이동구간 최적 제어에 의한 전력계통 안정화의 분산제어 접근 방법

論 文
48A-7-1

A Decentralized Approach to Power System Stabilization by Artificial Neural Network Based Receding Horizon Optimal Control

崔 勉 松*
(Myeon-Song Choi)

Abstract -This study considers an implementation of artificial neural networks to the receding horizon optimal control and its applications to power systems. The Generalized Backpropagation-Through-Time (GBTT) algorithm is presented to deal with a quadratic cost function defined in a finite-time horizon. A decentralized approach is used to control the complex global system with simpler local controllers that need only local information. A *Neural network based Receding horizon Optimal Control* (NROC) law is derived for the local nonlinear systems. The proposed NROC scheme is implemented with two artificial neural networks, *Identification Neural Network* (IDNN) and *Optimal Control Neural Network* (OCNN). The proposed NROC is applied to a power system to improve the damping of the low-frequency oscillation. The simulation results show that the NROC based power system stabilizer performs well with good damping for different loading conditions and fault types.

Key Words : Neural network, Receding horizon optimal control, Decomposition, Power system stabilizing control

1. Introduction

Control of large scale systems such as a power system has been recognized as one of the foremost challenges in control engineering due to its nonlinearty and complexity. In handling nonlinearty, the use of an artificial neural network is very attractive because of its nonlinear mapping ability. For complexity coming from high dimension or for the spatial distribution of a large scale system, decentralized control is a practical approach in solving the problem. Neural networks have attractive capacity in handling sensory information, and performing collective learning from the data sets given for a sub-system in the decentralized control approach. The approximation property of neural networks can make it possible to organize a sub-system dynamics, including interaction effects between sub-systems, to a certain degree by training the input/output relationships obtained in the full system operation. From this point of view, a Neural network based Receding horizon Optimal Control (NROC) for a large scale system is proposed when only a local information of input/output operation data for a sub-system is available.

An important control problem arising from the interconnection of power systems is the stability problem, usually in the form of self-excited low frequency oscillations [1]. This type of instability is actually caused by the cross-coupling between the speed control loop and the voltage control loop shown in Fig. 1.

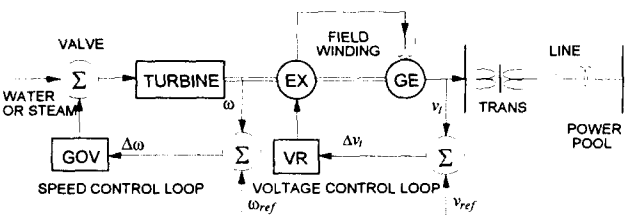

Fig. 1 Basic components of an electric power plant

Over the past decades, there have been considerable researches on Power System Stabilizer (PSS) design to damp the low-frequency oscillations. A practical PSS to enhance the damping of the low-frequency oscillations must be robust over a wide range of operating conditions and capable of damping oscillation modes [1-5]. However, conventional PSS design approaches based on linearlized models at the normal operating point have deficiencies and difficulties coming from nonlinearties in the system model when the system in abnormal or emergency state resulted from a fault.

There are cases where neural networks are applied for power system stabilizing control [6, 7]. However, these cases are limited to speed deviation control with supplementary excitation signal and presented for a system with a single generator connected to an infinite bus to avoid the complexity of the interconnected power system dynamics. Yu [1] stressed that it is desirable in the PSS controller design to consider mutual interactions among generators networked in a power system. In this

* 正 會 員 : 明知大 電氣情報制御工學部 敎授
接受日字 : 1998年 2月 7日
最終完了 : 1999年 6月 7日

paper, neural networks are applied to stabilize the low-frequency oscillation for a power system with two generators interconnected by a tie-line. In our earlier paper[6], neural networks were used to design a power system stabilizer. However, due to the complexity of the problem neural networks were used only to estimate power flow dynamic, and other machine dynamics were assumed known, By taken advantage of the decentralized control capability of NROC, we have extended the approach to estimate the plant output response directly. Thus no assumptions are made on plant parameters, resulting in al general purpose PSS that can be applied to different machines.

The use of neural networks in control has been focused mostly on the Model Reference Adaptive Control (MRAC) problem [7-9]. However, the MRAC approach has difficulty in selecting an appropriate reference model. In order to develop a general purpose PSS, this paper introduces a new class of control problems with neural networks, namely the receding horizon optimal control problem, which resulted in the proposed NROC. The control objective is to minimize a general quadratic cost function of output errors and control efforts for a finite time receding horizon. The general purpose NROC is developed for an arbitrary nonlinear plant and the Generalized Backpropagation-Through-Time (GBTT) algorithm is developed to train NROC in Section II. The NROC is then applied to the power system stabilization problem along with number of case studies in Section III, and conclusions are drawn in Section IV.

2. Design of Neuro-Controller

2.1. Receding Horizon Optimal Control Problem

We consider a system in the form of the general nonlinear auto-regressive moving average (NARMA) model:

$$y(k+1) = f(y(k), y(k-1), \cdots, y(k-n+1), u(k), u(k-1), \cdots, u(k-m+1)), \quad (1)$$

where y and u, respectively, represent output and input variables, k represents time index, and n and m represent the respective output and input delay orders. When the target output of a plant holds up for some time and varies from time to time, the control objective can be defined as minimizing the following well-known quadratic cost function in a finite time receding horizon:

$$J = \sum_{k=1}^{Nh} \{(y^{ref} - y^{(k+1)})^T Q(y^{ref} - y^{(k+1)}) + (u^{(k)})^T R(u^{(k)})\}, \quad (2)$$

where y_{ref} is a given target reference output. Q and R are positive weighting factors, and Nh is the length of time horizon considered.

An explicit solution for the optimal control problem is available for linear systems. However, it is impossible to get a solution for nonlinear systems in general. To cope with this problem, neural network based modeling and control approach for nonlinear systems has been suggested. Iiguni and Sakai [10] tried to design a nonlinear regulator using neural networks, however they are limited to a linear system with uncertain parameters.

In this paper, a novel architecture, NROC, is proposed as a new approach to the receding horizon optimal control problem for a nonlinear system using neural networks. The proposed NROC is designed with two neural networks. One is *Identification Neural Network* (IDNN) which is used to identify the nonlinear characteristics of a sub-system dynamics, and the other is *Optimal Control Neural Network* (OCNN) to find the receding horizon optimal control law for the sub-system. Fig. 2 shows an architecture of NROC for a nonlinear system.

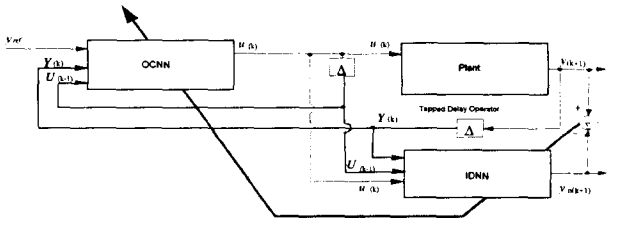

Fig. 2. Overall scheme of the neural network based receding horizon optimal control

2.2. Training of Identification Neural Network (IDNN)

The proposed NROC is made of two multilayer feedforward neural networks. The function of the training of IDNN corresponds to the identification of the plant dynamics. It is then used to backpropagate the equivalent error to OCNN. Training the IDNN can be regarded as an approximation process of a nonlinear function using input-output data sets [11]. A NARMA model (1) can be viewed as a nonlinear mapping from (n+m)-dimensional input space to a one-dimensional output space:

$$y(k+1) = f(I_i(k)), \quad (3)$$

where $I_{i(k)}$ is the identifier input vector defined as $I_i(k) \triangleq \{y(k), y(k-1), \cdots, y(k-n+1), u(k), u(k-1), \cdots, u(k-m+1)\}$.

The IDNN can be viewed as a nonlinear function, F, to approximate f ;

$$\hat{y}_{(k+1)} = F(I_i(k), \vec{W}_i), \quad (4)$$

where $\vec{W}_i$ is the weight parameter matrix in the IDNN and $\hat{y}$ is estimated of y. Then, training of the IDNN is to adjust weight parameters so that IDNN can emulate the nonlinear function of the plant dynamics using the

input-output training patterns that are obtained from the wide range plant operation under various operating conditions. The error backpropagation algorithm (BPA)[11] can be used to reduce an error function for identification (EI) defined as follows:

$$EI = \frac{1}{Nd}\sum_{i=1}^{Nd}\sum_{k=0}^{Nh-1}\frac{1}{2}(y^i(k+1) - \hat{y}^i(k+1))^2, \quad (5)$$

where $y_{(k)}$ is the output of the plant, $\hat{y}_{(k)}$ is the output of IDNN at time step k, Nd is the number of training data sets, the superscript, i, represents the i-th training sample, and Nh is the number of time horizon. Learning process in the BPA is based upon an output sensitivity as follows:

$$\delta_y^k \triangleq -\frac{\partial EI}{\partial EI\ \hat{y}_{(k)}} = \frac{1}{Nd}(y^i(k) - \hat{y}^i(k))$$

$$i = 1, 2, \ldots, Nd,\ k = 1, 2, \ldots, Nh \quad (6)$$

This error is then used to compute an equivalent error for a node in an arbitrary layer using the BPA. Through the learning process, the plant dynamic characteristics are stored in the weight parameters of IDNN. When training is finished, IDNN is presumed to have learned the plant characteristics approximately with converged weight parameters , $\vec{W}_i^*$,i.e.,

$$y_{(k+1)} = f(I_{i(k)}) \approx \hat{y}_{(k+1)} = F(I_{i(k)}, \vec{W}_i^*) \quad (7)$$

2.3. Training of Optimal Control Neural Network (OCNN)

The role of the OCNN is to stabilize tracking error dynamics by generating control inputs which minimizes the quadratic performance index (2) discussed before. From the NARMA model Eq. (1), the feedback control input can be viewed as an inverse mapping

$$u(k) = h(y(k), y(k-1), \cdots, y(k-n+1), u(k-1), u(k-2), \cdots, u(k-m+1)). \quad (8)$$

Its corresponding OCNN can be represented as a nonlinear network H:

$$u(k) = H(y(k), y(k-1), \cdots, y(k-n+1), u(k-1), u(k-2), \cdots, u(k-m+1), \vec{W}). \quad (9)$$

Since the optimal control law is not known for the system, its not available for training. Therefore, the OCNN will learn the control law by trial and error. The learning process by trial and error consists of two parts. First, from the given initial state, the OCNN drives the IDNN for N steps forward. Secondly, update the weight parameters of the OCNN using the equivalent error generated by the Generalized Backpropagation-Through-Time algorithm presented in the following section.

2.4. Generalized Backpropagation-Through-Time algorithm

Generalized Backpropagation-Through-Time (GBTT) [12] is to generate an equivalent error from a general quadratic cost function (2), and it is an extension of BTT algorithm of Werbos [13]. The original BTT was for the cost function with output error only. On the other hand, GBTT is for the general quadratic cost function (2) which includes not only output errors, but also input variables as well. The GBTT is based upon two sensitivities, Output Sensitivity, δ_y^k, and Input Sensitivity, δ_u^k, of the cost function defined by:

$$\delta_y^k \triangleq \frac{\partial J}{\partial y(k)}, \qquad k = 1, 2, 3, \ldots, N+1. \quad (10)$$

$$\delta_u^k \triangleq \frac{\partial J}{\partial u(k)}, \qquad k = 0, 1, 2, \ldots, N. \quad (11)$$

An output $y_{(k)}$ at an arbitrary time-step k influences both the plant dynamics (1) and the inverse dynamics (8). Since the plant dynamics (1) is defined with n delayed output variables, i.e., an arbitrary output $y_{(k)}$ will influence the input for the next n steps, i.e., $y_{(k+i)}$ is a function of $y_{(k)}$ for $i = 1, 2, \ldots, n$. Similarly, since the inverse dynamics (8) also has n delayed output variables, an output $y_{(k)}$ will influence the plant dynamics for the next n steps, i.e., $u_{(k+1)}$ is a function of $y_{(k)}$ for $i = 0, 1, 2, \ldots, n-1$. Recall that the performance index (2) is defined on a finite interval, i.e.,

$$J = J(y_{(j+1)}, u_{(j)}; j = 1, 2, \cdots, N). \quad (12)$$

Thus, the $y_{(k)}$ for some k is gradient of J with respect to an output

$$\frac{\partial J}{\partial y(k)} = \sum_{\substack{i=1 \\ k+i \le N+1}}^{n} \frac{\partial J}{\partial y(k+i)}\frac{\partial y(k+i)}{\partial y(k)} + \sum_{\substack{i=0 \\ k+i \le N}}^{n-1} \frac{\partial J}{\partial u(k+i)}\frac{\partial u(k+i)}{\partial y(k)} - Q(y_{ref} - y(k))$$

$$= \sum_{\substack{i=k+1 \\ i \le N+1}}^{k+n} \frac{\partial J}{\partial y(i)}\frac{\partial y(i)}{\partial y(k)} + \sum_{\substack{i=k \\ i \le N}}^{k+n-1} \frac{\partial J}{\partial u(i)}\frac{\partial u(i)}{\partial y(k)} - Q(y_{ref} - y(k)). \quad (13)$$

From the definition of sensitivities, (10) and (11), the above becomes

$$\delta_y^k = \sum_{\substack{i=k+1, \\ i \le N+1}}^{k+n} \delta_y^i \frac{\partial y(i)}{\partial y(k)} + \sum_{\substack{i=k, \\ i \le N}}^{k+n-1} \delta_u^i \frac{\partial u(i)}{\partial y(k)} + Q(y_{ref} - y(k)). \quad (14)$$

Note that this output sensitivity is depending on the input sensitivities as well. The input $u_{(k)}$ for some k is

sensitivity can be derived in a way similar to the output sensitivity. The gradient of J with respect to an input

$$\frac{\partial J}{\partial u(k)} = \sum_{\substack{i=1 \\ k+i\le N+1}}^{m} \frac{\partial J}{\partial y(k+i)}\frac{\partial y(k+i)}{\partial u(k)} + \sum_{\substack{i=1 \\ k+i\le N}}^{m-1} \frac{\partial J}{\partial u(k+i)}\frac{\partial u(k+i)}{\partial u(k)} + Ru(k)$$
$$= \sum_{\substack{i=k+1 \\ i\le N+1}}^{k+m} \frac{\partial J}{\partial y(i)}\frac{\partial y(i)}{\partial u(k)} + \sum_{\substack{i=k+1 \\ i\le N}}^{k+m-1} \frac{\partial J}{\partial u(i)}\frac{\partial u(i)}{\partial u(k)} + Ru(k). \quad (15)$$

Again, from the definition of sensitivities, (10) and (11), it becomes

$$\delta_u^k = \sum_{\substack{i=k+1, \\ i\le N+1}}^{k+m} \delta_y^i \frac{\partial y(i)}{\partial u(k)} + \sum_{\substack{i=k+1, \\ i\le N}}^{k+m-1} \delta_u^i \frac{\partial u(i)}{\partial u(k)} - Ru(k) \quad (16)$$

This input sensitivity is also depending on the output sensitivities, and both sensitivity equations are coupled to one another. Since the plant dynamics (1) and the inverse dynamics (8) are not known, they are approximated by the corresponding networks, the IDNN F and the feedback neuro-controller H, to yield

$$\delta_y^k = \sum_{\substack{i=k+1, \\ i\le N+1}}^{k+m} \delta_y^i \frac{\partial F_i}{\partial \hat{y}_{(k)}} + \sum_{\substack{i=k, \\ i\le N}}^{k+m-1} \delta_u^i \frac{\partial H_i}{\partial \hat{y}_{(k)}} + Q(y_{ref} - \hat{y}_{(k)}) \quad (17)$$

$$\delta_u^k = \sum_{\substack{i=k+1, \\ i\le N+1}}^{k+m} \delta_y^i \frac{\partial F_i}{\partial u(k)} + \sum_{\substack{i=k+1, \\ i\le N}}^{k+m-1} \delta_u^i \frac{\partial H_i}{\partial u(k)} - Ru(k) \quad (18)$$

It should be noted that the las in (17) and (18) are, respectively, the error terms for the output and input variables, and the terms under summation operations are the error (or delta) terms backpropagated through the networks F and H. For example, $\frac{\partial F_i}{\partial \hat{y}_{(k)}}$ (or $\frac{\partial F_i}{\partial \hat{u}_{(k)}}$) is the error δ_y^i (or δ_u^i) backpropagated through the network F to the input node $\hat{y}_{(k)}$(or $u_{(k)}$).

The objective of the GBTT is to compute the sensitivity δ_u^k, which will be used as the equivalent error for training of the OCNN. This can be achieved by solving (17) and (18) backward starting from $j = N+1$.

j=N+1:

$$\delta_u^{N+1} = 0$$
$$\delta_y^{N+1} = Q(y_{ref} - \hat{y}_{(N+1)})$$

j=N:

$$\delta_u^N = \delta_y^{N+1}\frac{\partial F_{N+1}}{\partial u(N)} - Ru(N)$$
$$\delta_y^N = \delta_y^{N+1}\frac{\partial F_{N+1}}{\partial \vec{y}(N)} + \delta_y^N \frac{\partial H_N}{\partial \vec{y}(N)} + Q(y_{ref} - \hat{y}_{(N)}),$$

j=N-1:

$$\delta_u^{N-1} = \delta_y^{N+1}\frac{\partial F_{N+1}}{\partial u(N-1)} + \delta_y^N \frac{\partial F_N}{\partial u(N-1)} + \delta_u^N \frac{\partial H_N}{\partial u(N-1)} - Ru(N-1)$$
$$\delta_y^{N-1} = \delta_y^{N+1}\frac{\partial F_{N+1}}{\partial \vec{y}(N-1)} + \delta_y^N \frac{\partial F_N}{\partial \vec{y}(N-1)} + \delta_u^N \frac{\partial H_N}{\partial \vec{y}(N-1)} + \delta_u^{N-1}\frac{\partial H_{N-1}}{\partial \vec{y}(N-1)} + Q(y_{ref} - \hat{y}_{(N-1)}),$$

j=N-2:

$$\delta_u^{N-2} = \delta_y^{N+1}\frac{\partial F_{N+1}}{\partial u(N-2)} + \delta_y^N \frac{\partial F_N}{\partial u(N-2)} + \delta_y^{N-1}\frac{\partial F_{N-1}}{\partial u(N-2)} + \delta_u^N \frac{\partial H_N}{\partial u(N-2)} + \delta_u^{N-1}\frac{\partial H_{N-1}}{\partial u(N-2)} - Ru(N-2)$$
$$\delta_y^{N-2} = \delta_y^{N+1}\frac{\partial F_{N+1}}{\partial \vec{y}(N-2)} + \delta_y^N \frac{\partial F_N}{\partial \vec{y}(N-2)} + \delta_y^{N-1}\frac{\partial F_{N-1}}{\partial \vec{y}(N-2)} + \delta_u^N \frac{\partial H_N}{\partial \vec{y}(N-2)} + \delta_u^{N-1}\cdot\frac{\partial H_{N-1}}{\partial \vec{y}(N-2)} + \delta_u^{N-2}\frac{\partial H_{N-2}}{\partial \vec{y}(N-2)} + Q(y_{ref} - \hat{y}_{(N-2)}),$$

j=k:

$$\delta_u^k = \delta_y^{k+m}\frac{\partial F_{k+m}}{\partial u(k)} + \cdots + \delta_y^{k+1}\frac{\partial F_{k+1}}{\partial u(k)} + \delta_u^{k+m-1}\frac{\partial H_{k+m-1}}{\partial u(k)} + \cdots + \delta_u^{k+1}\frac{\partial H_{k+1}}{\partial u(k)} - Ru(k)$$
$$\delta_y^k = \delta_y^{k+n}\frac{\partial F_{k+n}}{\partial \vec{y}(k)} + \cdots + \delta_y^{k+1}\frac{\partial F_{k+1}}{\partial \vec{y}(k)} + \delta_u^{k+n-1}\frac{\partial H_{k+n-1}}{\partial \vec{y}(k)} + \cdots + \delta_u^{k+1}\frac{\partial H_{k+1}}{\partial \vec{y}(k)} + \delta_u^k\frac{\partial H_k}{\partial \vec{y}(k)} + Q(y_{ref} - \hat{y}_{(k)}).$$

(1) Forward simulator: Before solving the sensitivity equations (17) and (18), the error terms need to be generated. This can be done by driving IDNN with OCNN for N step forward. This process is illustrated by the forward simulator shown in Fig. 3, where Δ is the tapped delay operator defined in Fig. 2.

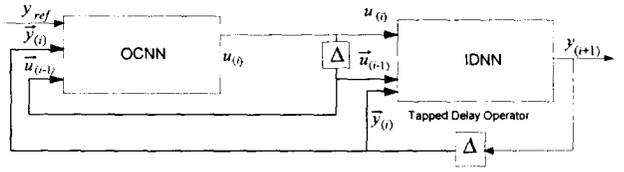

Fig. 3. Forward simulator for GBTT

Starting from an initial conditions, $y_{(1)}$, $u_{(0)}$, the forward simulator generates sequences of inputs, $u_{(1)}, u_{(2)}, \ldots, u_{(N)}$, and outputs, $\hat{y}_{(2)}$, $\hat{y}_{(3)}$, $\hat{y}_{(N+1)}$. These variables provides the error terms ($y_{ref} - \hat{y}_{(k)}$) and $u_{(k)}$ for any k.

(2) Backward simulator : The sensitivity equations (17) and (18), are to be simulated backward in order to backpropagate the error terms. This can be performed

by the backward simulator shown in Fig. 4, where, the summed advance operator ∇ is defined as a *dual* of the tapped delay operator mapping from a sequence of vector input $\{\vec{x}_{(i)}\}$ where $\vec{x}_{(i)} \; (x_{(i,1)}, x_{(i,2)}, \cdots, x_{(i,n)})$, to a scalar output defined as $x_{(i)} = \sum_{k=1}^{n} x_{(i+k,k)}$. The backward simulator can be shown to be the dual of the forward simulator, which is then constructed by using the duality prizncple, i.e., reversing the direction of arrows, interchanging the summers and nodes, and replacing the tapped delay operators with the summed advance operators.

Using the errors generated by the forward simulator, the backward simulator runs backward starting from $i = N$. It generates the sensitivities δ_y^{i+1} and δ_u^i. Noting that δ_u^i is in the output node of OCNN, it is used as the equivalent error in computing the weight parameter adjustment in OCNN, $\triangle \vec{W}^{-1}$.

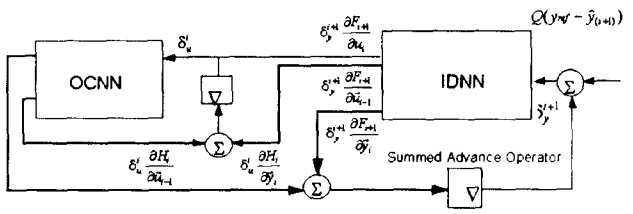

Fig. 4. Backward simulator for GBTT.

The process of the GBTT training algorithm is summarized as follows:

1) Set the weight parameters of OCNN with small random numbers.
2) Set the reference output and initial state with random numbers in the operation region of the plant.
3) Run the forward simulator for N step forward from $i = 1$.
4) Using the operation result in step 3), run the backward simulator backward from $i = N$ to evaluate the equivalent error δ_u^i and the weight adjustment vector $\triangle \vec{W}^i$.
5) Update the weight parameters in the OCNN by using the average of the weight adjustment vectors found in step 4).
6) Go to 2).

Training of OCNN is finished when the average decrease of the cost function converges to a small value for arbitrary reference outputs and initial conditions. Since the training algorithm is essentially a gradient descent method, the local minimum problem is a possibility. However, this problem can be avoided by starting with different initial weight parameters or with different number of nodes in OCNN [13].

2.5. Decomposition of Receding Horizon Optimal Control for a Large Scale System

Let us consider a MIMO discrete-time dynamics of a large scale system given by a general Nonlinear Auto-Regressive Moving Average (NARMA) model:

$$\vec{y}_{(k+1)} = f(\vec{Y}_{[k,n]}, \vec{U}_{[k,m]}), \tag{19}$$

where

$\vec{y}_k = (y_k^1, y_k^2, \cdots, y_k^M)^T$: output vector,

$\vec{u}_k = (u_k^1, u_k^2, \cdots, u_k^M)^T$: input vector,

$\vec{Y}_{[k,n]} = (\vec{y}_{(k)}: \vec{y}_{(k-1)} \cdots\cdots \vec{y}_{(k-n+1)})$: output history matrix,

$\vec{U}_{[k,n]} = (\vec{u}_{(k)}: \vec{u}_{(k-1)} \cdots\cdots \vec{u}_{(k-m+1)})$: input history matrix,

$\vec{f}' = (f'^1, f'^2, \cdots, f'^M)^T$: vector of input/output mapping.

Here, n and m are the appropriate orders for the input/output mapping, and M is the number of input/output pairs of the full system.

Design of an adequate optimal controller is difficult when the system is of high dimension or the global information of the system is not available. Therefore, the full or global control problem is decomposed into several local sub-system problems of low dimension.

Lets consider a large scale system shown in Fig. 5, where the full system is decomposed into several sub-systems by introducing interaction variables, Z^i.

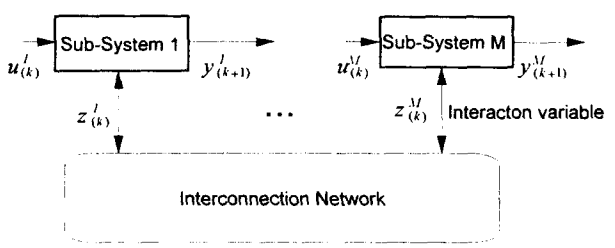

Fig. 5 Decomposition of a large scale system into sub-systems

The i-th sub-system dynamics can be represented as

$$\begin{aligned} y_{(k+1)}^i &= f^i(\vec{Y}_{[k,n]}, \vec{U}_{[k,m]}) \\ &= \vec{f}^i(Y_{[k,n]}^i U_{[k,m]}^i . Z_{[k,o]}^i) \end{aligned} \tag{20}$$

where $Y^i{}_{[k,n]} = \{y_{(k)}^i, y_{(k-1)}^i, \cdots, y_{(k-n+1)}^i\}$,

$U^i{}_{[k,m]} = \{u_{(k)}^i, u_{(k-1)}^i, \cdots, u_{(k-m+1)}^i\}$,

$Z^i{}_{[k,o]} = \{z_{(k)}^i, z_{(k-1)}^i, \cdots, z_{(k-o+1)}^i\}$,

Here, $y_{(k)}^i$ is the plant output scalar, $u_{(k)}^i$ is the control input scalar and , $z_{(k)}^i \in \Re^{Nz}$ is the interaction vector in Nz space which represents interaction effects of other sub-systems to the i-th sub-system by a nonlinear function as:

$$z_{(k)}^i \triangleq f_z^i(\vec{Y}_{[k,n]}, \vec{U}_{[k,m]}), \tag{21},$$

where n, m, and o represent the delay orders for output, input, and interaction variables, respectively. Then, the following approximate dynamics can be obtained by inserting equation (21) to equation (20) for some large $N3$ and $M3$ as a sub-system model :

$$y^i(k+1) \approx f^i(Y^i[k,N3], U^i[k,M3]). \quad (22)$$

In the neural network approach, the sub-system dynamics (22) can be approximated by training of a neural network by increasing the number of input/outputs for neural networks..

3. Power System Stabilization

3.1. Problem Description

The proposed NROC is applied in a power system to enhance the power system stabilization. In a normal operation of a generator, the turbine power keeps balance with the electric air-gap power resulting in zero acceleration and a constant speed or frequency. However, on occasion, disturbances or load perturbations upset the balance, and the power system experiences low-frequency oscillations. Thus our control objective is to minimize the low-frequency oscillations in the electric frequency with keeping terminal voltage of generator within limits. NROC is applied in the voltage control loop of a generator to minimize the quadratic performance index:

$$J = \frac{1}{2}\sum_{k=0}^{Nh}\{Q(\omega_{(k+1)} - \omega_{ref})^2 + R(u_{(k)})^2\}, \quad (23)$$

where is the output frequency of the plant, u is the supplementary excitation input and ω_{ref} is the normal synchronous speed. The weight parameters are set as Q=1.0 and R=0.04. This quadratic performance index not only keeps the frequency output from fluctuating near the reference frequency, the low frequency oscillation, but also limits the excitation input. The proposed NROC is applied to a power system network shown in Fig. 6. The power system consists of two power plants connected with four parallel transmission lines: one plant is a thermal unit and the other is an hydro unit. Parameters of the generators are given in Table 1. Two operating conditions, a normal and heavy loading operating conditions for the first unit, are given in Table 2 and Table 3, respectively. The study power system has sustained typical low-frequency oscillations. The control objective is to improve the system damping, i.e., to reduce the low-frequency oscillations in the outputs of the first power plant by applying the proposed NROC. The power system stabilizing control input is the supplementary excitation signal applied to the first power plant.

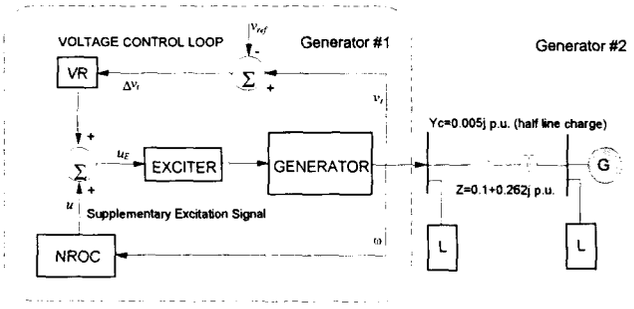

Fig. 6 A neural network based control architecture for a power plant

Table. 1. Parameters of generators

	T'_{do}	D	H	x_d	x'_d	x_q	x'_q	Selfbase [MVA]
Gen. #1	4.0	1.0	4.46	1.25	0.6	0.9	0.6	150
Gen. #2	6.3	1.0	5.5	0.94	0.4	0.65	0.4	150

Table. 2. Bus data for the normal loading condition (0.8 p.u.)

Bus	v_t	θ	P_e	Q_e	P_l	Q_l
1	0.997	0.0475	120	-20	70	10
2	1.0	0.0	146	37.4	195	10

Table. 3. Bus data for the heavy loading operating condition (1.0 p.u.)

Bus	v_t	θ	P_e	Q_e	P_l	Q_l
	0.993	0.0357	150	-20	100	10
	1.0	0.0	146	37.4	195	10

To study the low-frequency oscillation problem, the state equation for the synchronous machine is written as a third-order model. The state variables are (rotor speed), δ (torque angle), e'_q (voltage behind the transient reactance), where the change in flux linkage of the field winding is considered. The third-order model for a synchronous machine connected to a network at the j-th bus is [1]:

$$\frac{d\omega}{dt} = \frac{1}{M}(T_m - T_e + D(\omega_o - \omega)), (T_m = P_m/\omega, T_e = P_e/\omega) \quad (24)$$

$$\frac{d\delta}{dt} = \omega_b(\omega - \omega_o), (\omega_b = 2\pi f_o, \omega \approx 1) \quad (25)$$

$$\frac{de'_q}{dt} = \frac{1}{T'_{do}}[E_{FD} - e'_q - \frac{(x_d - x'_d)}{x'_d}\cdot(e'_q - v_j\cos(\delta - \theta_j))], \quad (26)$$

where the variables are defined in [1]. In (26), dynamics of e'_q is controlled by the field excitation voltage E_{FD},

which is the output of a conventional exciter, and the proposed NROC is to be attached to control the exciter input. The generator powers always satisfy the algebraic power balance constraint:

$$P_{ei}(e'_q, v_i, \delta, \theta_i) + P_{li}(v_i, \theta_i) = P_{Ni}(\vec{v}, \vec{\theta}),$$
$$Q_{ei}(e'_q, v_i, \delta, \theta_i) + Q_{li}(v_i, \theta_i) = Q_{Ni}(\vec{v}, \vec{\theta}), \quad \text{for } i = 1, 2 \qquad (27)$$

where P_{ei} and Q_{ei} are, respectively, the real and reactive power of the i-th generator, P_{Ni} and Q_{Ni} are the net powers injected to the i-th bus of the networks, and P_{li} and Q_{li} are the local loads on the i-th bus. The loads can be modeled as nonlinear functions of system variables. The real and reactive powers are then presented as;

$$P_{ei}(e'_q, v_i, \delta, \theta_i) = \frac{e'_q v_i}{x'_d}\sin(\delta - \theta_i) + \frac{v_i^2(x'_d - x_q)}{2x'_d x_q}\sin(2(\delta - \theta_i))$$
$$Q_{ei}(e'_q, v_i, \delta, \theta_i) = \frac{e'_q v_i \cos(\delta - \theta_i)}{x'_d} - \frac{v_i^2(x_q \cos^2(\delta - \theta_i) + x'_q \sin^2(\delta - \theta_i))}{2x'_d x_q} \qquad (28)$$

$$P_{li}(v_i, \theta_i) = P_{l0} \cdot \left(\frac{v_i}{v_{i0}}\right)^{\alpha_P} \cdot (1 + \beta_P \cdot \Delta f_i),$$
$$Q_{li}(v_i, \theta_i) = Q_{l0} \cdot \left(\frac{v_i}{v_{i0}}\right)^{\alpha_Q} \cdot (1 + \beta_Q \cdot \Delta f_i), \qquad (29)$$

$$P_{Ni}(\vec{v}, \vec{\theta}) = \sum_{k=1}^{2} v_i v_k (g_{ik}\cos(\theta_i - \theta_k) + b_{ik}\sin(\theta_i - \theta_k)),$$
$$Q_{Ni}(\vec{v}, \vec{\theta}) = \sum_{k=1}^{2} v_i v_k (g_{ik}\sin(\theta_i - \theta_k) - b_{ik}\cos(\theta_i - \theta_k)), \qquad (30)$$

where the parameters α, β and g_{ik}, b_{ik} are given to represent the load characteristics and the line admittance from i-th bus to k-th bus, respectively.

Typical IEEE governor and turbine models [14] are used; TGOV1 for the first generator and IEEEG2 for the second generator in the study power system. The IEEE exciter and voltage regulator model EXST1 is used for both generators.

3.2. Design of NROC

A paradigm for the neural networks in NROC is chosen by trial and error: IDNN and OCNN have one hidden layer with 40 nodes each. In IDNN, the input layer has seven nodes: four for output history, and three for input history. The corresponding OCNN has seven nodes in the input layer: four for output history, two for input history, and one for reference output.

The proper choice of a sampling period is very important. In this case, the frequency band of the plant output in the low frequency oscillation is about 1~2 [Hz]. Training patterns for IDNN are obtained with time step size of 0.04 [sec] in the simulation of the power system. This sampling time allows at least twenty sample points in one cycle of low-frequency oscillation whose frequency band is less than 1.25 [Hz]. It takes about 20 minutes in an IBM-PC 486 for the training of IDNN.

3.3. Simulation and Discussion

Two cases of disturbances are considered which are typical in a real power system operation: one is a line fault and the other is an abrupt load level change. Two operating conditions for the unit one are considered, normal loading condition for 0.8 [p.u.] and heavy loading condition for 1.0 [p.u.] .

Power system is controlled with three different options: 1) there is only the conventional voltage regulator; 2) the conventional voltage regulator with the conventional PSS (STAB4 [14]); and 3) the conventional voltage regulator with the proposed NROC.

(1) Case 1- Line Fault: In this case the tie-line admittance between the two generators changes from 100% to 75% at t=1.2 seconds undergoing a transient state when the 25% of line admittance is grounded at half point and cleared at t=1.85 second. This fault imitates one line outage when the tie line consists of parallel four line elements in Fig. 6. Fig. 7 , Fig. 8 and Fig. 9 show trajectories of the speed deviation and terminal voltage deviation of the first generator due to the line fault when the power system is in the normal loading and heavy loading conditions. Both the conventional PSS and the proposed NROC effectively damp the oscillation of voltage deviation. However, it is shown that the initial swing of voltage deviation with conventional PSS is larger than that with proposed NROC.

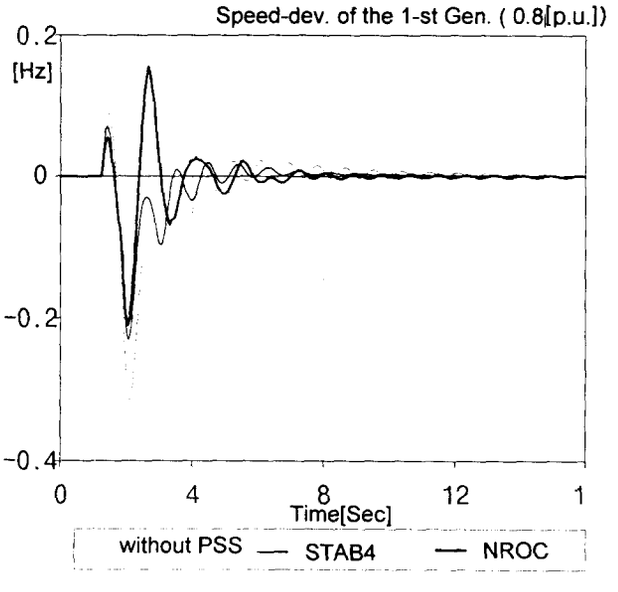

Fig. 7 The speed deviation of the first generator due to the line fault under the normal loading condition.

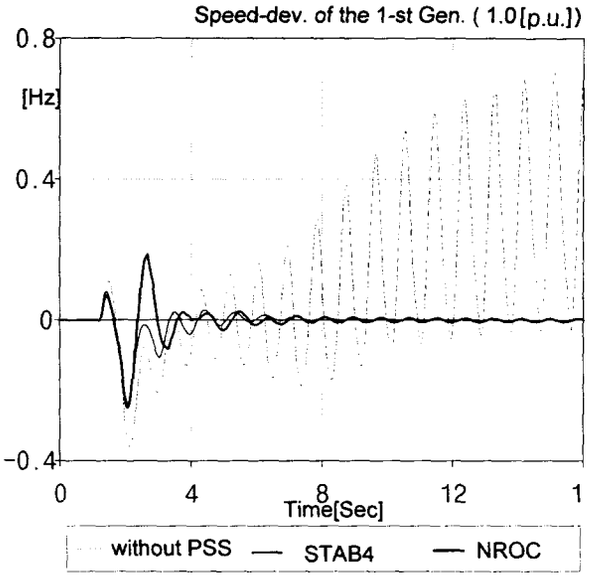

Fig. 8 The speed deviation of the first generator due to the line fault under heavy loading condition.

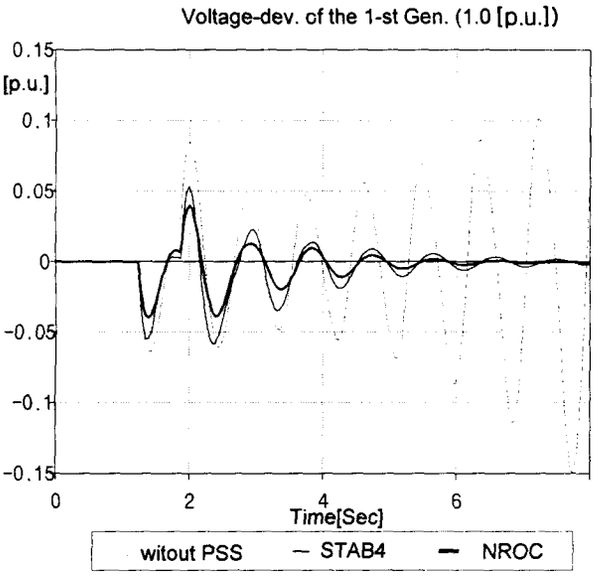

Fig. 9 The voltage deviation of the first generator due to the line fault under heavy loading condition.

(2) Case 2- Load changes: This simulates a case of disturbance when there is 5% stepwise changes. The changes are taken place at t=1.2 [sec] with 5% increase, at t=3.7[sec] with 5% decrease, at t=5[sec] with 5% decrease, and at t=6.3[sec] with 5% increase. Fig. 10 , Fig. 11 and Fig. 12 show the trajectories for the speed deviation and the terminal voltage deviation of the first generator due to load change disturbance when the power system is in the normal loading and in the heavy loading conditions. The conventional PSS keeps the oscillation small but have a large initial peak in the speed deviation. Moreover, It generates very large voltage deviation. However, the proposed NROC keeps the oscillation small with small peaks. It is shown that the performance of the NROC is better than the conventional PSS in the case of load change disturbance.

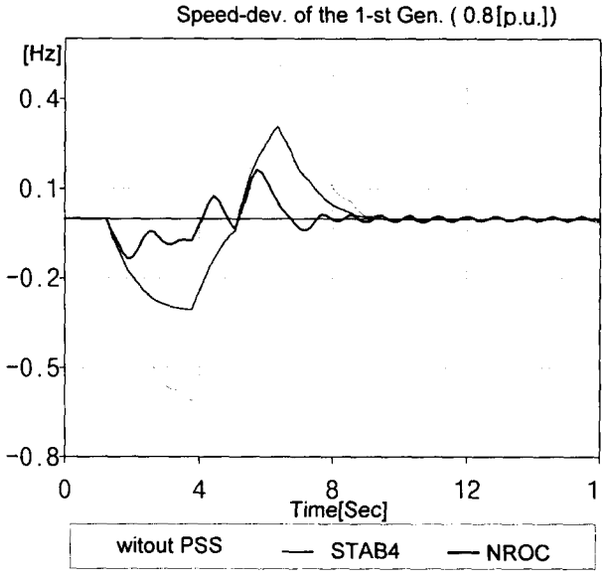

Fig. 10 The speed deviation of the first generator due to the load change under the normal loading condition.

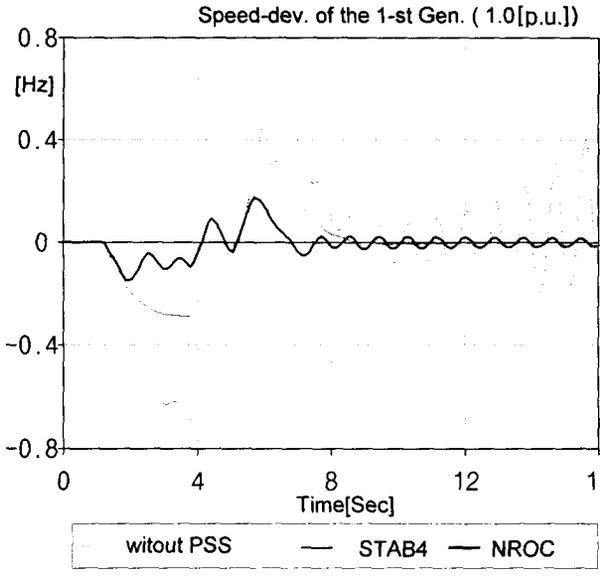

Fig. 11 The speed deviation of the first generator due to the load change under the heavy loading condition.

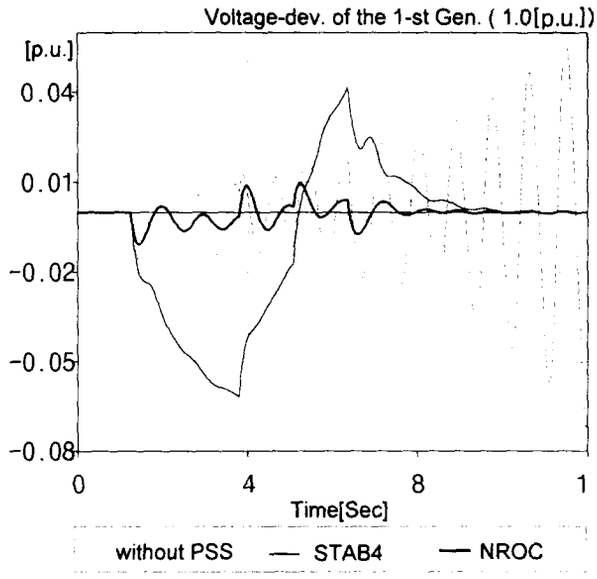

Fig. 12 The voltage deviation of the first generator due to the load change under heavy loading condition.

4. Conclusions

This study considers an application of neural networks to the receding horizon optimal control (NROC) problem for power systems. A decentralized approach for local system control by neural networks was proposed in which the complex power system was to be controlled by local system controllers that need only the local information. For each local system, a receding horizon optimal control law was implemented by two artificial neural networks: Identification Neural Network (IDNN) and Optimal Control Neural Network (OCNN). In training OCNN, the Generalized Backpropagation Through Time (GBTT) algorithm was used to minimize a quadratic cost function. The proposed NROC was applied to a power system to improve the damping of low-frequency oscillation. The proposed NROC demonstrated a favorable control performance compared to a conventional power system stabilizer for various operating conditions and fault types.

Reference

[1]Yao-nan Yu, Electric Power System Dynamics, Academic Press, New York, pp. 114-118, 1983.

[2]F. P. deMello, C.A.Concordia, "Concept of Synchronous Machine Stability as Affected by Excitation Control", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-103, pp. 316-319, 1969.

[3]Olle I. Elgerd, Electric Energy Systems Theory, MaGraw-Hill Book Company, New York. 1982, pp. 340-358.

[4]F. P. deMello, P. J. Nolan, T. F. Laskowoski, and J. M. Undrill, " Coordinate application of stabilizers in multimachine power systems", IEEE Trans. on Power Apparatus and Systems, pp. 892-901, May/June 1980.

[5]K. T. Law, D. J. Hill and N. R. Godfrey, "Robust controller structure for coordinate power system voltage regulator and stabilizer design", IEEE. Trans. on Control Systems Technology, Vol. 2, No. 3, pp. 220-232, September 1994.

[6]Q. H. Wu, B.W. Hogg, and G.W. Irwin, " A neural network regulator for turbo generators ", IEEE Trans. on Neural Networks, Vol. 3, No. 1, Jan. 1992.

[7]D. C. Kennedy and V. H. Quintana, "Neural network regulators for synchronous machines ", Proc. American Control Conference. Vol. 5, pp. 131-136, 1993.

[8]C. C. Ku and K. Y. Lee, " Diagonal recurrent neural network for dynamic system control ", IEEE Trans. on Neural Networks, Vol. 6, pp. 144-156, Jan. 1995.

[9]Y. Iiguni and H. Sakai, A nonlinear regulator design in the presence of system uncertainties using multilayered neural networks, IEEE Trans. on Neural Networks, pp. 410-417, vol. 3, July 1991.

[10] W. T. Miller, R. S. Sutton and P. J. Werbos, Neural Networks for Control, The MIT Press, 1990, pp. 28-65.

[11] P. J. Werbos, Backpropagation through time: What it does and how to do it, Proc of IEEE, pp.1550-1560, vol. 78, no. 10, Oct. 1990.

[12] Y. M. Park, M. S. Choi and K. Y. Lee, "A neural network-based power system stabilizer using power flow characteristics, IEEE Trans. on Energy Conversion, Vol. 11, No. 2, pp. 435-441, 1996.

[13] M. I. Jordan and D. E. Rumelhart, Forward models: Supervised learning with a distal teacher, Cognitive Science, vol. 16, pp. 307-354, 1992.

[14]T.E. Kostyniack, PSS/E Program Operation Manual, Power Technology Inc., October 31, 1983.

저 자 소 개

최 면 송(崔 勉 松)

1967년 4월생. 1989년 서울대 공대 전기공학과 졸업. 1991년 동 대학원 전기공학과 졸업(석사). 1996년 동 전기공학과 졸업(공박). 1995년 Pensylvania State Univ. 방문 연구원. 1992년 기초전력공학 공동연구소 전임연구원. 현재 명지대학교 공대 전기정보제어공학부 조교수.

Tel : +82-335-336-3290, Fax : +82-335-321-0271

E-mail : mschoi@wh.myongji.ac.kr