(k+1) \tag{2}$$

$$\hat{E}(k+1)=\{E(k)+\Delta\hat{E}(k+1)\} \tag{3}$$

where K is the proportional feedback gain chosen to satisfy control specifications, $\hat{y}(k+1)$ is the predicted output of the plant, $y_d(k+1)$ is the next desired output, $\mathbf{y}_d=[y_d(k)\cdots y_d(k-n_a+1)]^T$, $\mathbf{u}_f=[u_f(k-1)\cdots u_f(k-n_b+1)]^T$, and n_a, n_b represent the maximum dynamic lags in the reference input and the feedforward control input, respectively, k is the time instant.

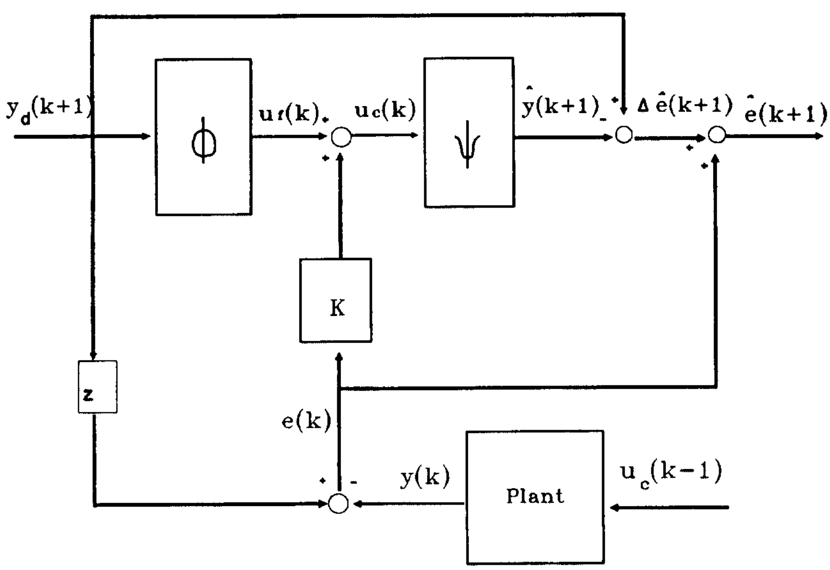

Fig. 2 Block diagram of error prediction mechanism

3. Control Architecture

The control scheme adopted in this work is composed of a feedforward inverse controller and a feedback controller. The feedforward inverse controller is a neural network the dynamics of which is the inverse dynamics of the plant to be controlled. This control network, Φ^c, is a direct copy of the network, Φ, modelling the inverse dynamics of the controlled plant through training (training network). The control network Φ^c is placed in series with the controlled plant. The feedforward control signal is generated by Φ^c, taking as inputs the desired output (reference). The feedback control signal is produced by a proportional feedback controller through the feedback error predictor to compensate for discrepancies between the desired and actual plant outputs.

Consider a single - input - single - output dynamic plant which can be represented in the discrete input - output format as

$$y(k+1)=h\{y(k),\cdots y(k-n_a+1),\ u(k),\ u(k-1),\ \cdots,\ u(k-n_b+1)\} \quad (4)$$

where y and u are the plant output and input, and h is a function mapping the present and past inputs to the plant and the present and past outputs from the plant to the next output of the plant.

By assuming that there exists a function $g(\cdot)$ such that equation (4) is invertible, an inverse description for the plant can be given as follows:

$$u(k)=g\{y(k+1),\ y(k),\cdots,\ y(k-n_a+1),\ u(k-1),\ \ldots,\ u(k-n_b+1)\} \quad (5)$$

Equation (5) can be implemented by a controller, $\Phi(\cdot)$, with input vector $\mathbf{x}(k)$, where

$$\mathbf{x}(k)\equiv[y_d(k+1)\ \mathbf{y}\ \mathbf{u}]^T \quad (6)$$

where $y=[y(k)\ \cdots\ y(k-n_a+1)]^T$ is the vector of present and past outputs (output dynamic memory), and $\mathbf{u}=[u(k-1)\ \cdots\ u(k-n_b+1)]^T$ is the vector of the input dynamic memory.

The controller output u(k) and the plant output y(k+1) are, respectively,

$$\begin{aligned} u(k)&=\Phi\{\mathbf{x}(k)\} \\ &=u_d{}^*(k) \\ y(k+1)&=h\{\mathbf{y},\ \mathbf{u},\ \Phi\{\mathbf{x}(k)\}\} && (7) \\ &=y_d(k+1) && (8) \end{aligned}$$

In equation (7), $u_d{}^*(k)$ represents the controller output needed to produce the output of the plant. Thus, if the output of $\Phi(\cdot)$ approximates sufficiently accurately that of $g(\cdot)$ for a corresponding input, the plant output $y(k+1)$ will equate the desired output, $y_d(k+1)$.

According to equations (7) - (8), the actual plant output y can be arbitrarily controlled to any desired output y_d if a complete inverse mapping is available. However, in practice, when one considers Φ to be the inverse model of the plant, the identity between equations (5) and (7) is generally very difficult to realise due to noise, modelling uncertainty, and changes in the environment.

To control the plant with a practical inverse controller, a control law is defined as follows :

$$\begin{aligned} u_c(k)&=u_f(k)+u_b(k) \\ &=\Phi^c\{y_d(k+1),\ \mathbf{y}_d,\ \mathbf{u}\}+K\cdot\hat{E}(k+1) \quad (9) \end{aligned}$$

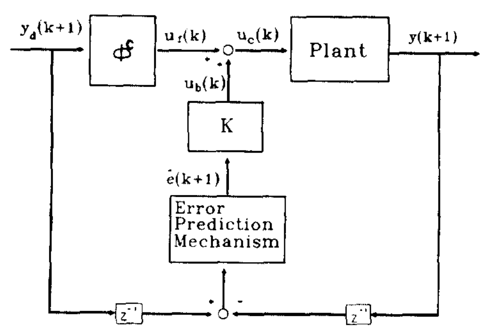

Fig. 3 Schematic configuration of control system

where u_c represents the actual control input, u_f and u_b are the feedforward and feedback control signals. The schematic configuration of the overall control system is shown in Figure 3.

Ⅲ. SIMULATION RESULTS

In this Section, experimental results for an example plant are given to illustrate the modelling validation and the performance of the proposed control scheme.

An unknown plant[11] represented by the following input - output equation was to be controlled :

$$y(k+1)=0.68y(k)+0.22\,y(k-1)+0.26u(k)+0.08u(k-1) \qquad (10)$$

For training the forward dynamics modelling network, a uniformly distributed random input signal was used. In the inverse dynamics modelling case, a sinusoidal signal $u(k)=0.05\sin(2\pi k*0.1)+0.2\sin(2\pi k*0.03)+0.3\sin(2\pi k*0.01)$ was chosen. 400 training data were used for both the forward and inverse dynamics cases, and $d=1$ was selected for modelling inverse dynamics. The parameters for the networks are given in Table 1. In the simulation for control, $K=1.0$ and step reference input were used. Figure 4 shows the training results of the inverse dynamics modelling network with the sinusoidal input signal after 100,000 training iterations. In this case, the mean squared error (MSE defined as follows) for the training phase

Table 1 Training Parameters For Neural Networks

Parameter	η	μ	β	n	N	Hid. Layer Activation	Training Signal
Forward	0.0015	0.05	0.80	7	100,000	Linear	Uniformly Random
Inverse	0.0015	0.05	0.80	7	100,000	Linear	Sinusoid

η : learning rate, μ : momentum term, β : self feedback gain
n : mumber of hidden/state units, N : number of training iterations.

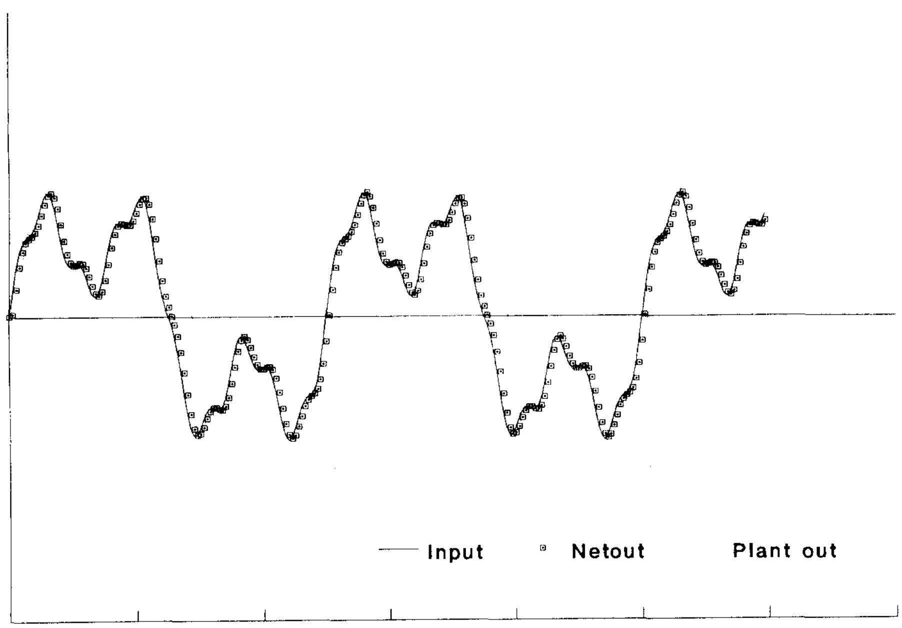

Fig. 4 Training results

became nearly zero. The recall results of the inverse modelling network are given in Figure 5 with different sinusoidal input signal u(k)= 0.1sin(2πk * 0.15)+0.15sin(2πk * 0.025)+ 0.25sin(2πk * 0.01) from the training phase. In the recall phase, 0.002 of the normalised mean squared error (NMSE) defined as follows was obtained.

$$MSE = \frac{1}{N}\sum_{k=1}^{N}\{u(k) - \hat{u}(k)\}^2 \tag{11}$$

$$NMSE = \frac{\frac{1}{N}\sum_{k=1}^{N}\{u(k) - \hat{u}(k)\}^2}{\frac{1}{N}\sum_{k=1}^{N}\{u(k)\}^2} \tag{12}$$

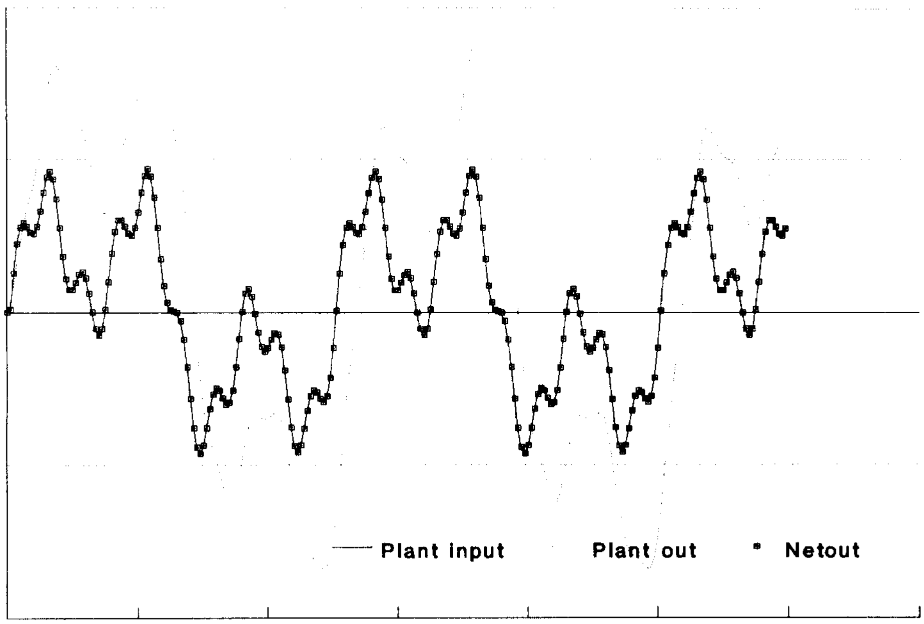

Fig. 5 Recall results

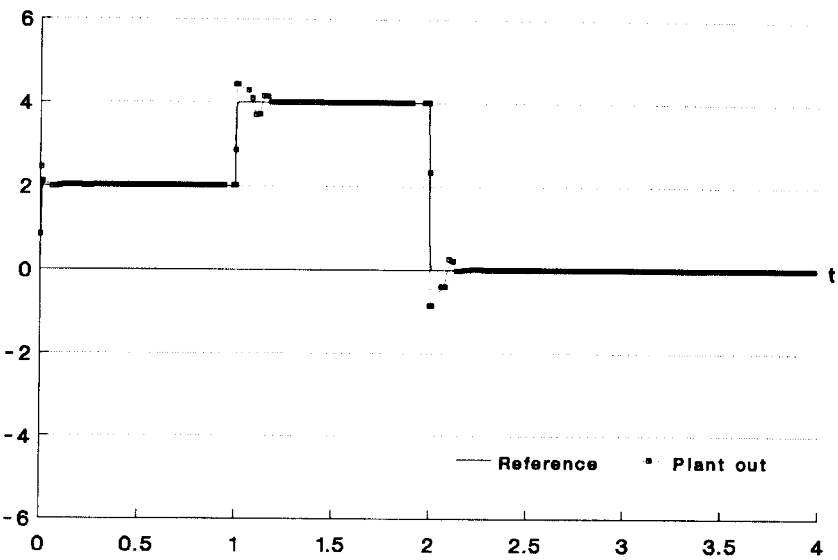

Fig. 6 Step response without predictor

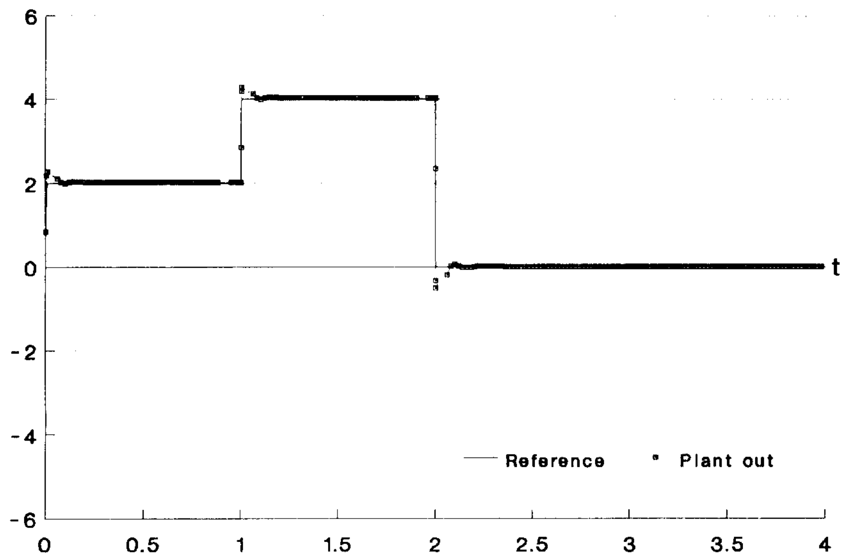

Fig. 7 Step response with predictor

Step responses for the plant were obtained with the control system shown in Figure 3. Figures 6 and 7 show the control response without and with the error predictor, respectively. As can be seen from the simulation results, the proposed control scheme is useful for such a tracking problem. The error prediction mechanism improved the control performance by maximum overshoot 6.5% (from 21.8% to 15.3%)

Ⅳ. CONCLUSION

This paper has presented a new tracking control scheme for dynamic systems with unknown structures by using neural networks. The approach is based on a feedforward neural inverse controller and a predicted error feedback controller. The scheme makes use of the forward and inverse dynamics models of the sequential recurrent network for predictive tracking problems. The feedback error prediction mechanism has improved the control performance. The proposed controller can easily be used in applications requiring fast and precise action without prior knowledge of the plant. The simulation results obtained for the example plant have shown the applicability of the proposed method to predictive tracking control.

ACKNOWLEDGEMENT

This paper was supported in part by NON DIRECTED RESEARCH FUND Korea Research Foundation.

REFERENCES

1) Landau, I. D. System identification and control design. Prentice - Hall, Eaglewood Cliffs, NJ, 1990
2) Pham, D. T., Oh, S. J. A recurrent backpropagation neural network for dynamic system identification. Journal of Systems Engineering, Vol.2, No.4, 1992, pp 213 - 223
3) Pham, D. T., Oh, S. J. Adaptive control of dynamic systems using neural networks. Proc. IEEE SMC Conference, Vol. 4, Le Touquet, France, Oct. 1993, pp. 97 - 102

4) Narendra, K. S., Parthasarathy, K. Identification and control of dynamical systems using neural networks. IEEE Trans. on Neural Networks, Vol.1, No.1, 1990, pp. 4 - 27

5) Hunt, K. J., Sbarbaro, D., Zbikowski, R., Gawthrop, P. J. Neural networks for control systems - A survey. Automatica, Vol. 28, No. 6, 1992, pp. 1083 - 1112

6) Oh, S. J. Inverse dynamics modelling by NNs for control systems. Proc. Symposium of KSME, Jeju, April 1994, pp. s1 - s6

7) Miller, W. T., Sutton, R. S., Werbos, P. J. Neural networks for control. MIT Press, Cambridge, MA, 1990

8) Psaltis,D., Sideris, A., Yamamura,A. A multilayered neural network controller. IEEE Control Systems Magazine, 1989, pp. 17 - 21

9) Miller, W. T., Glanz, F. H., Klaft, L. G. CMAC : Application of a general learning algorithm to the control of robotic manipulators. Int. J. Robotics Research, Vol. 6, No. 2, 1987, pp. 84 - 98

10) Pham, D. T., Oh, S. J. Control of a two - axis robot using neural networks. Proc. IMACS - IEEE/SMC Symposium on Signal Processing, Robotics and Neural Networks, Lille, France, April 25 - 27 1994

11) Yamamoto, T., Omatu, S., Ishihara, H. A construction of self - tuning PID control algorithm and its applications. Proc. SICE, Matsuyama, July 25 - 27 1989