01 Jan 2003

Adaptive-critic-Based Optimal Neurocontrol for Synchronous Generators in a Power System using MLP/RBF Neural Networks

Jung-Wook Park

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Ronald G. Harley

Recommended Citation

J. Park et al., "Adaptive-critic-Based Optimal Neurocontrol for Synchronous Generators in a Power System using MLP/RBF Neural Networks," *IEEE Transactions on Industry Applications*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2003.
The definitive version is available at https://doi.org/10.1109/TIA.2003.816493

Adaptive-Critic-Based Optimal Neurocontrol for Synchronous Generators in a Power System Using MLP/RBF Neural Networks

Jung-Wook Park, *Member, IEEE*, Ronald G. Harley, *Fellow, IEEE*, and Ganesh Kumar Venayagamoorthy, *Senior Member, IEEE*

***Abstract*—This paper presents a novel optimal neurocontroller that replaces the conventional controller (CONVC), which consists of the automatic voltage regulator and turbine governor, to control a synchronous generator in a power system using a multilayer perceptron neural network (MLPN) and a radial basis function neural network (RBFN). The heuristic dynamic programming (HDP) based on the adaptive critic design technique is used for the design of the neurocontroller. The performance of the MLPN-based HDP neurocontroller (MHDPC) is compared with the RBFN-based HDP neurocontroller (RHDPC) for small as well as large disturbances to a power system, and they are in turn compared with the CONVC. Simulation results are presented to show that the proposed neurocontrollers provide stable convergence with robustness, and the RHDPC outperforms the MHDPC and CONVC in terms of system damping and transient improvement.**

***Index Terms*—Adaptive critic design (ACD), heuristic dynamic programming (HDP), multiplayer perceptron network (MLPN), optimal neurocontroller, radial basis function network (RBFN), synchronous generator.**

INTRODUCTION

A SYNCHRONOUS generator in a power system is a nonlinear fast-acting multiple-input–multiple-output (MIMO) device [1], [2]. Conventional linear controllers (CONVCs) for the synchronous generator consist of the automatic voltage regulator (AVR) to maintain constant terminal voltage and the turbine governor to maintain constant speed and power at some set point. They are designed to control, in some optimal fashion, the generator around one particular operating point; and at any other point the generator's damping performance is degraded. As a result, sufficient margins of safety are included in the generator maximum performance envelope in order to allow for degraded damping when transients occur. Due to a synchronous generator's wide operating range, its complex dynamics [3], [4], its transient performance, its nonlinearities, and a changing system configuration, it cannot be accurately modeled as a linear device.

Artificial neural networks (ANNs) offer an alternative for the CONVC as nonlinear adaptive controllers. Researchers in the field of electrical power engineering have until now used two different types of neural networks, namely, a multilayer perceptron network (MLPN), or a radial basis function network (RBFN), both in single and multimachine power system studies [3]–[7]. Proponents of each type of neural network have claimed advantages for their choice of ANN, without comparing the performance of the other type for the same study. The applications of ANNs in the power industry are expanding, and at this stage there is no authoritative fair comparison between the MLPN and the RBFN [8], [9].

The authors' earlier work comparing performance of the above two ANNs for the *indirect adaptive control* of the synchronous generator showed that the RBFN-based neurocontroller improves the system damping and transient performance more effectively and adaptively than the MLPN-based neurocontroller [9]. Also, the different damping properties of the above two neurocontrollers and the stability issue during transients were analyzed and proven based on the Lyapunov direct method. However, one cannot avoid the possibility of instability during steady state at the various different operating conditions when using the indirect adaptive control based on the gradient descent algorithm. To overcome the issue of instability and provide strong robustness for the controller, the adaptive critic design (ACD) technique [10]–[16] for the optimal control has been recently developed where the ANNs are used to identify and control the process. Without the highly extensive computational efforts and difficult mathematical analyzes required by using the dynamic programming (DP) in classical optimal control theory [17]–[20], the ACD technique provides an effective method to construct an optimal and robust feedback controller by exploiting backpropagation for the calculation of all the derivatives of a target quantity [10], [21] in order to minimize/maximize the heuristic cost-to-go approximation.

In this paper, the background of adaptive critic designs with relation to optimal control theory, and a general description

Paper MSDAD-A 03–07, presented at the 2002 Industry Applications Society Annual Meeting, Pittsburgh, PA, October 13–18, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Industrial Automation and Control Committee of the IEEE Industry Applications Society. Manuscript submitted for review October 15, 2002 and released for publication June 2, 2003. This work was supported by the National Science Foundation under Grant ECS-0080764.

J.-W. Park was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0254 USA. He is now with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: jungwookpark@ieee.org).

R. G. Harley is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0254 USA (e-mail: rharley@ece.gatech.edu).

G. K. Venayagamoorthy is with the Department of Electrical and Computer Engineering, University of Missouri, Rolla, MO 65409-0249 USA (e-mail: ganeshv@ece.umr.edu).

Digital Object Identifier 10.1109/TIA.2003.816493

for the MLPN/RBFN, are presented. Based on the heuristic dynamic programming (HDP), which is a class of ACD family, the two optimal neurocontrollers using the MLPN and RBFN (called MHDPC and RHDPC, respectively) are designed. In addition, their performances for the on-line control of synchronous generators in an electric power grid (multimachine power system as well as single machine connected to an infinite bus (SMIB) system) are illustrated and compared with several case studies by time-domain simulation.

I. Background on ACDs and Description of MLPN/RBFN

How can the ANNs be applied to handle optimal control theory at the level of human intelligence? As one approach for solution of this problem, this section describes the framework behind the adaptive critic neural network based design for solving optimal control problems such as in the design of an optimal controller for the nonlinear synchronous generator in a power system network.

A. Optimal Control Problem

The continuous-time dynamic systems to be considered in finite state problem are as follows:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t), \qquad 0 \le t \le T \tag{1}$$

where $\mathbf{x}(t) \in \mathbf{R^n}$ is the state vector at time t, $\dot{\mathbf{x}}(t) \in \mathbf{R^n}$ is the vector of first-order time derivatives of the states at time t, $\mathbf{u}(t) \in \overline{\mathbf{U}} \subset \mathbf{R^m}$ is the control vector at time t, $\overline{\mathbf{U}}$ is the control constraint set, and T is the terminal time. It is assumed that the system function f is continuously differentiable with respect to $\mathbf{x}$ and is continuous with respect to $\mathbf{u}$. The *admissible* control functions, which are called *control trajectories*, are the piecewise continuous functions $\{\mathbf{u}(t)|t \in [0,T]\}$ with $\mathbf{u}(t) \in \overline{\mathbf{U}}$ for all $t \in [0,T]$. The task to be performed is to transfer the state from a known initial state $\mathbf{x}(0)$ to a specified final state $\mathbf{x}(T)$ in the target set of the state space. The task is implicitly specified by the performance criteria $\mathbf{J}(t,\mathbf{x})$, namely, optimal cost-to-go function at time t and state $\mathbf{x}$.

$$\mathbf{J}(t,\mathbf{x}) = h(\mathbf{x}(T)) + \int_0^T g(\mathbf{x}(t), \mathbf{u}(t))\, dt \tag{2}$$

where h is the cost or penalty associated with the error in the terminal state at time T, and g is the cost function associated with transient state errors and control effort. Then, the *optimal control problem* can be considered as finding the $\mathbf{u} \in \overline{\mathbf{U}}$ to minimize the total cost function $\mathbf{J}$ in (2) subject to the dynamic system constraints in (1) and all initial and terminal boundary conditions that maybe specified.

The Hamilton–Jacobi–Bellman (HJB) equation in (3), which is analogous with the DP algorithm, gives the solution to determine optimal controls in offline by deriving a partial differential equation satisfied by the function $\mathbf{J}$ with assumed differentiability as the sufficient condition

$$\begin{aligned} 0 &= \min_{\mathbf{u}\in\overline{\mathbf{U}}} [g(\mathbf{x},\mathbf{u}) + \nabla_t \mathbf{J}(t,\mathbf{x}) + \nabla_{\mathbf{x}} \mathbf{J}(t,\mathbf{x})' f(\mathbf{x},\mathbf{u})], \\ &\quad \text{for all } t, \mathbf{x} \\ \mathbf{J}(T,\mathbf{x}) &= h(\mathbf{x}), \quad \text{for boundary condition} \end{aligned} \tag{3}$$

where ∇_t denotes partial derivatives with respect to t and $\nabla_{\mathbf{x}}$ denotes an n-dimensional vector of partial derivatives with respect to $\mathbf{x}$. The HJB equation in (3) requires $\nabla_{\mathbf{x}}\mathbf{J}$ to be known at all values of $\mathbf{x}$ and t. However, the value of $\nabla_{\mathbf{x}}\mathbf{J}$ is possible to be known at only one value of $\mathbf{x}$ for each t given in (4), and therefore $\nabla_{\mathbf{x}}\mathbf{J}(t,\mathbf{x}^*(t))$ can be calculated more easily than the HJB equation. This is known as the *adjoint equation* for the optimal state trajectory

$$\begin{aligned} \mathbf{u}^*(t) = \arg\min_{\mathbf{u}\in\overline{\mathbf{U}}} [&g(\mathbf{x}^*(t),\mathbf{u}) + \nabla_t \mathbf{J}(t,\mathbf{x}^*(t)) \\ &+ \nabla_{\mathbf{x}} \mathbf{J}(t,\mathbf{x}^*(t))' f(\mathbf{x}^*(t),\mathbf{u})] \end{aligned} \tag{4}$$

where $\mathbf{u}^*(t)$ is the optimal control trajectory with corresponding state trajectory $\mathbf{x}^*(t)$ for all $t \in [0,T]$. Then, the generalization of the calculus of variations known as the *Pontryagin's Minimum Principle* is summarized as follows:

$$p_0(t) = \nabla_t \mathbf{J}(t,\mathbf{x}^*(t)), \quad \dot{p}_0(t) = 0 \tag{5}$$

$$p(t) = \nabla_{\mathbf{x}} \mathbf{J}(t,\mathbf{x}^*(t)), \quad p(T) = \nabla h(\mathbf{x}^*(T)) \tag{6}$$

$$\dot{p}(t) = -\nabla_{\mathbf{x}} f(\mathbf{x}^*(t),\mathbf{u}^*(t))\, p(t) - \nabla_{\mathbf{x}} g(\mathbf{x}^*(t),\mathbf{u}^*(t)) \tag{7}$$

$$\begin{aligned} \mathbf{u}^*(t) = \arg\min_{\mathbf{u}\in\overline{\mathbf{U}}} &[g(\mathbf{x}^*(t),\mathbf{u}) + p(t)' f(\mathbf{x}^*(t),\mathbf{u})], \\ &\text{for all } t \in [0,T]. \end{aligned} \tag{8}$$

B. ACDs

For constant coefficient systems of which the operating time is very long, especially in real-time operation, it is often justifiable to assume that the terminal time is infinitely far in the future, which is called *infinite horizon problem*. This approximation may cause little or no degradation in optimality because the optimal time-varying gains such as the *costate equation* in (7) approach constant values in a few time stages. Thus, the optimal gains are constant for most of the operating period.

The continuous-time cost function $\mathbf{J}$ in (2) can be reformulated as the total cost-to-go function of the infinite horizon problem in (9) for the discrete-time dynamic system

$$\mathbf{J}_\pi(\mathbf{x}_0) = \sum_{k=0}^{\infty} \gamma^k g(\mathbf{x}(k), \mathbf{u}(k)) \tag{9}$$

where k is a discrete-time index at each step, $\mathbf{J}_\pi(\mathbf{x}_0)$ denotes the cost associated with an initial state $\mathbf{x}_0$, and a control policy $\pi = \{\mathbf{u}_0, \mathbf{u}_1, \ldots\}$, and γ is the discount factor $(0 < \gamma < 1)$. The Bellman equation using the DP in (10) is iteratively solved at each time step to find the optimal control $\mathbf{u}^*$ corresponding to the optimal cost-to-go function $\mathbf{J}^*$ in (11)

$$\begin{aligned} \mathbf{J}_{k+1}(\mathbf{x}) &= \min_{\mathbf{u}\in\overline{\mathbf{U}}} [g(\mathbf{x},\mathbf{u}) + \gamma \mathbf{J}_k(f(\mathbf{x},\mathbf{u}))], \\ &\quad k = 0, 1, \ldots \\ \mathbf{J}_0(\mathbf{x}) &= 0, \qquad \text{for all } \mathbf{x} \end{aligned} \tag{10}$$

$$\mathbf{J}^*(\mathbf{x}) = \lim_{k\to\infty} (T^k \mathbf{J}_0)(\mathbf{x}), \qquad \text{for all } \mathbf{x} \in S \tag{11}$$

where $(T\mathbf{J})(\mathbf{x})$ is a DP mapping function defined in (12) on the state space S for any function $\mathbf{J} : S \to \mathbf{R}$

$$(T\mathbf{J})(\mathbf{x}) = \min_{\mathbf{u}\in\overline{\mathbf{U}}} [g(\mathbf{x},\mathbf{u}) + \gamma \mathbf{J}(f(\mathbf{x},\mathbf{u}))]. \tag{12}$$

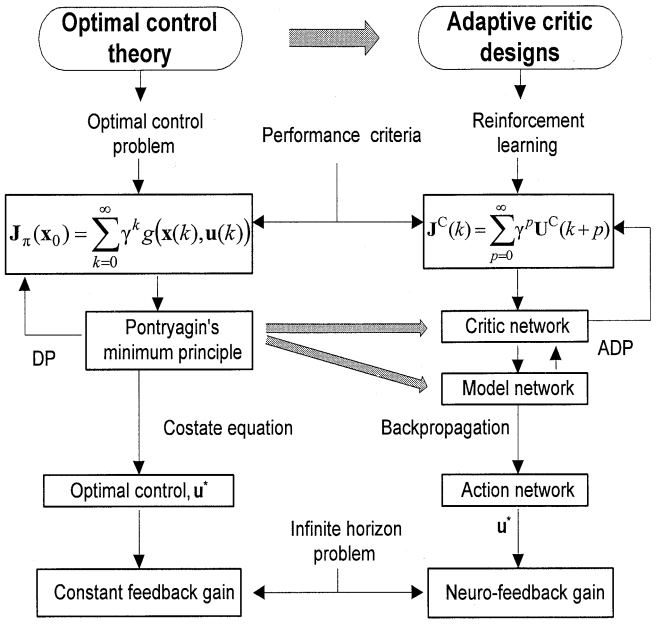


Fig. 1. Optimal controller design for infinite horizon problem: optimal control theory versus ACDs.

However, the above optimal control theory cannot readily be applied to deal with a large number of control variables of a nonlinear dynamic system such as synchronous generators in a multimachine power system. Also, the classical DP algorithm requires extensive computations and memory, known as the so-called "curse of dimensionality." To overcome this problem, several alternative methods have been proposed depending on manner in which the cost-to-go approximation is selected, and one of those approaches is the neuro-dynamic programming (NDP) using some form of "least-squares fit" for the heuristic cost-to-go approximation [19]. ACDs technique can be classified as one of the NDP families using function approximator such as ANN architectures. In other words, this novel technique provides an alternative approach to handle the optimal control problem combining concepts of the *reinforcement learning* and the *approximate dynamic programming* (ADP). The illustration relating the optimal control theory to the ACD is shown in Fig. 1. The ACD described in this paper uses three different types of neural networks, namely, the critic, model, and action. In Fig. 1, the utility function or cost function $\mathbf{U}^{\mathrm{C}}$ to be minimized is called "reinforcement" in the ACD. In applying the ANNs to reinforcement learning, there are two major steps to account for the link between *present* actions and *future* consequences for the ACD technique [10]. The first step is to build a "model" network for identifying the plant, and use backpropagation to calculate the derivatives of future utility with respect to present actions through the model network. The second step is to adapt a "critic" network, a special network that outputs an estimate of the total future value of $\mathbf{U}^{\mathrm{C}}$, which will arise from the present and past states and the control information. From the viewpoint of optimal control theory, the backpropagation is the same as the first-order calculus of variations to calculate the costate equation in (7) by taking the derivatives.

Likewise in the adaptive critic, $\mathbf{J}^{\mathrm{C}}(k)$ can be derived using the ADP. In other words, the critic network learns to approximate the heuristic cost-to-go function in (13)

$$\mathbf{J}^{\mathrm{C}}(k)=\sum_{p=0}^{\infty}\gamma^p\mathbf{U}^{\mathrm{C}}(k+p) \tag{13}$$

where γ is the discount factor $(0<\gamma<1)$.

After minimizing the $\mathbf{J}^{\mathrm{C}}$ in (13) by the critic network, the "action" network is trained with the estimated output backpropagated from the critic network to obtain the converged weight for the optimal control $\mathbf{u}^*$.

The design and training of the model, critic, and action networks are described in Section III together with their mathematical analyses.

C. MLPN

In this paper, the MLPN consists of three layers of neurons [input, hidden, and output layer as shown in Fig. 2(a)] interconnected by the weight vectors, $\boldsymbol{W}$ and $\boldsymbol{V}$.

The weights of the MLPN are adjusted/trained using the gradient-descent-based backpropagation algorithm. The activation function for neurons in the hidden layer is given by the following sigmoidal function:

$$\Phi(x)=\frac{1}{1+\exp(-x)}. \tag{14}$$

The output layer neurons are formed by the inner products between the nonlinear regression vector from the hidden layer and the output weight matrix, $\boldsymbol{V}$. Generally, the MLPN starts with random initial values for its weights, and then computes a one-pass backpropagation algorithm at each time step k, which consists of a forward pass propagating the input vector through the network layer by layer, and a backward pass to update the weights by the gradient descent rule. By trial and error, 14, 10, and 13 neurons in the hidden layer for the model, action, and critic network, respectively, are optimally chosen for this study. These values depend on a tradeoff between convergence speed and accuracy.

D. RBFN

Like the MLPN, the RBFN also consist of three layers [Fig. 2(b)]. However, the input values are each assigned to a node in the input layer and passed directly to the hidden layer without weights. The hidden layer nodes are called RBF units, determined by a parameter vector called *center* and a scalar called *width*. The gaussian density function is used as an activation function for the hidden neurons in Fig. 2(b).

The overall input–output mapping equation of the RBFN is as follows:

$$y_i=b_i+\sum_{j=1}^{h}v_{ji}\exp\left(-\frac{\|\mathbf{X}-\mathbf{C}_j\|^2}{\beta_j^2}\right) \tag{15}$$

where $\mathbf{X}$ is the input vector, $\mathbf{C}_j$ is the jth center of RBF unit in the hidden layer, h is the number of RBF units, b_i and v_{ji} are the bias term and the weight between the hidden and output layers, respectively, and y_i is the ith output. Once the centers of RBF

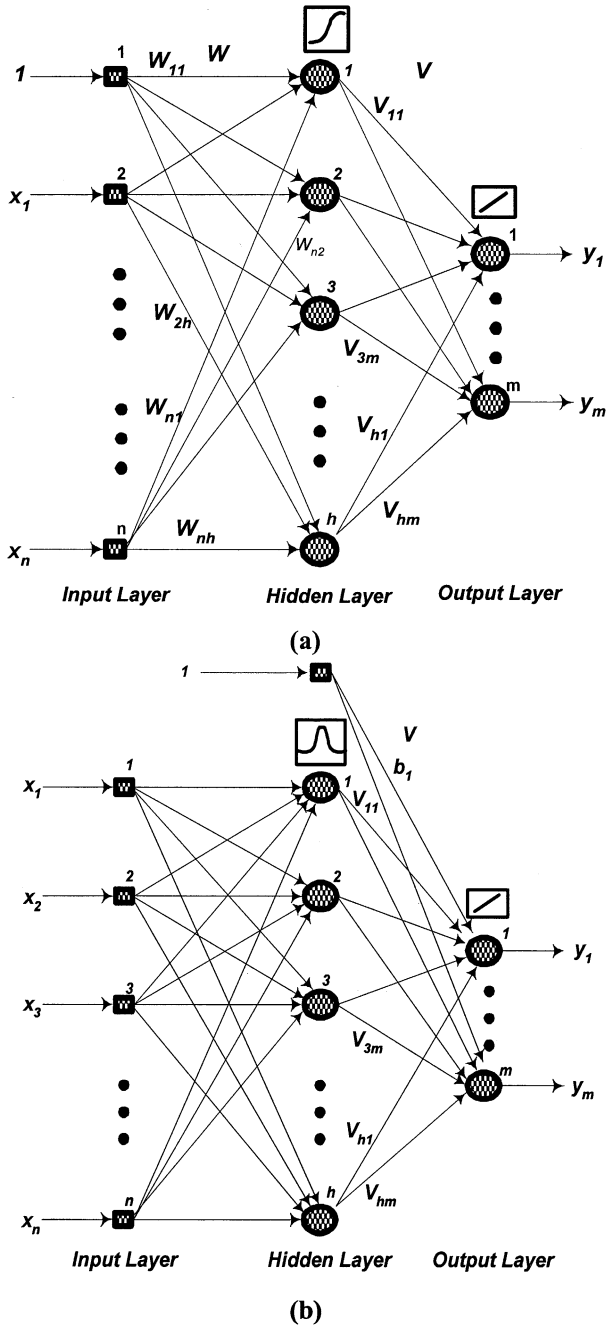


Fig. 2. Feedforward ANNs. (a) MLPN. (b) RBFN.

units are established, the width of the ith center in the hidden layer is calculated by (16)

$$\beta_i = \left[\frac{1}{h} \sum_{j=1}^{h} \sum_{k=1}^{n} (\|\mathbf{c}_{ki} - \mathbf{c}_{kj}\|) \right]^{\frac{1}{2}} \tag{16}$$

where $\mathbf{c}_{ki}$ and $\mathbf{c}_{kj}$ are the kth value of the center of ith and jth RBF units. In (15) and (16), $\|\cdot\|$ represents the Euclidean norm. There are four different ways for input–output mapping using the RBFN, depending on how the input data is fed to the network: [22].

- batch mode clustering of centers and batch mode gradient descent for linear weights;
- batch mode clustering of centers and pattern mode gradient descent for linear weights;
- pattern mode clustering of centers and pattern mode gradient descent for linear weights;
- pattern mode clustering of centers and batch mode gradient descent for linear weights.

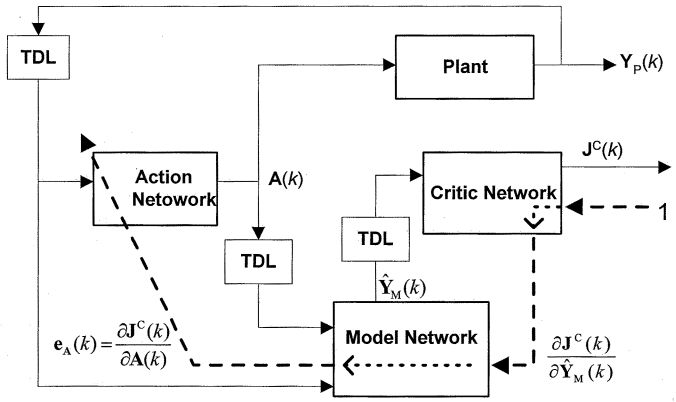


Fig. 3. Structure of the HDP configuration: action adaptation in HDP.

To avoid the extensive computational complexity during training, the batch mode k-means clustering algorithm for centers is initially calculated for the centers of the RBF unit. Thereafter, the pattern mode least-mean-square (LMS) algorithm is calculated to update the output linear weights [8], [9]. By trial and error, 12 neurons for the model network and six neurons for the action and critic networks in the hidden layer are optimally chosen for this study.

II. HDP Neurocontroller

The structure of the HDP configuration is shown in Fig. 3. The critic network is connected to the action network through the model network, and is therefore called a model-dependent critic design. All these three different ANNs are described in the following sections.

In the literature so far, only the MLPN has been reported for the implementation of the ACD. In this paper, the performance of an optimal neurocontroller based on the HDP using the MLPN and RBFN is compared. The HDP is the simplest of the ACDs, and it provides a framework to compare the performance of two optimal neurocontrollers (MHDPC/RDHPC).

A. Plant Modeling

The synchronous generator, turbine, exciter, and transmission system connected to an infinite bus in Fig. 4 form the plant (dotted block in Fig. 4). that has to be controlled. Nonlinear equations are used to describe and simulate the dynamics of the plant in order to generate the data for the optimal neurocontrollers. On a physical plant, this data would be measured. The generator $(\mathbf{G})$ with its damper windings is described by the seventh order d–q axis set of equations with the generator current, speed, and rotor angle as the state variables [1], [2]. In the plant, P_t and Q_t are the real and reactive power at the generator terminal, respectively, Z_e is the transmission line impedance, P_m is the mechanical input power to the generator, V_{fd} is the exciter field voltage, V_b is the infinite bus voltage, $\Delta\omega$ is the

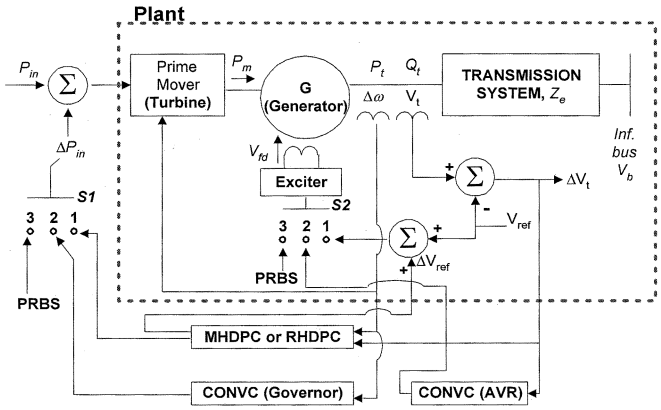


Fig. 4. Plant model used for the control of a synchronous generator connected to an infinite bus.

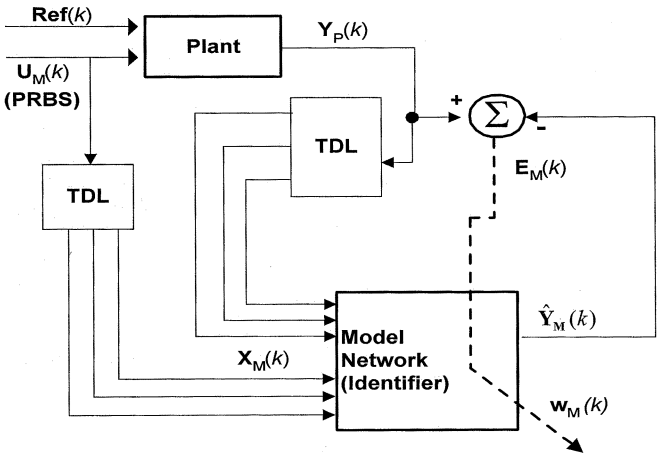


Fig. 5. Training of the model network using the backpropagation algorithm.

speed deviation, ΔV_t is the terminal voltage deviation, V_t is the terminal voltage, $\Delta V_{\rm ref}$ is the reference voltage deviation, $V_{\rm ref}$ is the reference voltage, $\Delta P_{\rm in}$ is the input power deviation, and $P_{\rm in}$ is the turbine input power.

The positions 1 and 2 of switches $S1$ and $S2$ in Fig. 4 determine whether the optimal neurocontroller (MHDPC or RHDPC), or the CONVC consisting of governor and AVR, is controlling the plant. Block diagrams and data for the CONVC as well as the mathematical expression of transmission system appear in the Appendix [5].

B. Design and Training of the Model Network

Fig. 5 illustrates how the model network (identifier) is trained online to identify the dynamics of the plant in Fig. 4. At this stage, there is no action network or critic network or CONVC present. Switches $S1$ and $S2$ in Fig. 4 are in position 3. The nonlinear autoregressive moving average with exogenous inputs (NARMAX) model is used as the benchmark model for online identification [8].

The input vector $\mathbf{U}_{\rm M}(k)$ consists of the turbine input power deviation $(\Delta P_{\rm in})$ and exciter input voltage deviation $(\Delta V_{\rm ref})$, that is, $\mathbf{U}_{\rm M}(k) = [\Delta P_{\rm in}(k),\ \Delta V_{\rm ref}(k)]$, and is fed into the plant with the vector, $\mathbf{Ref}(t) = [P_{\rm in}(k),\ V_{\rm ref}(k)]$. The input signals of $\mathbf{U}_{\rm M}(k)$ are 5-Hz pseudorandom binary signals (PRBSs)

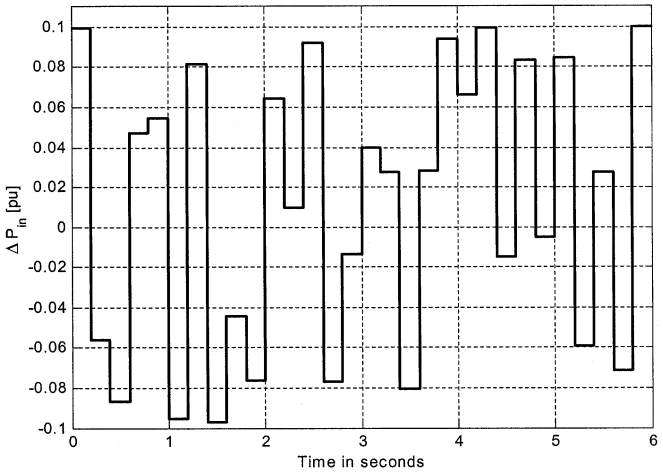


Fig. 6. Input power deviation PRBS applied to the turbine.

within $\pm 10\%$ of the magnitude of the reference values of the turbine input power $(P_{\rm in})$ and exciter input voltage $(V_{\rm ref})$ at a particular plant operating point. As an example, the PRBS of $\Delta P_{\rm in}$ is shown in Fig. 6.

The output vector of the plant $\mathbf{Y}_{\rm P}(k)$ consists of the speed deviation $(\Delta\omega)$ and terminal voltage deviation (ΔV_t), that is, $\mathbf{Y}_{\rm P}(k) = [\Delta\omega(k),\ \Delta V_t(k)]$. The model network output $\hat{\mathbf{Y}}_{\rm M}(k) = \hat{f}(\mathbf{X}_{\rm M}(k))$, where $\mathbf{X}_{\rm M}(k)$ is the input vector to the model network consisting of three time lags of system input and output, respectively,

$$\mathbf{X}_{\rm M}(k) = [[\mathbf{Y}_{\rm p}(\xi)\mathbf{U}_{\rm M}(\xi)]\,|\xi = \{k-1, k-2, k-3\}]^{\rm T}. \tag{17}$$

The residual vector $\mathbf{E}_{\rm M}(k)$ given in (18) is used for updating the model network's weights $\mathbf{W}_{\rm M}(k)$ during training by the backpropagation algorithm

$$\mathbf{E}_{\rm M}(k) = \mathbf{Y}_{\rm p}(k) - \hat{\mathbf{Y}}_{\rm M}(k). \tag{18}$$

This training is carried out at several different operating conditions within the stability limit of the synchronous generator until the training error has converged to a small value so that if training were to stop, and the weights fixed, then the neural network would continue to identify the plant correctly after changing the operating conditions. At this point, the model network has reached *global convergence*, and its weights $\mathbf{W}_{\rm M}$ are held fixed during the training of the critic and action networks. The steps of training for the critic and action networks are described in Section III-E below. The result for online identification of $\Delta\omega$, after the weights have been fixed at $\rm t = 0$ s, in Fig. 7, shows that both the MLPN- and RBFN-based model networks are able to correctly identify the dynamics of the plant.

The details of the training time and computational complexity to process the data by the MLPN- and RBFN-based identifiers, are shown in [8] and [9].

C. Critic Network

The critic network in the HDP approximates the function $\mathbf{J}^{\rm C}$ itself in (13). The configuration for training the critic network is

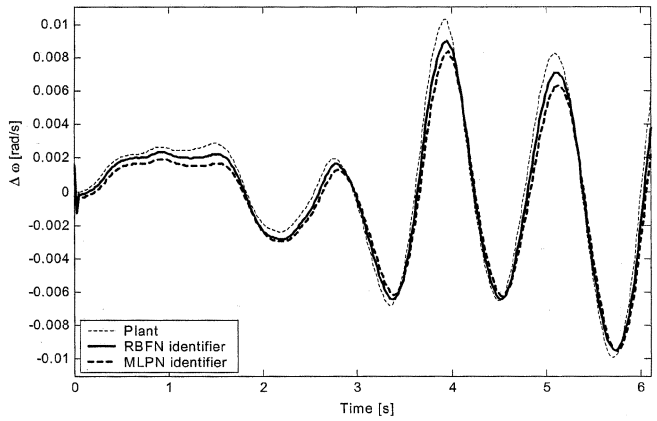


Fig. 7. Online training of the model network: speed deviation response.

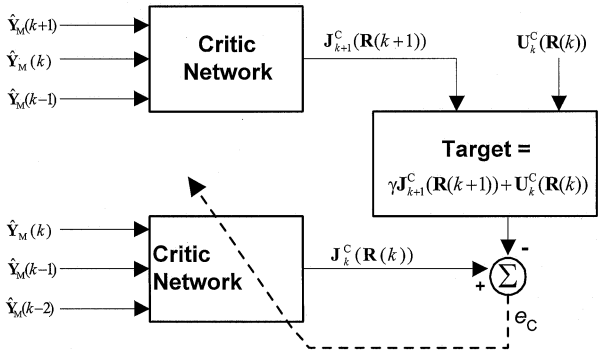


Fig. 8. Critic adaptation in HDP: the same critic network is shown for two consecutive times, $k+1$ and k. The critic's output $\mathbf{J}^{\mathrm{C}}_{k+1}(\mathbf{R}(k+1))$ at time $k+1$ is necessary for the ADP to generate a target signal $\gamma \mathbf{J}^{\mathrm{C}}_{k+1}(\mathbf{R}(k+1)) + \mathbf{U}^{\mathrm{C}}_{k}(\mathbf{R}(k))$ for training the critic network.

shown in Fig. 8. The Bellman equation in DP in (10) is implemented by the ADP using two critic networks. From (10), we get the following:

$$e_{\mathrm{DP}} = [g(\mathbf{x},\mathbf{u}) + \gamma \mathbf{J}_k(f(\mathbf{x},\mathbf{u}))] - \mathbf{J}_{k+1}(\mathbf{x}). \quad (19)$$

Note that the time indexing in (19) needs to be reversed for the problem discussed in this paper. In other words, the initial cost-to-function $\mathbf{J}^{\mathrm{C}}$ at time zero has a positive value α because the initial weights $\mathbf{W}_{\mathrm{C}}(0)$ of critic network are randomly chosen and the value of $\mathbf{J}^{\mathrm{C}}$ is kept minimizing as the time goes to an infinite. Therefore, the following error equation for the adaptation of critic network can be obtained:

$$e_{\mathrm{C}}(k) = \mathbf{J}^{\mathrm{C}}_{k}(\mathbf{R}(k)) - \gamma \mathbf{J}^{\mathrm{C}}_{k+1}(\mathbf{R}(k+1)) - \mathbf{U}^{\mathrm{C}}_{k}(\mathbf{R}(k)) \quad (20)$$

where $\mathbf{R}(k+1)$ and $\mathbf{R}(k)$ is a vector of observables of the plant, which are the output vectors from the model network in Fig. 5 at present and two consecutive past time stages.

Then, the critic network's weights $\mathbf{W}_{\mathrm{C}}$ are updated as follows:

$$\mathbf{W}_{\mathrm{C}}(k+1) = \mathbf{W}_{\mathrm{C}}(k) + \Delta \mathbf{W}_{\mathrm{C}}(k) \quad (21)$$

$$\Delta \mathbf{W}_{\mathrm{C}}(k) = -\eta_{\mathrm{C}} \cdot e_{\mathrm{C}}(k) \cdot \frac{\partial e_{\mathrm{C}}(k)}{\partial \mathbf{W}_{\mathrm{C}}(k)} \quad (22)$$

where η_{C} is the positive learning rate.

The training for critic network by the backpropagation algorithm is carried out until the value of $\mathbf{J}^{\mathrm{C}}$ is minimized as small as possible, which is almost zero. This adaptation process is considered as the *value iteration* in (12) to reach the optimal cost-to-go function $\mathbf{J}^{*}$ in (11) by the ADP provided from two critic neural networks.

D. Action Network

The input of the action network in Fig. 3 is the output vector of the plant, $\mathbf{Y}_{\mathrm{P}}$ and its two time-delayed values. The output of the action network is $\mathbf{A}(k) = [\Delta P_{\mathrm{in}}(k),\ \Delta V_{\mathrm{ref}}(k)]$.

The objective of the action network shown in Fig. 3 is to find the optimal control $\mathbf{u}^{*}$, as in (8), to minimize $\mathbf{J}^{\mathrm{C}}$ in the immediate future, thereby optimizing the overall cost expressed as a sum of all $\mathbf{U}^{\mathrm{C}}$ over the horizon of the problem in (13). This is achieved by training the action network with an error vector $\mathbf{e}_{\mathrm{A}}(\mathrm{k})$ in (23).

$$\mathbf{e}_{\mathrm{A}}(k) = \frac{\partial \mathbf{J}^{\mathrm{C}}(k)}{\partial \mathbf{A}(k)}. \quad (23)$$

The derivative of the cost function $\mathbf{J}^{\mathrm{C}}(k)$ with respect to $\mathbf{A}(k)$ in (23) is obtained by backpropagating $\partial \mathbf{J}^{\mathrm{C}}/\partial \mathbf{J}^{\mathrm{C}}$ (recall that the HDP approximates the function $\mathbf{J}^{\mathrm{C}}$ itself.) through the critic network and then through the pretrained model network to the action network. This gives $\partial \mathbf{J}^{\mathrm{C}}(k)/\partial \hat{\mathbf{Y}}_{\mathrm{M}}(k)$ and $\partial \mathbf{J}^{\mathrm{C}}(k)/\partial \mathbf{A}(k)$ in Fig. 3 for the weights $\mathbf{W}_{\mathrm{A}}(k)$ and the output vector $\mathbf{A}(k)$ of the action network. The expression for the weights' update in the action network is given in (24)

$$\Delta \mathbf{W}_{\mathrm{A}}(k) = -\eta_{\mathrm{A}} \cdot \mathbf{e}_{\mathrm{A}}(k) \cdot \frac{\partial \mathbf{e}_{\mathrm{A}}(k)}{\partial \mathbf{W}_{\mathrm{A}}(k)} \quad (24)$$

where η_{A} is the positive learning rate. The mathematical closed forms of $\partial \mathbf{J}^{\mathrm{C}}(k)/\partial \hat{\mathbf{Y}}_{\mathrm{M}}(k)$ and $\partial \mathbf{J}^{\mathrm{C}}(k)/\partial \mathbf{A}(k)$ are given in (25) and (26) for the MLPN and RBFN, respectively, as shown at the bottom of the next page, where definitions are as follows.

- t is target value.
- m_l is the number of neurons in the hidden layer.
- p is the output of the activation function for a neuron.
- q is the regression vector as the activity of a neuron.
- L and l denote the output and hidden layer, respectively.
- The subscripts M and C for *center* $\mathbf{C}$ and *width* β of the RBFN denote the model and critic network, respectively.
- The function f_1 is the sigmoidal function in (14).
- The function f_2 is the Gaussian density function defined in the right-hand side in (15) as an exponential form.

E. Training Procedure for the Critic and Action Networks

The online training procedure for the critic and action networks (with the model network's weights fixed) is explained in more detail in [10] and [12]. It consists of two training cycles: one for the critic network and the other for the action network.

The critic network's training is carried out first with the switches $S1$ and $S2$ in position 3 (with initial weights of the action network that ensure stabilizing control at an operating point) until convergence is reached as illustrated in Fig. 9.

The critic network's weights $\mathbf{W}_{\mathrm{C}}$ are initialized with small random values, and in its training cycle, the incremental optimization is carried out by (20)–(22). The critic network's weights are now fixed, and training of the action network

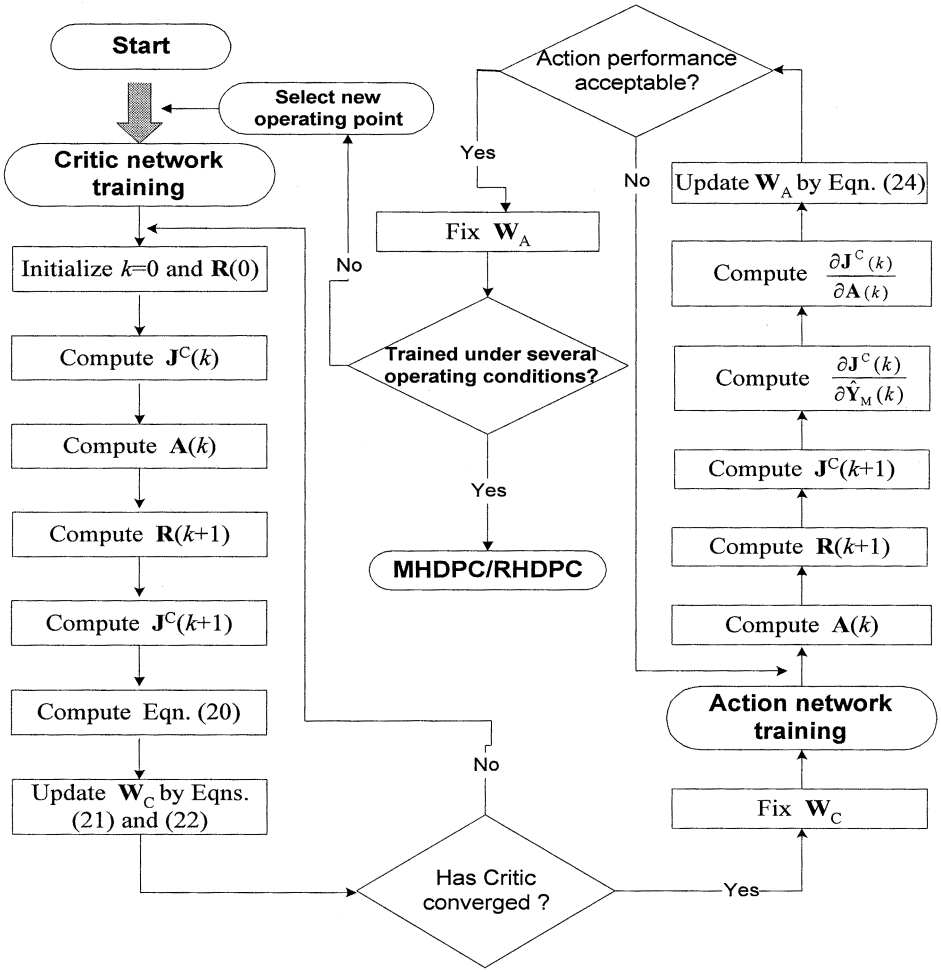


Fig. 9. Training procedure for the critic and action networks.

continues by using (23) and (24) until convergence of the action network's weights are achieved.

The action network's weights are now fixed, the plant operating condition is changed, and training of the critic network starts again. In this way, the training alternates between the critic and action networks while from time to time changing the plant operating point.

The convergence of the action network's weights means that the training procedure has found weights that yield optimal control like the $\mathbf{u}^*$ in (8) for the plant under consideration. The result of critic network's training using the MLPN and RBFN is illustrated in the Appendix. The discount factor γ of 0.5 and the utility function given in (27) are used for the heuristic cost-to-go function in (13)

$$\mathbf{U}^{C}(k) = [4\Delta V_t(k) + 4\Delta V_t(k-1) + 16\Delta V_t(k-2)]^2 + [0.4\Delta\omega(k) + 0.4\Delta\omega(k-1) + 0.16\Delta\omega(k-2)]^2. \quad (27)$$

After the above training procedure has been carried out, switches $S1$ and $S2$ are moved to position 1, and training continues for large disturbances applied to the plant.

$$\frac{\partial \mathbf{J}^{C}}{\partial \hat{\mathbf{Y}}_M} = \frac{\partial \mathbf{J}^{C}}{\partial t}\frac{\partial t}{\partial p_L}\frac{\partial p_L}{\partial q_L}\frac{\partial q_L}{\partial p_l}\frac{\partial p_l}{\partial q_l}\frac{\partial q_l}{\partial \hat{\mathbf{Y}}_M} = \begin{cases} \left[\{f_1(q_l)(1-f_1(q_l))\,\mathbf{W}_{C,1}\}\sum_{j=1}^{m_l} 1\cdot\mathbf{W}_{C,L}\right]|_{\mathrm{MLPN}} \\ \left[\left\{\left(2\sum_{p=1}^{m_l}\frac{\mathbf{C}_{C,p}-q_l}{\beta_{C,p}^2}\right)f_2(q_l)\right\}\sum_{j=1}^{m_l} 1\cdot\mathbf{W}_{C,L}\right]|_{\mathrm{RBFN}} \end{cases} \quad (25)$$

$$\frac{\partial \mathbf{J}^{C}}{\partial \mathbf{A}} = \frac{\partial \mathbf{J}^{C}}{\partial t}\frac{\partial t}{\partial p_L}\frac{\partial p_L}{\partial q_L}\frac{\partial q_L}{\partial p_l}\frac{\partial p_l}{\partial q_l}\frac{\partial q_l}{\partial \mathbf{A}} = \begin{cases} \left[\{f_1(q_l)(1-f_1(q_l))\,\mathbf{W}_{M,l}\}\sum_{j=1}^{m_l}\frac{\partial \mathbf{J}^{C}}{\partial \hat{\mathbf{Y}}_M}\cdot\mathbf{W}_{M,L}\right]|_{\mathrm{MLPN}} \\ \left[\left\{\left(2\sum_{p=1}^{m_l}\frac{\mathbf{C}_{M,p}-q_l}{\beta_{M,p}^2}\right)f_2(q_l)\right\}\sum_{j=1}^{m_l}\frac{\partial \mathbf{J}^{C}}{\partial \hat{\mathbf{Y}}_M}\cdot\mathbf{W}_{M,L}\right]|_{\mathrm{RBFN}} \end{cases} \quad (26)$$

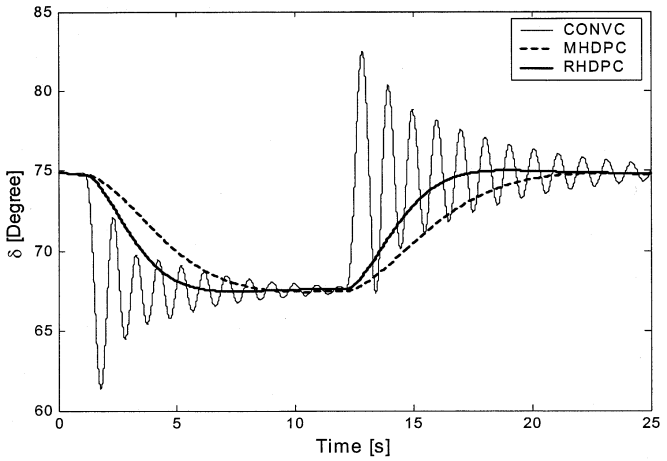


Fig. 10. ±5% step changes in reference voltage of exciter: rotor angle.

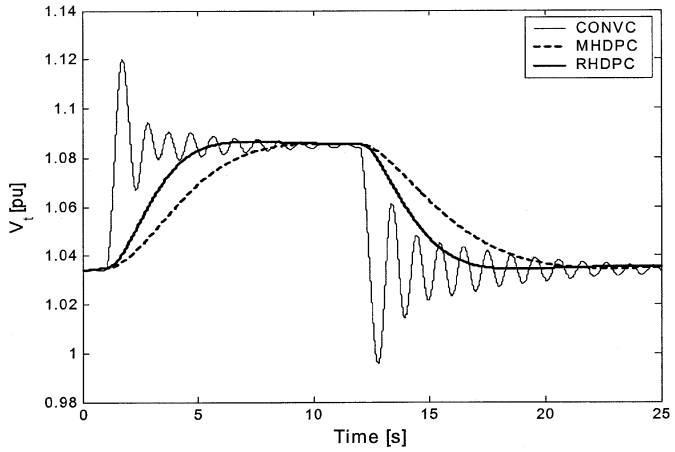


Fig. 11. ±5% step changes in reference voltage of exciter: terminal voltage.

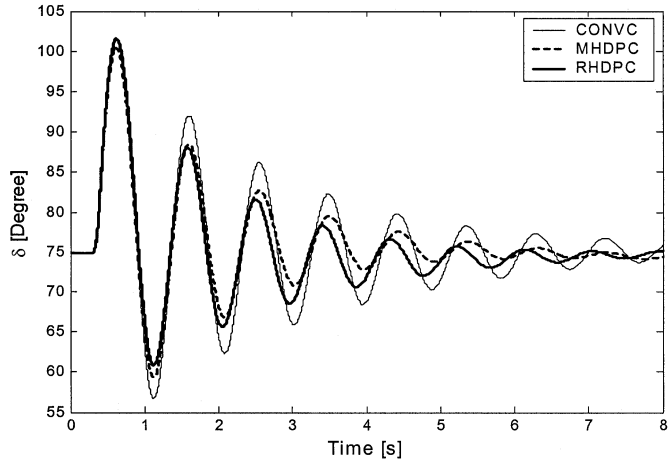


Fig. 12. Three-phase short-circuit test: rotor angle.

III. Case Studies in an SMIB System

After training the critic and action network on-line with the acceptable performance, the MHDPC and RHDPC with fixed weights are ready to control the plant for the real-time operation. The performances of the optimal neurocontrollers, which are the MHDPC and RHDPC trained with deviation signals, are compared with CONVC for the improvement of system damping and transient stability. Two different types of disturbances, namely, a ±5% step change in the reference voltage of exciter and a three phase short circuit at the infinite bus, are carried out to evaluate the performance of the controllers.

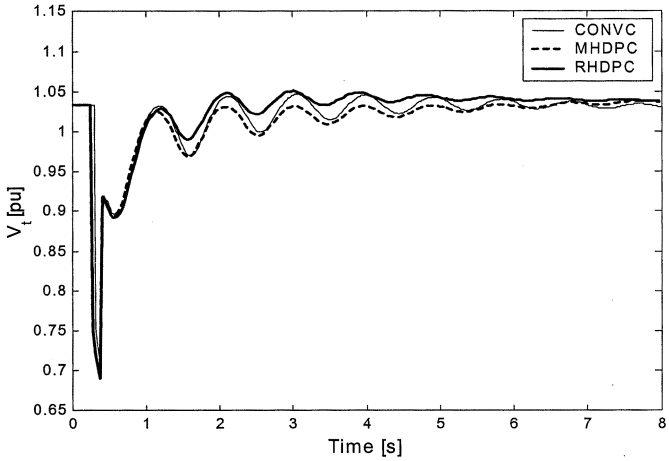


Fig. 13. Three-phase short-circuit test: terminal voltage.

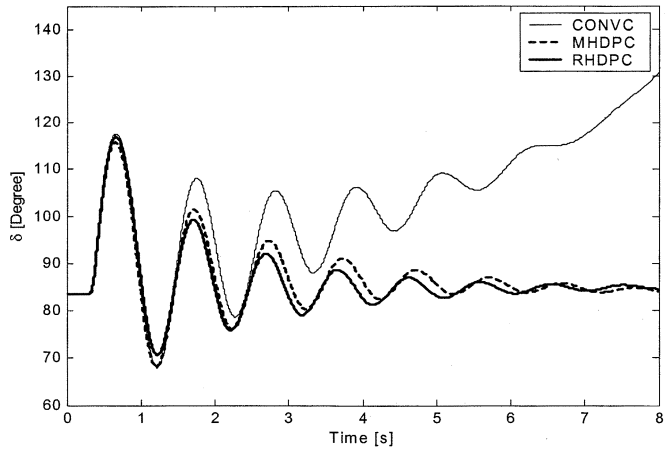


Fig. 14. Three-phase short-circuit test: rotor angle response at $P = 1.1$ pu and $Q = 0.19$ pu operating point.

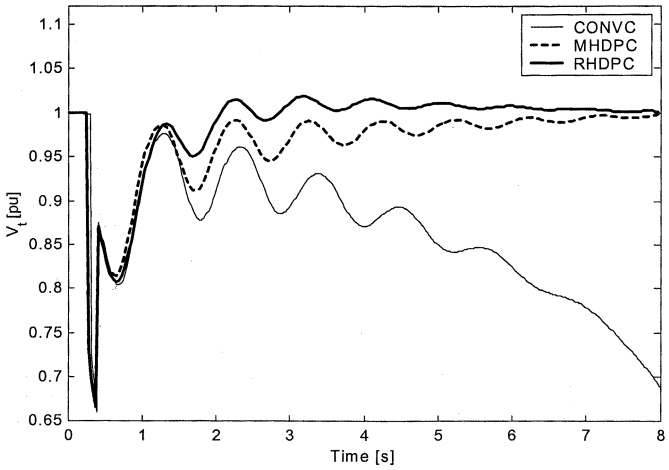


Fig. 15. Three-phase short-circuit test: terminal voltage response at $P = 1.1$ pu and $Q = 0.19$ pu operating point.

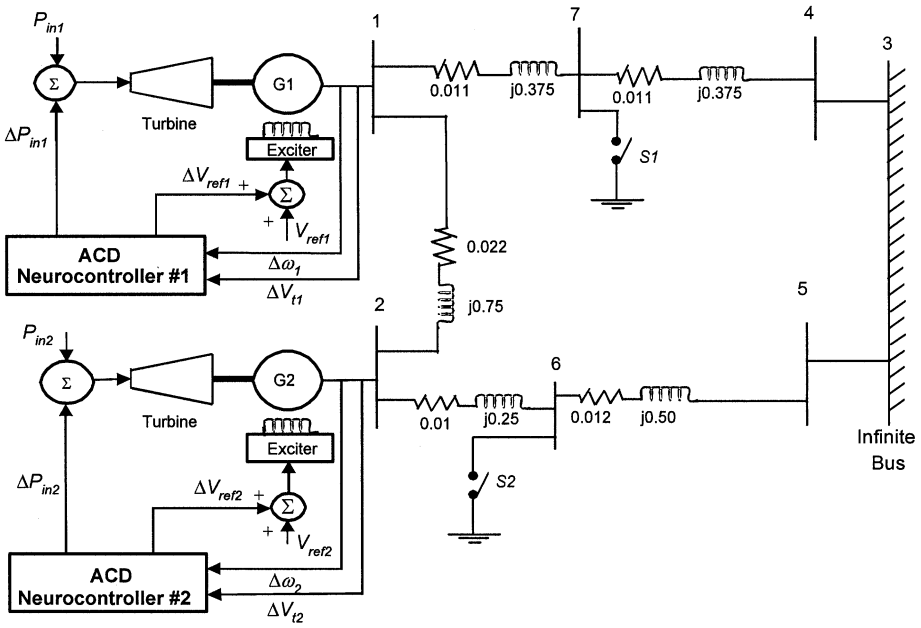


Fig. 16. Multimachine power system with two ACD neurocontrollers connected to generators G1 and G2.

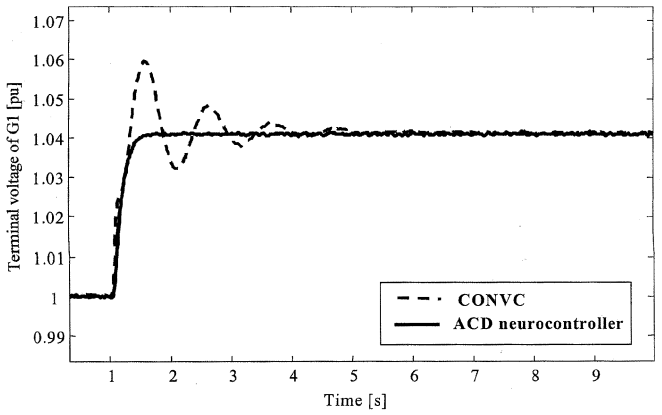


Fig. 17. Terminal voltage of G1 for a 4% step change in reference voltage $V_{\rm ref1}$ of exciter.

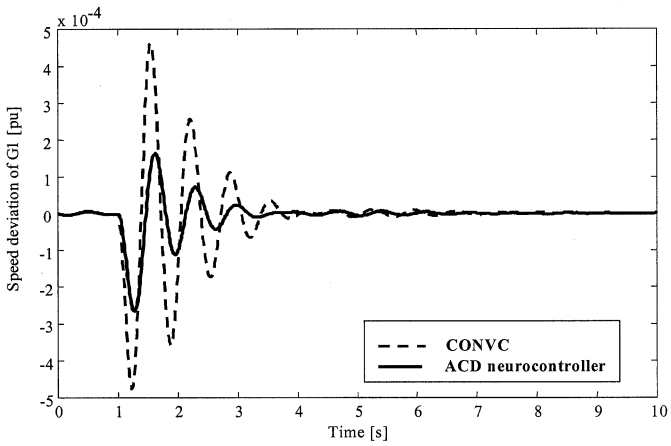


Fig. 18. Speed deviation of G1 for a 4% step change in reference voltage $V_{\rm ref1}$ of exciter.

A. $\pm$5% Step Changes in the Reference Voltage of Exciter

The plant is operating at a steady-state condition ($P_t = 1[\rm pu]$, $Q_t = 0.234$ [pu], and $Z_e = 0.02 + \rm j0.4$ [pu]). At $\rm t = 1$ s, a 5% step increase in the reference voltage of the exciter is applied. At $\rm t = 12$ s, the 5% step increase is removed, and the system returns to its initial operating point. The results in Figs. 10 and 11 show that the optimal neurocontrollers improve the transient system damping compared to the CONVC, and that the RHDPC outperforms the MHDPC, i.e., the RHDPC has the faster transient response than the MHDPC.

B. Three-Phase Short-Circuit Test to Represent a Large-Impulse-Type Disturbance

A severe test is now carried out to evaluate the performances of the controllers under a large disturbance. At $t = 0.3$ s, a temporary three-phase short circuit is applied at the infinite bus for 100 ms from $t = 0.3$ s to 0.4 s for the plant operating at the same steady state condition as previous test. The results comparing the performance of the MHDPC, RHDPC, and CONVC,

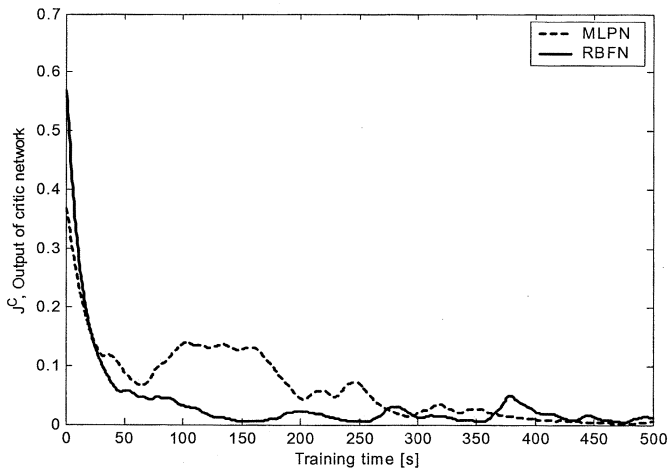


Fig. 19. Output of the critic network $\mathbf{J}^{\rm C}$ versus training time using the MLPN and RBFN.

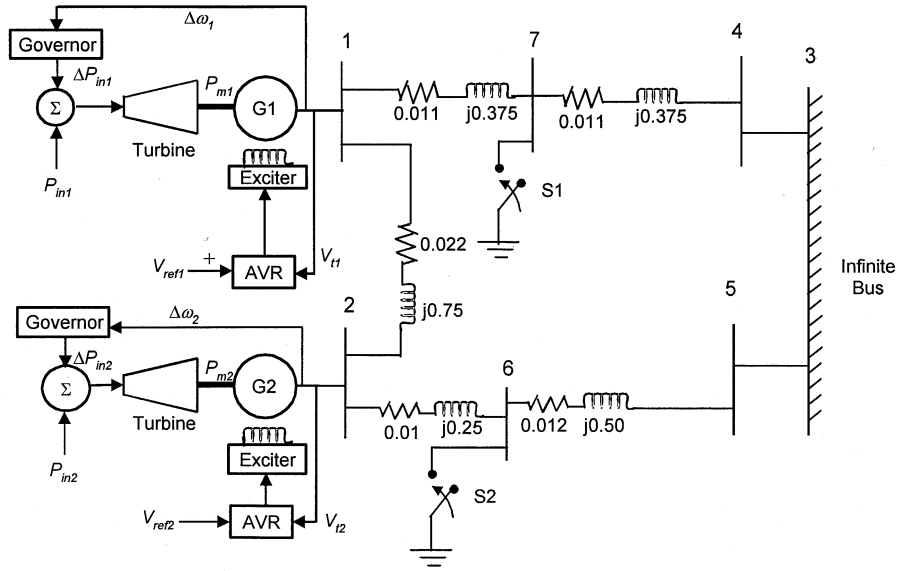


Fig. 20. Multimachine power system with the CONVCs.

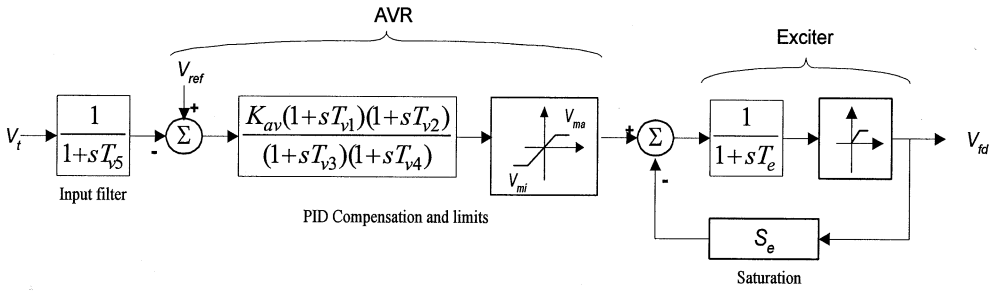


Fig. 21. Block diagram of the AVR and exciter combination.

are shown in Figs. 12 and 13. They show that the optimal neurocontrollers (MHDPC/RHDPC) damp out the low frequency oscillations for the rotor angle (δ) and terminal voltage (V_t) more effectively than the CONVC.

C. Three-Phase Short-Circuit Test Close to the Stability Limit

In order to test the robustness of the proposed neurocontrollers, the plant pre-fault operating point is now changed to a different steady state condition from the previous tests. The active power from the generator is increased by 10% to $P_t = 1.1$ [pu], and $Q_t = 0.19$ [pu], which is closer to the stability limit of the generator. At $t = 0.3$ s, the same 100 ms three phase short circuit is again applied at the infinite bus. The same controller parameters for the MHDPC, RDHPC, and CONVC, used in previous tests, are again used.

The performances of the CONVC, MHDPC, and RHDPC in Figs. 14 and 15 show that the synchronous generator controlled by the CONVC goes unstable and loses synchronism after the disturbance. In contrast, the two neurocontrollers damp out the oscillations and restore the generator to a stable mode. This means that a generator equipped with neurocontrollers based on the HDP algorithm can be operated at 110% power and still remain stable after such a severe fault. This has major implications on being able to produce more power per dollar of invested capital.

TABLE I
TIME CONSTANTS AND GAINS OF AVR-EXCITER/TURBINE–GOVERNOR SYSTEMS

Time constants	Actual values	Time constants	Actual values
T_{v1}	0.616 s	T_{g4}	0.594 s
T_{v2}	2.266 s	T_{g5}	2.662 s
T_{v3}	0.189 s		
T_{v4}	0.039 s	Gains	Actual values
T_{v5}	0.0235 s	K_{av}	1
T_e	0.47 s	V_{ma}	19.59
T_{g1}	0.264 s	V_{mi}	-14.51
T_{g2}	0.0264 s	F	0.322
T_{g3}	0.15 s	K_g	0.05

Also, these results prove the robustness of the neurocontrollers, which provides a good damping performance under the different operating conditions (close to stability limit of the synchronous generator) with feedback loop parameters determined from the infinite horizon optimal control problem.

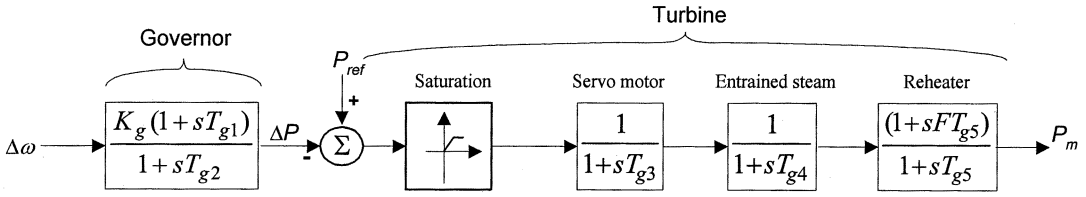


Fig. 22. Block diagram of the turbine and governor combination.

IV. Case Study in a Multimachine Power System

The feasibility of the adaptive critic based neurocontroller on the multimachine power system shown in Fig. 16 is now evaluated. Two generators (G1 and G2) are equipped with the CONVC and then with an adaptive-critic-based neurocontroller. The neurocontoller, which has the model, critic, and action networks, as before, is trained for each generator as described for the SMIB system earlier in this paper, at different operating points. The multimachine power system with the conventional controllers is shown in Section B of the Appendix, and their parameters are identical to those in Sections D and E of the Appendix.

To evaluate the dynamic performances of the controllers, the two generators are operated at an operating condition ($P_{t1} = 0.2$ [pu], $Q_{t1} = -0.02$ [pu] and $P_{t2} = 0.2$ [pu], $Q_{t2} = -0.02$ [pu]), and a 4% step increase in the reference voltage V_{ref1} of the exciter connected to the G1 occurs at $\text{t} = 1$ s. The results of this change appear in Figs. 17 and 18. This shows that the proposed neurocontroller ensures superior transient (for the terminal voltage change) and damping (for the low-frequency oscillation of speed deviation) responses of the system compared to the CONVC.

V. Conclusion

This paper has shown the adaptive critic neural network design as an alternative to the classical optimal control method. The MLPN- and RBFN-based HDP optimal neurocontrollers (MHDPC/RHDPC) have been designed for the control of a synchronous generator in a single machine connected to an infinite bus (SMIB) system and on two generators in a multimachine power system.

The results show that not only do the optimal neurocontrollers improve the system damping and dynamic transient stability more effectively than the CONVC for the large disturbance such as a three phase short circuit, but also the RHDPC has a faster transient response than the MHDPC for a small disturbance like $\pm 5\%$ step changes in the reference voltage of the exciter in a SMIB system. Moreover, the performance of the proposed ACD-based neurocontroller also demonstrates the usefulness of this technique on a practical multimachine power system.

Appendix

A. Results of Critic Network's Training

The results of the critic network's on-line training using the MLPN and RBFN are shown in Fig. 19. With respect to the output of the critic network $\mathbf{J}^{\text{C}}$, it can be observed that the critic network based on the RBFN has a faster convergence capability than the critic network using the MLPN.

B. Multimachine Power System With CONVCs

See Fig. 20.

C. Transmission Line

The transmission line system in Fig. 4 is modeled with the d-q equations given in (A.1) and (A.2)

$$v_{td} = V_b \sin\delta - R_e i_d - L_e \frac{di_d}{dt} + \omega L_e i_q \qquad \text{(A.1)}$$

$$v_{tq} = V_b \cos\delta - R_e i_q - L_e \frac{di_q}{dt} - \omega L_e i_d. \qquad \text{(A.2)}$$

D. AVR and Exciter System

The conventional AVR and exciter combination transfer function block diagram is shown in Fig. 21. The time constants/gains are given in Table I. The exciter saturation factor S_e is given by (A.3). T_{v1}, T_{v2}, T_{v3}, and T_{v4} are the time constants of the PID voltage regulator compensator; T_{v5} is the input filter time constant; T_e is the exciter time constant; K_{av} is the AVR gain; V_{ma} and V_{mi} are AVR maximum and minimum ceilings

$$S_e = 0.6093 \ \exp(0.2165 V_{fd}). \qquad \text{(A.3)}$$

E. Turbine and Governor System

The turbine and governor combination transfer function block diagram is shown in Fig. 22, and the time constants/gain are also given in Table I. The output power of the turbine is limited between zero and 120%. T_{g1} and T_{g2} are the time constants for phase advance compensation, T_{g3} is the servo motor time constant, T_{g4} is the entrained steam delay, T_{g5} is the steam reheat time constant, F is the pu shaft output ahead of reheater, and K_g is the governor gain.

References

[1] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*. New York: IEEE Press, 1994.

[2] B. Adkins and R. G. Harley, *The General Theory of Alternating Current Machines*. London, U.K.: Chapman & Hall, 1975.

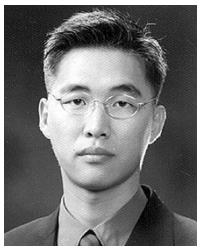
[3] P. Shamsollahi and O. P. Malik, "Direct neural adaptive control applied to synchronous generator," *IEEE Trans. Energy Conversion*, vol. 14, pp. 1341–1346, Dec. 1999.

[4] Y.-M. Park, M.-S. Choi, and K. Y. Lee, "A neural network-based power system stabilizer using power flow characteristics," *IEEE Trans. Energy Conversion*, vol. 11, pp. 435–441, June 1996.

[5] G. K. Venayagamoorthy and R. G. Harley, "A continually online trained neurocontroller for excitation and turbine control of a turbogenerator," *IEEE Trans. Energy Conversion*, vol. 16, pp. 261–269, Sept. 2001.

[6] R. Segal, M. L. Kothari, and S. Madnani, "Radial basis function (RBF) network adaptive power system stabilizer," *IEEE Trans. Power Syst.*, vol. 15, pp. 722–727, May 2000.

[7] E. Swidenbank, S. McLoone, D. Flynn, G. W. Irwin, M. D. Brown, and B. W. Hogg, "Neural network based control for synchronous generators," *IEEE Trans. Energy Conversion*, vol. 14, pp. 1673–1678, Dec. 1999.
[8] J.-W. Park, R. G. Harley, and G. K. Venayagamoorthy, "Comparison of MLP and RBF neural networks using deviation signals for on-line identification of a synchronous generator," in *Proc. 2002 IEEE PES Winter Meeting*, vol. 1, New York, January 2002, pp. 274–279.
[9] ——, "Comparison of MLP and RBF neural networks using deviation signals for indirect adaptive control of a synchronous generator," in *Proc. Int. Joint Conf. Neural Networks, IJCNN02*, May 2002, pp. 919–924.
[10] W. T. Miller, R. S. Sutton, and P. J. Werbos, *Neural Networks for Control*. Cambridge, MA: MIT Press, 1990.
[11] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control*, D. White and D. Sofge, Eds. New York: Van Nostrand Reinhold, 1992, pp. 493–526.
[12] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Networks*, vol. 8, pp. 997–1007, Sept. 1997.
[13] D. V. Prokhorov, "Adaptive critic designs and their applications," Ph.D. dissertation, Texas Tech University, Lubbock, TX, 1997.
[14] J. Si and Y.-T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Networks*, vol. 12, pp. 264–276, Mar. 2001.
[15] G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch, "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator," *IEEE Trans. Neural Networks*, vol. 13, pp. 764–773, May 2002.
[16] ——, "Dual heuristic programming excitation neurocontrol for generators in a multimachine power system," in *Conf. Rec. IEEE-IAS Annu. Meeting*, vol. 2, 2001, pp. 926–931.
[17] W. L. Brogan, *Modern Control Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[18] J. Gregory and C. Kin, *Constrained Optimization in the Calculus of Variations and Optimal Control Theory*. New York: Van Nostrand Reinhold, 1992.
[19] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2001.
[20] D. A. White and M. I. Jordan, "Optimal control: a foundation for intelligent control," in *Handbook of Intelligent Control*, D. White and D. Sofge, Eds. New York: Van Nostrand Reinhold, 1992, pp. 185–214.
[21] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, pp. 1550–1560, Oct. 1990.
[22] Z. Uykan, C. Guzelis, M. E. Celebi, and H. N. Koivo, "Analysis of input-output clustering for determining centers of RBFN," *IEEE Trans. Neural Networks*, vol. 11, pp. 851–858, July 2000.



Jung-Wook Park (S'00–M'03) was born in Seoul, Korea. He received the B.S. degree (*summa cum laude*) from the Department of Electrical Engineering, Yonsei University, Seoul, Korea, in 1999, and the M.S. degree in electrical and computer engineering and the Ph.D. degree from Georgia Institute of Technology, Atlanta, in 2000 and 2003, respectively.

Since August 2003, he has been a Post-Doctoral Research Associate for the IEEE Task Force on benchmark systems (for global optimization of power networks) at the University of Wisconsin, Madison. His current research interests are in power system dynamics, electric machines, FACTS devices, power quality, optimization control algorithms, and application of artificial neural networks.

Ronald G. Harley (M'77–SM'86–F'92) was born in South Africa. He received the B.Sc.Eng. (*cum laude*) and M.Sc.Eng. (*cum laude*) degrees from the University of Pretoria, Pretoria, South Africa, in 1960 and 1965, respectively, and the Ph.D. degree from London University, London, U.K., in 1969.

In 1971, he was appointed to the Chair of Electrical Machines and Power Systems at the University of Natal, Durban, South Africa. He was a Visiting Professor at Iowa State University, Ames, in 1977, at Clemson University, Clemson, SC, in 1987, and at Georgia Institute of Technology, Atlanta, in 1994. He is currently the Duke Power Company Distinguished Professor at Georgia Institute of Technology. His research interests include the dynamic behavior and condition monitoring of electric machines, motor drives, and power systems, and controlling them by the use of power electronics and intelligent control algorithms. He has coauthored some 280 papers published in refereed journals and international conference proceedings, of which nine have received prizes.

Dr. Harley was elected as a Distinguished Lecturer by the IEEE Industry Applications Society (IAS) for the years 2000 and 2001. He is currently the Vice-President of Operations of the IEEE Power Electronics Society, and the Chair of the Distinguished Lecturers and Regional Speakers Program of the IAS. He is a Fellow of the Institution of Electrical Engineers, U.K., and the Royal Society in South Africa, and a Founder Member of the Academy of Science in South Africa formed in 1994.

Ganesh Kumar Venayagamoorthy (S'91–M'97–SM'02) was born in Jaffna, Sri Lanka. He received the B.Eng. (Honors) degree with a first class in electrical and electronics engineering from the Abubakar Tafawa Balewa University, Bauchi, Nigeria, and the M.Sc.Eng. and Ph.D. degrees in electrical engineering from the University of Natal, Durban, South Africa, in 1994, 1999, and 2002, respectively.

He was appointed as a Lecturer at the Durban Institute of Technology, Durban, South Africa, during the period March 1996–April 2001 and thereafter as a Senior Lecturer from May 2001 to April 2002. He was a Research Associate at Texas Tech University, Lubbock, in 1999 and at the University of Missouri, Rolla, in 2000/2001. He is currently an Assistant Professor at the University of Missouri, Rolla. His research interests are in power systems, control systems, computational intelligence, and evolving hardware. He has authored more than 70 papers published in refereed journals and international conference proceedings.

Dr. Venayagamoorthy was a 2001 recipient of the IEEE Neural Network Society summer research scholarship and the 2003 International Neural Network Society Young Investigator Award recipient. He is a member of the South Africa Institute of Electrical Engineers. He was Technical Program Co-Chair of the International Joint Conference on Neural Networks (IJCNN) 2003, Portland, OR.