On-line Learning Control of Nonlinear Systems Using Local Affine Mapping-based Networks

Jin Young Choi*, Dong Sung Kim**

ABSTRACT

This paper proposes an on-line learning controller which can be applied to nonlinear systems. The proposed on-line learning controller is based on the universal approximation by the local affine mapping-based neural networks. It has self-organizing and learning capability to adapt itself to the new environment arising from the variation of operating point of the nonlinear system. Since the learning controller retains the knowledge of trained dynamics, it can promptly adapt itself to situations similar to the previously experienced one. This prompt adaptability of the proposed control system is illustrated through simulations.

Keywords : On-line learning control, Adaptation, Affine mapping, Nonlinear system

Ⅰ. INTRODUCTION

To apply linear control theory to nonlinear systems, linearization in the vicinity of a specific operating point is inevitable. If there are variations in the operating point, the entire control system may be unstable due to the effect of the unmodeled dynamics. By using robust control, the system can maintain stability within a moderate range[1]. Another method to overcome this problem is to employ a real-time compensation using adaptive control[2]. By real-time adaptive compensation, the controller can maintain a desirable performance to some degree. However, if there are sudden changes in the parameters or operating points of the nonlinear system, the control performance falls drastically due to the long transitional adaptation period. Furthermore, since the conventional adaptive control can not store the control information on the past environment, it can not rapidly adapt even when the same environment occurs again.

To the contrary, a living thing always adapts itself quickly to any situation similar to that of the past since it retains previously experienced one. Therefore, a similar mechanism to living things can be applied to the control of nonlinear systems with time varying environments. In neural adaptive control, linear compensators used in conventional adaptive control are replaced by neural networks. It results in a good performance by handling the nonlinearity of the wide area[4][5]. Multi-Layer Perceptron(MLP) is widely used in neural adaptive control. But, depending on the complexity of the input/output mapping

---

*School of Electrical Engineering/ERC-ACI, Seoul National University
**Engineering Research Center for Advanced Control Instrumentation, Seoul National University

and the initial weight of the MLP, the error backpropagation learning may fall into a local minimum or a flat area which may require a long learning time or lead to unsuccessful learning. Hence a fast adaptive neural network is required for the on-line learning control systems.

This paper proposes an approach of on-line learning control for the nonlinear system in the environment of a varying operating point. For the fast on-line adaptation, we introduce a special type of neural network referred to as Local Affine Mapping-based Network. This network is used for identifying the controller and the nonlinear dynamics of a plant. In addition, simulations illustrate the prompt adaptation capability of the proposed on-line learning control systems when the operating point varies.

Ⅱ. LOCAL AFFINE MAPPING-BASED NETWORK

Given that the input/output data, which is produced from the model, have a relation to the nonlinear function($f: X \in R^n \rightarrow R$) about the universal approximator for identifying input/output mapping has to be considered. This approximator is usually expressed by using a specific parameterized element function. A nonlinear mapping model is characterized by the form of element function, the combination method of element functions and the estimation method of parameters. In this paper, we use the affine mapping for element function. The affine mapping is defined as

$$\psi_i(x) = \boldsymbol{w}^T{}_i\, x + b_i. \tag{1}$$

where $\boldsymbol{w}_i$ is an n-dimensional parameter vector and b_i is a bias constant. If a nonlinear mapping is given, an affine mapping can represent local area mapping approximately. So, we can set the fuzzy-rule as follows:

$$\begin{aligned} &Rule(i): IF \quad \boldsymbol{x} \in N(\boldsymbol{c}_i,\ \delta_i), \\ &\qquad then\ f(\boldsymbol{x}) = \psi_i(\boldsymbol{x}), \quad i = 1, \ldots, M. \end{aligned} \tag{2}$$

where $N(\boldsymbol{c}_i,\ \delta_i)$ represents a neighborhood with center vector $\boldsymbol{c}_i$ and radius δ_i, while M is the number of fuzzy rules. In this fuzzy rule, the premise part is represented by the following radial basis function

$$\mu_i(\boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{c}_i\|^2 / \gamma_i), \quad i, \ldots, M. \tag{3}$$

where $\|\boldsymbol{x} - \boldsymbol{c}_i\|$ represents the euclidean distance between the input $\boldsymbol{x}$ and the central point $\boldsymbol{c}_i$. The vigilance parameter γ_i determines the magnitude of the neighborhood. For the satisfaction to partition of unity[7], the membership function is normalized as follows:

$$\bar{\mu}_i(\boldsymbol{x}) = \frac{\mu_i(\boldsymbol{x})}{\sum_{j=1}^{M} \mu_j(\boldsymbol{x})}. \tag{4}$$

For the identification function $\hat{F}$, we use the membership function for the integration of several affine mappings.

$$\hat{F}(\theta,\ \boldsymbol{x}) = \sum_{i=1}^{M} \bar{\mu}(\boldsymbol{x}) \phi_i(\boldsymbol{x}), \quad \boldsymbol{x} \in X. \tag{5}$$

where $\bar{\mu}_i(\boldsymbol{x})$ is a basis function that satisfies the partition of unity. The parameter set θ expressed as

$$\theta = \{ (\boldsymbol{c}_i, \boldsymbol{w}_i, b_i) \mid \boldsymbol{c}_i \in R^n, \boldsymbol{w}_i \in R^n, b_i \in R, i = 1, 2, \ldots, M \}. \tag{6}$$

The identification of $\hat{F}$ can be realized by a connectionist model[5] as shown in Fig. 1. The connectionist model is composed of 3 layers(input, hidden and output). Each unit of the hidden layer has 2 nodes: the RF and the LF node.

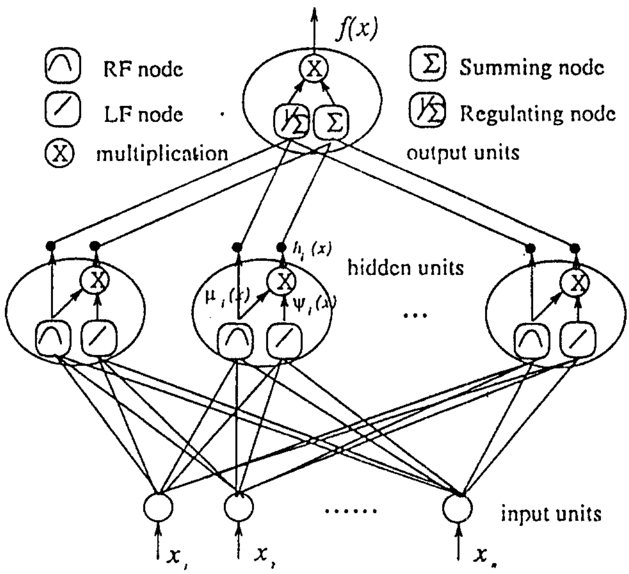


Figure 1 : Structure of the local affine mapping-based network

The RF node takes the central point of a specific receptive field as a parameter, where the output is determined by radial basis function. The product of the LF and the RF node becomes the output of the hidden unit, i.e. the i th output of the hidden units is as follows:

$$h_i(\boldsymbol{x}) = \mu_i(\boldsymbol{x})\, \psi_i(\boldsymbol{x}). \tag{7}$$

The RF node transmit unchanged signals to the output unit. The output unit is made up of 2 nodes: the Regulating node and the summing node. The role of the former is to regulate the output of the RF nodes so that the sum of the RF nodes satisfies partition of unity. The regulating node produce inverse value of the sum of the input. The role of the latter is to add localized affine functions that are produced by the hidden units. The product of the outputs in the regulating node and summing node becomes final output.

Ⅲ. ON-LINE LEARNING CONTROL SYSTEM

3.1 System Structure

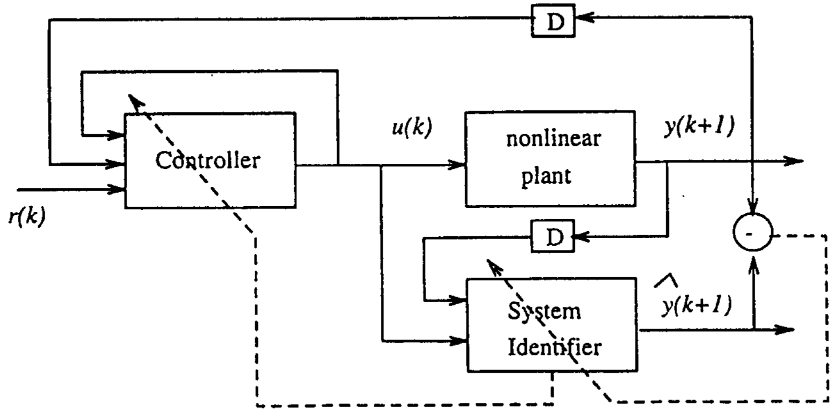


Figure 2 : Overall architecture of on-line learning control systems

The whole structure of the control system which was a the predescribed neural network is shown in Fig.2. This control system is composed of the identification part and the control part. The identification part continuously identifies nonlinear dynamics with the observed informations and consequently determines parameters of control part in the on-line environment. The dynamics of the plant is expressed by an input/output mapping of

$$y(k+1) = f(u(k), \ldots, u(k-m), y(k), \ldots, y(k-n)). \tag{8}$$

For the application of the predescribed neural network to the identification part, the input of the LF node is defined by

$$\bar{\boldsymbol{x}} = [u(k), \ldots, u(k-m), y(k), \ldots, y(k-n), 1]. \tag{9}$$

and the input of the RF node

$$\boldsymbol{x} = [u(k), \ldots, u(k-m), y(k), \ldots, y(k-n)]. \tag{10}$$

Then, the output of the identification part becomes

$$\hat{y}(k+1) = \sum_i \hat{\mu}_i(\boldsymbol{x})\ \boldsymbol{w}^T{}_i\ \bar{\boldsymbol{x}}. \tag{11}$$

where $\boldsymbol{w}_i$ is a $(n+m+1)$ dimensional weight vector including bias which is expressed as

$$\boldsymbol{w}_i = [w_{i0}, w_{i1}, \ldots, w_{i(n+m)}]. \tag{12}$$

This weight vector is updated in the on-line situation by the learning method described in the next section. The structure of the control part also uses the predescribed neural network. In order to align it with the identification part of the system, the RF node of the control part uses the same input as the RF node of the identification part.

The input of the LF node in the control part is

$$\bar{v} = [r(k), u(k-1), ..., u(k-m+1), y(k), ..., y(k-n+1), 1] \quad (13)$$

where $r(k)$ is the reference input. Then, the output of the control part becomes

$$\hat{u}(k) = \sum_i \bar{\mu}_i(\boldsymbol{x}) \ \theta_i^T \ \bar{\boldsymbol{v}}. \quad (14)$$

where θ_i is the $(n+m+1)$ dimensional weight vector including bias and it is represented by

$$\theta_i = [\theta_{i0}, \theta_{i1}, ..., \theta_{i(n+m)}]. \quad (15)$$

Since the control part basically represents the inverse dynamics of the plants, the weight vector of the control part is determined by the following weight vector of the identification part.

$$\theta_{i0} = \frac{1}{w_{i0}} \ , \quad (16)$$

$$\theta_{ij} = \frac{-w_{ij}}{w_{i0}} \ , \qquad j = 1, ..., n+m. \quad (17)$$

3.2 Learning of System Identifier

The learning of the proposed model is composed of the unsupervised learning for the RF node and the supervised learning for the LF node. In the identification part of the system, the learning of the connectionist model is done by automatically generating the hidden units according to the given input patterns and tuning the parameters. The learning strategy of the proposed model is expressed as:

- When a new input pattern is presented and if the output value of the regulating node is larger than an arbitrary threshold value T, then it is regarded that the unit which is assigned to the receptive field does not exist. So, a new hidden unit is generated. The input vector is used to initialize the central point of the RF node and the weight value of the LF node is set arbitrarily.
- On the other hand if the regulating node's output is smaller than T, we update the parameters output regulates the parameters of the biggest hidden unit which has the largest RF node's output.

The central point of the RF node is updated by

$$\boldsymbol{c}_i(k+1) = \boldsymbol{c}_i(k) + \frac{1}{k+1} \ (x - \boldsymbol{c}_i(k)). \quad (18)$$

where k is the number of updating the ith center of the RF node, $\boldsymbol{c}_i$. Also, the weight value of the ith LF node is updated by

$$G_i(k) = \frac{R_i(k-1) \ \bar{\boldsymbol{x}}}{1 + \bar{\boldsymbol{x}}^T R_i(k-1) \ \bar{\boldsymbol{x}}} \ , \quad (19)$$

$$R_i(k) = R_i(k-1) - G_i(k) \ \bar{\boldsymbol{x}}^T R_i(k-1), \ R_i(0) = \alpha \ I, \quad (20)$$

$$\bar{\boldsymbol{w}}_i(k) = \bar{\boldsymbol{w}}_i(k-1) + G_i(k) [f(\boldsymbol{x}) - h_i(\boldsymbol{x})], \quad (21)$$

where $\bar{x}$ is the $(n+m+1)$ dimensional vector $[x\ 1]$, I is the unit vector, and α is a constant satisfying $\alpha \gg 1$.

Ⅳ. COMPUTER SIMULATIONS

To test the effectiveness of the proposed control system, it was applied to the following nonlinear system.

$$y(k+1) = \frac{y(k) + y(k-1)}{1 + y^2(k) + y^2(k-1)} + u^3(k) \tag{22}$$

Two types of signals were used as reference inputs.

Reference input 1 :

$$r(k) = 0.6 + 0.2s(300-\bar{k}) - 0.1s(600-\bar{k}) - 0.3s(900-\bar{k}) - 0.2s(1200-\bar{k}) + 0.1s(1500-\bar{k}) + 0.3s(1800-\bar{k}). \tag{23}$$

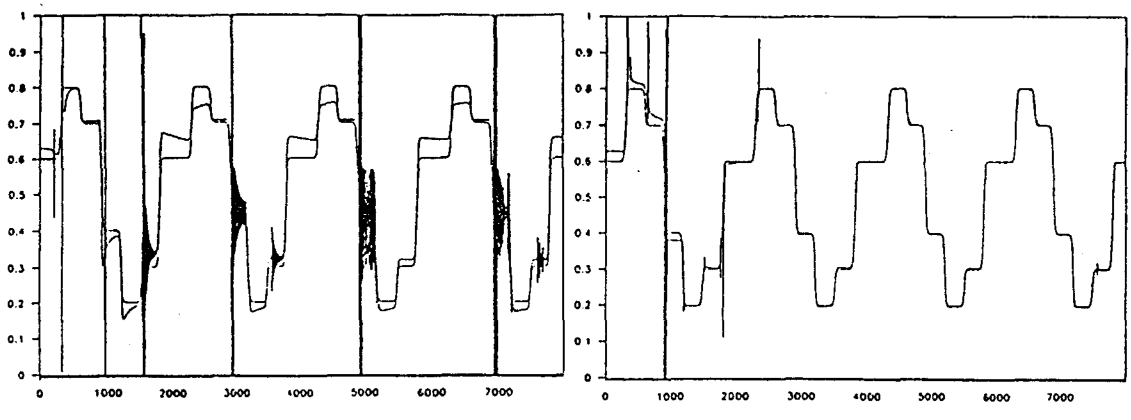
where $\bar{k} = k\%2000$; % denotes the remaining modules, therefore the above mentioned signal has a period of 2000.

Also $s(t)$ is expressed as follows :

$$s(t) = \frac{1 - e^{0.1t}}{1 + e^{0.1t}} \ . \tag{24}$$

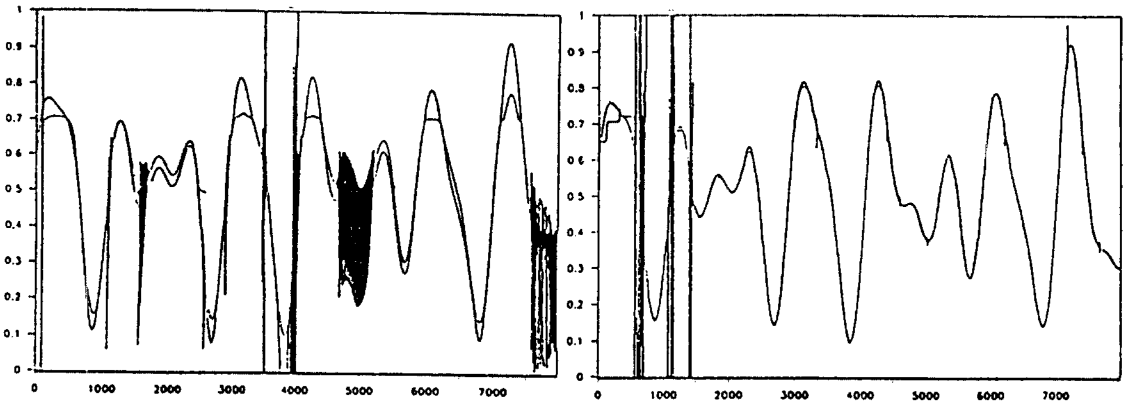
Reference input 2 :

$$r(k) = 0.5 + 0.2\sin(\pi t/300) + 0.2\cos(\pi t/500) + 0.1\sin(\pi t/700). \tag{25}$$



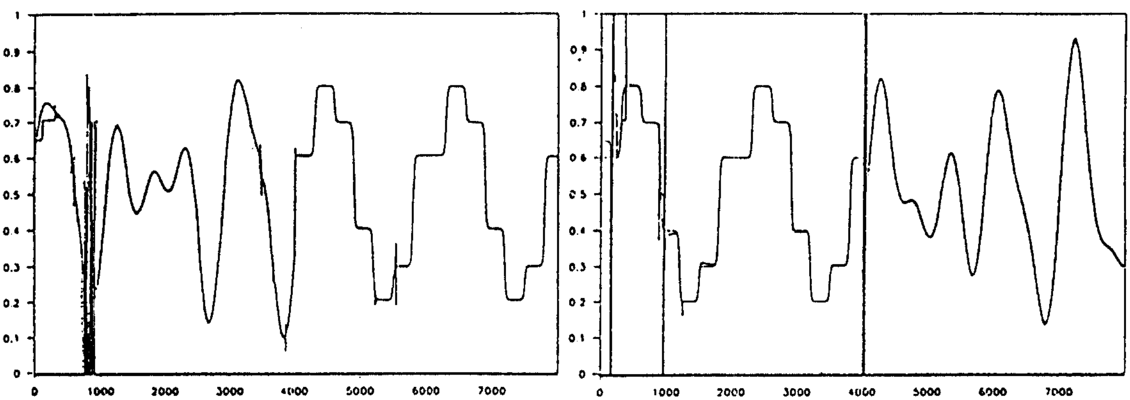
(a) (b)

Figure 3 : Adaptation result for reference input 1 :
(a) is the result of the linear model, (b) is the result of the model with $\gamma = 0.01$(33 hidden units produced)



(a) (b)

Figure 4 : Adaptation result for reference input 1 :
(a) is the result of the linear model, (b) is the result of the model with $\gamma=0.01$ (101 hidden units produced)



(a) (b)

Figure 5 : Adaptation result of the on-line learning control when the model was trained in the off-line environment until k = 4000 :

The result of the on-line learning control for reference input 1 is shown in Fig.3. Fig.3(a) is the result when one hidden unit is used. It can be seen that the control system with a linear controller inevitably encounters the transitional adaptation period whenever the operating point vary. It adapts totally to the start even if a previously adapted situation arises again. In contrast to (a)'s linear models, as shown in Fig.3(b), the system retains information about previously adapted situations. When there is a repeat of the previously adapted situation, the control system can track the reference input instantly. In this experiment, 101 hidden layer units were automatically generated when γ was set to 0.01. The result of the continually varying reference input 2 is shown in Fig.4 where it shows a similar result to Fig.3. It can be seen that the initial transitional period in the proposed method is worse than in Fig.3. This is because of the self-organizing it needs to do at the start where the hidden units are continuously generated. Fig.5 shows the result of real-time adaptation in an experiment where off-line learning was done initially(k= 4000). It can be observed that the initial adaptation to a new input becomes fast in a transient period.

V. CONCLUSION

This paper has proposed an on-line learning control method using a local affine mapping-based neural network. The proposed controller possesses a fast adaptability than those based on existing neural networks. In addition, unlike adaptive control based on linear models, it retains previously experienced dynamical information. Hence, it can rapidly adapt itself to any situation similar that of the past. The proposed model has merits as follows: 1) Comparing with the MLP-based controller, it provides lower computational complexity and fast learning speed. 2) The local affine mapping provides effectiveness in application to the on-line learning control. Affine mappings facilitate the design of the system by applying the linear system theory to each hidden units. The proposed control system will be extended to the case of multi-input multi-output systems.

REFERENCES

1. Reichert R. T., "Dynamic Scheduling of Modern-Robust-Control Autopilot Designs for Missiles," IEEE Control Systems Magazine, Vol. 12, No. 5., pp. 35-42, 1992.
2. Ahmed-Zaid F., et al., "Accommodation of Failures in the F-16 Aircraft Using Adaptive Control," IEEE Control Systems Magazine, Vol. 11, No. 1., pp. 73-78, 1991.
3. Stengel R. F., "Intelligent Failure-Tolerant Control," IEEE Control Systems Magazine, Vol. 11, No. 4., pp. 14-23, 1991.
4. Farrel J., et al., "Using Learning Techniques to Accommodate Unanticipated Faults," IEEE Control Systems Magazine, Io. 13, No. 3., pp. 40-49, 1993.
5. Narendra K. S., et al., "Intelligent Control Using Neural Networks," IEEE Control Systems Magazine, Vol. 12, No. 2., pp. 11-18, 1992.
6. Choi J. Y., et al., "Nonlinear Estimation Network with Versatile Bump Shaping Units for Fast On-line Adaptation," IEEE International Conference on Neural Networks, Orlando, June 26, 1994.
7. Spivak M., Calculus on Manifild, W. A. Benjamin, New York, 1965.