01 Jan 2003

Dual Heuristic Programming Excitation Neurocontrol for Generators in a Multimachine Power System

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Donald C. Wunsch
Missouri University of Science and Technology, dwunsch@mst.edu

Ronald G. Harley

Recommended Citation

G. K. Venayagamoorthy et al., "Dual Heuristic Programming Excitation Neurocontrol for Generators in a Multimachine Power System," *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.2003.809438

Dual Heuristic Programming Excitation Neurocontrol for Generators in a Multimachine Power System

Ganesh Kumar Venayagamoorthy, *Senior Member, IEEE*, Ronald G. Harley, *Fellow, IEEE*, and Donald C. Wunsch, *Senior Member, IEEE*

***Abstract*—The design of nonlinear optimal neurocontrollers that replace the conventional automatic voltage regulators for excitation control of turbogenerators in a multimachine power system is presented in this paper. The neurocontroller design is based on Dual Heuristic Programming (DHP), a powerful adaptive critic technique. The feedback variables are completely based on local measurements from the generators. Simulations on a three-machine power system demonstrate that DHP-based neurocontrol is much more effective than the conventional proportional–integral–derivative control for improving dynamic performance and stability of the power grid under small and large disturbances. This paper also shows how to design optimal multiple neurocontrollers for nonlinear systems, such as power systems, without having to do continually online training of the neural networks, thus avoiding risks of neural network instability.**

***Index Terms*—Adaptive critics, artificial neural networks (ANNs), generators, multimachine power systems, multiple nonlinear optimal neurocontrollers, power system stability, voltage regulation.**

I. Introduction

POWER SYSTEMS containing turbogenerators are large-scale nonlinear systems. The traditional excitation controllers for the generators are designed by linear control theory based on a single-machine infinite-bus (SMIB) power system model. These SMIB power system mathematical models are linearized at specific operating points and then excitation controllers are designed. The machine parameters change with loading in a complex manner, resulting in different behavior at different operating points and the controller which stabilizes the system under specific operating conditions, may no longer yield satisfactory results when there is a drastic change in the power system operating conditions and configurations. Conservative designs are, therefore, traditionally used, particularly in multimachine systems, to attempt satisfactory control over the entire operating range of the power system.

In recent years, renewed interest has been shown in power systems excitation control using nonlinear control theory, particularly to improve system transient stability [1], [2]. Instead of using an approximate linear model, nonlinear models are used and nonlinear feedback linearization techniques are employed for the generator models, thereby alleviating the operating point dependent nature of the linear designs. Using nonlinear controllers, generator transient stability can be improved significantly. However, nonlinear controllers have a more complicated structure and are difficult to implement relative to linear controllers. In addition, feedback linearization methods require exact system parameters to cancel the inherent system nonlinearities, and this contributes further to the complexity of the stability analysis. However, the use of artificial neural networks (ANNs) as neurocontrollers offers a possibility to overcome this problem.

Multilayer-perceptron (MLP)-type ANNs are able to identify/model time-varying single turbogenerator systems [3] and, with continually online training, these models can track the dynamics of the power system, thus yielding adaptive identification. ANN controllers have been successfully implemented on single turbogenerators using ANN identifiers and indirect feedback control [4]. Adaptive critic designs which yield nonlinear optimal controllers have also been applied successfully to control generators in an SMIB power system [5]. Moreover, ANN identification of turbogenerators in a multimachine power system has also been reported [6], but not yet applied to a controller.

This paper extends previous work by the authors and shows the following.

- The design methodology of nonlinear optimal excitation neurocontrollers based on Dual Heuristic Programming (DHP) theory (a member of the adaptive critics family) for multimachine power systems, to replace the traditional automatic voltage regulators (AVRs) and which includes the ANN identifier [6].
- These nonlinear optimal neurocontrollers can be designed offline, avoiding the computational load of online learning and the issues of neural network instability.
- The simulation and practical results on a three-machine power system show that both voltage regulation and system stability enhancement can be achieved with these

Paper MSDAD-A 02–42, presented at the 2001 Industry Applications Society Annual Meeting, Chicago, IL, September 30–October 5, 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, 2001 and released for publication January 16, 2003. This work was supported by the National Research Foundation, South Africa, and the National Science Foundation.

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

R. G. Harley is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA, and also with the School of Electrical and Electronic Engineering, University of Natal, Durban 4041, South Africa (e-mail: ron.harley@ee.gatech.edu).

D. C. Wunsch is with the Applied Computational Intelligence Laboratory, University of Missouri, Rolla, MO 65409-0249 USA (e-mail: dwunsch@ece.umr.edu).

Digital Object Identifier 10.1109/TIA.2003.809438

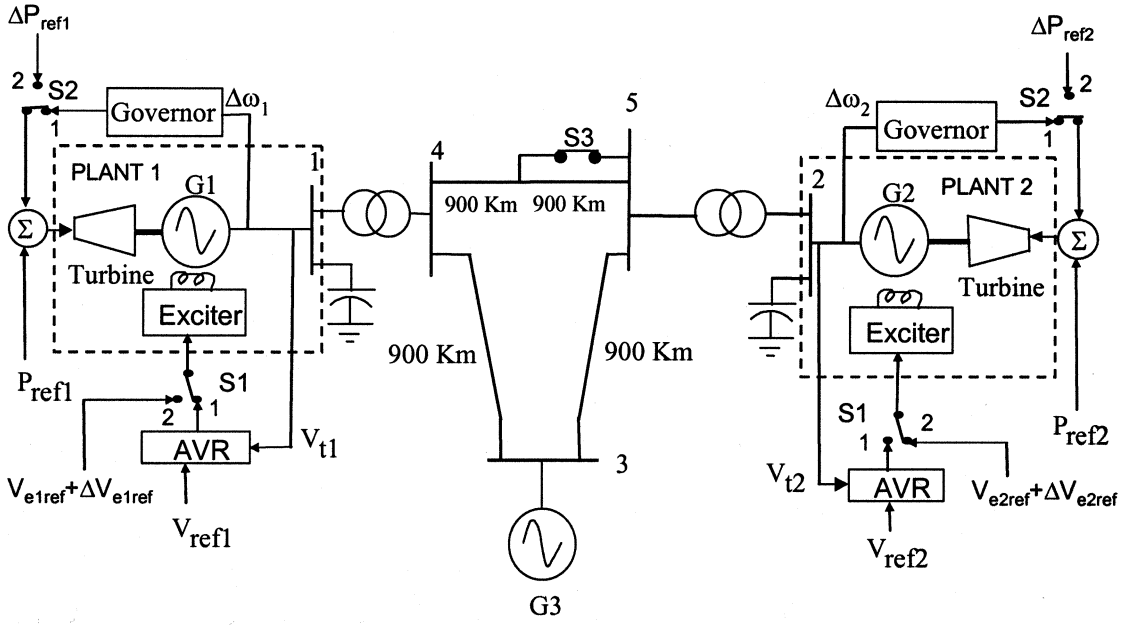

Fig. 1. Three-machine (G1, G2, and G3) five-bus power system.

TABLE I
GENERATORS G1 AND G2 PARAMETERS

T_{do}' = 4.50 s	X_d' = 0.205 pu	Rs = 0.006 pu
T_{do}" = 33 ms	X_d" = 0.164 pu	H = 5.68 s
T_{qo}" = 0.25 s	X_q = 1.98 pu	No. of Poles = 4
X_d = 2.09 pu	X_q" = 0.213 pu	

proposed neurocontrollers, regardless of the changes in the system operating conditions and configurations.

- It is possible to design and implement multiple neurocontrollers controlling multiple generators simultaneously.

II. MULTIMACHINE POWER SYSTEM

The three-machine five-bus power system in Fig. 1 is chosen, to illustrate the effectiveness of adaptive critic-based neurocontrollers. This system is simulated because the controllers designed in this paper will be implemented at the University of Natal's machines research laboratory in Durban, South Africa.

The laboratory power system consists of two generators, each 3 kW, 220 V, designed to have all their per-unit parameters, except the field-winding resistance, the same as those normally expected of a 1000-MW generator. The third machine is the infinite bus. The multimachine laboratory power system in Fig. 1 is simulated in the MATLAB/SIMULINK environment using the Power System Blockset (PSB) [7]. Generators 1 and 2 are each represented by a seventh-order model. There are three coils on the d -axis and two coils on the q axis and the stator transient terms are not neglected. The parameters of the generators, determined by the IEEE standards are given in Table I [8]. A time-constant regulator is used on each generator to insert negative resistance in series with the field-winding circuit, in order to reduce the actual field-winding resistance to the correct per-unit value.

The conventional AVR and exciter combination transfer function block diagram is similar for both generators and is shown in

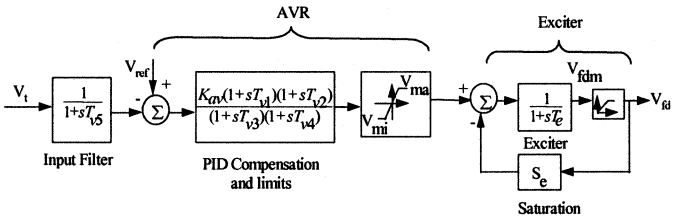

Fig. 2. Block diagram of the AVR and exciter combination of G1 and G2.

TABLE II
AVR AND EXCITER TIME CONSTANTS OF G1 AND G2

T_{v1}	0.616 s	T_{v5}	0.0235 s
T_{v2}	2.266 s	T_e	0.47 s
T_{v3}	0.189 s	K_{av}	0.003
T_{v4}	0.039 s		

Fig. 2, and the time constants are given in Table II. The exciter saturation factor S_e is given by

$$S_e = 0.6093 \exp(0.2165 V_{fd}). \tag{1}$$

T_{v1}, T_{v2}, T_{v3}, and T_{v4} are the time constants of the proportional–integral–derivative (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_{fdm} is the exciter ceiling; and, V_{ma} and V_{mi} are the AVR maximum and minimum ceilings.

A separately excited 5.6-kW dc motor is used as a prime mover, called the microturbine, to drive each of the generators. The torque–speed characteristic of the dc motor is controlled to follow a family of rectangular hyperbola for different positions of the steam valve, as would occur in a real typical high-pressure (HP) turbine cylinder. The three low-pressure (LP) cylinders' inertia are represented by appropriately scaled flywheels. The microturbine and the governor transfer function block diagram is shown in Fig. 3, where P_{ref} is the turbine input power

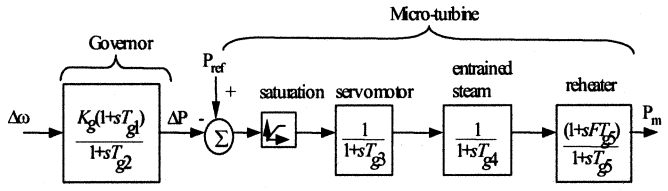

Fig. 3. Block diagram of the microturbine and governor combination of G1 and G2.

TABLE III
MICROTURBINE AND GOVERNOR TIME CONSTANTS OF G1 AND G2

Phase advance compensation, T_{g1}	0.264 s
Phase advance compensation, T_{g2}	0.0264 s
Servo time constant, T_{g3}	0.15 s
Entrained steam delay, T_{g4}	0.594 s
Steam reheat time constant, T_{g5}	2.662 s
pu shaft output ahead of reheater, F	0.322
Governor gain, K_g	0.05

TABLE IV
OPERATING POINTS OF G1 AND G2

	Condition one		Condition two	
	G1	G2	G1	G2
P_e (pu)	0.2000	0.2000	0.3000	0.3000
Q (pu)	-0.0200	-0.0200	-0.0400	-0.0400
V_t (pu)	1	1	1	1

set point value, P_m is the turbine output power, and $\Delta\omega$ is the speed deviation. The turbine and governor time constants are given in Table III.

The AVR and governor parameters, namely, the gain and time constants ($K_{av}, T_{v1}, T_{v2}, T_{v3}, T_{v4}, K_g, T_{g1}$, and T_{g2}), are fine tuned for the first operating condition given in Table IV [9].

III. DERIVATIVE ADAPTIVE-CRITICS-BASED EXCITATION CONTROLLER

A. Background

Adaptive critic designs (ACDs) are neural network designs capable of optimization over time under conditions of noise and uncertainty. A family of ACDs was proposed by Werbos [10] as a new optimization technique combining concepts of reinforcement learning and approximate dynamic programming. For a given series of control actions, that must be taken in sequence, and not knowing the quality of these actions until the end of the sequence, it is impossible to design an optimal controller using traditional supervised ANN learning.

Dynamic programming prescribes a search which tracks backward from the final step, rejecting all suboptimal paths from any given point to the finish, but retains all other possible trajectories in memory until the starting point is reached. However, many paths which may be unimportant are nevertheless also retained until the search is complete. The result is that the procedure is too computationally demanding for most real problems. In supervised learning, an ANN training algorithm utilizes a desired output and, comparing it to the actual output, generates an error term to allow learning. For an MLP-type ANN the backpropagation algorithm is typically used to get the necessary derivatives of the error term with respect to the training parameters and/or the inputs of the network. However, backpropagation can be linked to reinforcement learning via a network called the ***Critic*** network, which has certain desirable attributes.

Critic based methods remove the learning process one step from the control network (traditionally called the "***Action*** network" or "***actor***" in ACD literature), so the desired trajectory or control action information is not necessary. The Critic network learns to approximate the cost-to-go or strategic utility function, and uses the output of an Action network as one of its inputs directly or indirectly. When the Critic network learns, backpropagation of error signals is possible along its input pathway from the Action network. To the backpropagation algorithm, this input pathway looks like just another synaptic connection that needs weight adjustment. Thus, no desired signal is needed. All that is required is a desired cost function J given in (2)

$$J(t) = \sum_{k=0}^{\infty} \gamma^k U(t+k) \tag{2}$$

where γ is a discount factor for finite horizon problems ($0 < \gamma < 1$), and $U(.)$ is the utility function or local cost. Section IV describes how to obtain the utility function for the excitation neurocontroller design.

The Critic and Action networks, can be connected together directly (Action-dependent designs) or through an identification model of a plant (Model-dependent designs). There are three classes of implementations of ACDs called Heuristic Dynamic Programming (HDP), DHP, and Globalized Dual Heuristic Dynamic Programming (GDHP), listed in order of increasing complexity and power [11]. This paper presents the DHP, model-dependent design, and compares its performance against the results obtained using conventional PID controllers. The DHP Critic and Action neural networks are described below.

B. DHP Critic Neural Network

The Critic network is trained forward in time, which is of great importance for real-time optimal control/operation. The ability to foresee future cost and take preventive action ahead of time is important in optimal controller designs. DHP has a critic network which estimates the derivatives of J with respect to a vector of observables of the plant, ΔY. The Critic network learns minimization of the following error measure over time:

$$\|E\| = \sum_t E_C^T(t) E_C(t) \tag{3}$$

where

$$E_C(t) = \frac{\partial J\left[\Delta\hat{Y}(t)\right]}{\partial\,\Delta\hat{Y}(t)} - \gamma\frac{\partial \hat{J}\left[\Delta\hat{Y}(t+1)\right]}{\partial\,\Delta Y(t)} - \frac{\partial U\left[\Delta Y(t)\right]}{\partial\,\Delta Y(t)} \tag{4}$$

where $\partial(.)/\partial\Delta Y(t))$ is a vector containing partial derivatives of the scalar $(.)$ with respect to the components of the vector ΔY. The partial derivatives can be obtained for example by backpropagating through a neural network as described in Fig. 4. DHP has an important advantage over HDP since its critic neural network builds a representation for the derivatives of J directly by being explicitly trained on them through $\partial U(t)/\partial[\Delta Y(t)]$ and $\partial U(t)/\partial A(t)$. For instance, in the area of model-based control, as in the case of this paper, a pretrained Model neural

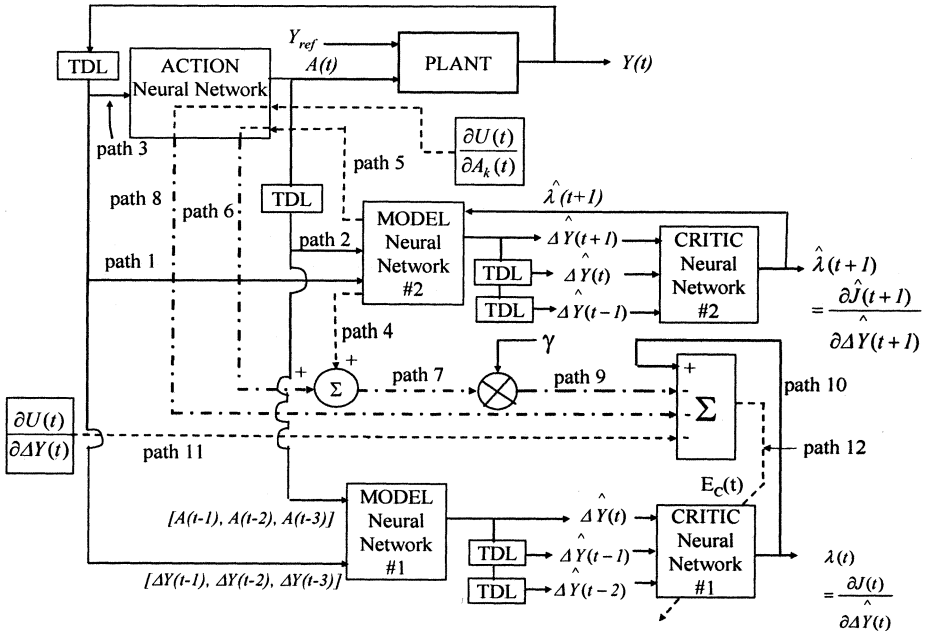

Fig. 4. DHP Critic network training. This diagram shows the implementation of (6). The same Critic network is shown for two consecutive times t and $t+1$. First and second backpropagation paths are shown by dashed lines and dashed–dotted lines, respectively. The output of the Critic network $\lambda(t+1)$ is backpropagated through the Model from its outputs to its inputs, yielding the first term of (5) and $\partial J(t+1)/\partial A(t)$. The latter is backpropagated through the Action from its output to its input forming the second term of (5). Backpropagation of the vector $\partial U(t)/\partial A(t)$ through the Action results in a vector with components computed as the last term of (6). The summation produces the error vector $E_C(t)$ for Critic network training.

network and well-defined $\partial U(t)/\partial[\Delta Y(t)]$ and $\partial U(t)/\partial A(t)$ exist. To adapt the Action neural network, only the derivatives $\partial J(t)/\partial[\Delta Y(t)]$ or $\partial J(t)/\partial A(t)$ are required, rather than the J function itself. However, the approximation of these derivatives is already a direct output of the DHP Critic.

The DHP Critic network's training is more complicated than that of the HDP Critic since there is a need to take into account all relevant pathways of backpropagation as shown in Fig. 4, where the paths of derivatives and training of the Critic are depicted by dashed lines. Details on the HDP Critic training are given in [5].

In DHP, application of the chain rule for derivatives of the jth output yields

$$\frac{\partial \hat{J}\left[\Delta \hat{Y}(t+1)\right]}{\partial \Delta Y_j(t)} = \sum_{i=1}^{n} \hat{\lambda}_i(t+1)\frac{\partial \Delta \hat{Y}_i(t+1)}{\partial \Delta Y_j(t)} \cdot \sum_{k=1}^{m}\sum_{i=1}^{n} \hat{\lambda}_i(t+1)\frac{\partial \Delta \hat{Y}_i(t+1)}{\partial A_k(t)}\frac{\partial A_k(t)}{\partial \Delta Y_j(t)} \quad (5)$$

where $\hat{\lambda}_i(t+1) = \partial \hat{J}[\Delta \hat{Y}(t+1)]/\partial \Delta \hat{Y}_i(t+1))$, and n, m are the numbers of outputs of the Model and the Action networks, respectively. By exploiting (5), each of n components of the vector $E_C(t)$ from (4) is determined for the jth output by (6)

$$E_{Cj}(t) = \frac{\partial J\left[\Delta \hat{Y}(t)\right]}{\partial \Delta \hat{Y}_j(t)} - \gamma \frac{\partial \hat{J}\left[\Delta \hat{Y}(t+1)\right]}{\partial \Delta Y_j(t)} - \frac{\partial U\left[\Delta Y(t)\right]}{\partial \Delta Y_j(t)} - \sum_{k=1}^{m}\frac{\partial U(t)}{\partial A_k(t)}\frac{\partial A_k(t)}{\partial \Delta Y_j(t)}. \quad (6)$$

The signals in Fig. 4, labeled with a path number, represent the following.

1) Path 1 represents the outputs of the plant fed into the Model neural network #2. These outputs are $\Delta Y(t)$, $\Delta Y(t-1)$ and $\Delta Y(t-2)$.
2) Path 2 represents the outputs of the Action neural network fed into the Model neural network #2. These outputs are $A(t)$, $A(t-1)$, and $A(t-2)$.
3) Path 3 represents the outputs of the plant fed into the Action neural network. These outputs are $\Delta Y(t)$, $\Delta Y(t-1)$, and $\Delta Y(t-2)$.
4) Path 4 represents a backpropagated signal of the output of the Critic neural network #2 through the Model neural network with respect to path 1 inputs. The backpropagated signal on path 4 is
$$\sum_{i=1}^{n} \hat{\lambda}_i(t+1)\frac{\partial \Delta \hat{Y}_i(t+1)}{\partial \Delta Y_j(t)}$$
in (5).
5) Path 5 represents a backpropagated signal of the output of the Critic neural network #2 through the Model neural network with respect to path 2 inputs. The backpropagated signal on path 3 is
$$\sum_{i=1}^{n} \hat{\lambda}_i(t+1)\frac{\partial \Delta \hat{Y}_i(t+1)}{\partial A_k(t)}$$
in (5).
6) Path 6 represents a backpropagation output of path 5 signal iv) above) with respect to path 3. The signal on path 6 is
$$\sum_{k=1}^{m}\sum_{i=1}^{n} \hat{\lambda}_i(t+1)\frac{\partial \Delta \hat{Y}_i(t+1)}{\partial A_k(t)}$$
in (5).

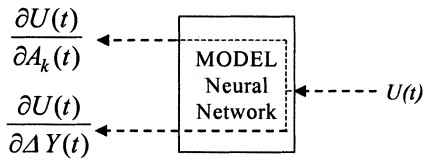

Fig. 5. Backpropagation of $U(t)$ through the Model neural network.

7) Path 7 is the sum of the path 4 and path 6 signals resulting in $\partial\hat{J}[\Delta\hat{Y}(t+1)]/\partial\,\Delta Y_j(t)$, given in (5).
8) Path 8 is the backpropagated signal of the term $\partial U(t)/\partial A_k(t)$ (Fig. 2) with respect to path 3 and is

$$\sum_{k=1}^{m}\frac{\partial U(t)}{\partial A_k(t)}\,\frac{\partial A_k(t)}{\partial\,\Delta Y_j(t)}$$

in (6).
9) Path 9 is a product of the discount factor γ and the path 7 signal, resulting in term $\gamma\partial\hat{J}[\Delta\hat{Y}(t+1)]/\partial\,\Delta Y_j(t)$ in (6).
10) Path 10 represents the output of the Critic neural network #1, $\partial J[\Delta\hat{Y}(t)]/\partial\,\Delta\hat{Y}(t)$.
11) Path 11 represents the term $\partial U(t)/\partial\,\Delta Y(t)$ (Fig. 2).
12) Path 12 represents $E_{Cj}(t)$ given 1 in (6) and is as follows:

$$Path12 = E_{Cj}(t) = path10\text{–}path9\text{–}path11\text{–}path8.$$

The partial derivatives of the utility function $U(t)$ with respect to $A_k(t)$, and $\Delta Y(t)$, $\partial U(t)/\partial A_k(t)$ and $\partial U(t)/\partial\,\Delta Y(t)$, respectively, are obtained by backpropagating the utility function, $U(t)$ through the Model network as shown in Fig. 5.

C. DHP Action Neural Network

The Action network is adapted in Fig. 6 by propagating $\lambda(t+1)$ back through the Model to the Action. The goal of such training is to minimize the sum of the derivatives of the local cost $U(t)$ and the total cost $J(t)$ with respect to the output of the Action network $A(t)$, which can be expressed by (7)

$$\frac{\partial U\,[\Delta Y(t)]}{\partial A(t)}+\gamma\,\frac{\partial\hat{J}\left[\Delta\hat{Y}(t+1)\right]}{\partial A(t)}=0\qquad\forall t. \tag{7}$$

The error signal for the Action network training is, therefore, given as follows:

$$E_A(t)=\frac{\partial U\,[\Delta Y(t)]}{\partial A(t)}+\gamma\,\frac{\partial\hat{J}\left[\Delta\hat{Y}(t+1)\right]}{\partial A(t)}. \tag{8}$$

The weights' update expression [11], when applying backpropagation, is as follows:

$$\Delta W_A=-\alpha\left[\frac{\partial U\,[\Delta Y(t)]}{\partial A(t)}+\gamma\,\frac{\partial\hat{J}\left[\Delta\hat{Y}(t+1)\right]}{\partial A(t)}\right]^T\frac{\partial A(t)}{\partial W_A} \tag{9}$$

where α is a positive learning rate and W_A are weights of the DHP Action neural network. The word "Dual" is used to describe the fact that the target outputs for the DHP Critic training are calculated using backpropagation in a generalized sense; more precisely, it does use dual subroutines (states and co-states) to backpropagate derivatives through the Model and Action neural networks, as shown in Fig. 4. The dual subroutines and more explanations are found in [10] and [12].

IV. Derivation of Utility Function

The utility function $U(t)$ in (2) is designed based on a desired response predictor which has the following characteristics.

1) It must be flexible enough to modify the performance of the turbogenerator.
2) The desired response signal must ensure that the turbogenerator is inherently stable at all times. In other words, the predictor must be stable.
3) The desired response signal must incorporate the effects of a power system stabilizer.

The optimal predictor is designed on the basis of guiding the disturbed output variables, in this case the terminal voltage and speed, of the turbogenerator to a desired steady operating point or set point, in a step-by-step fashion. In other words, a desired trace of outputs from t_i to t_{i+1} can be predicted, based on the present and past-time values of the outputs. Optimal here refers to predictions of the desired response for the turbogenerator and ensuring its stability over a wide range of operating conditions. The prediction equation of the optimal predictor is given in (10).

$$\hat{X}(k+1)=B_0X(k)+B_1X(k-1)+\cdots+B_NX(k-N). \tag{10}$$

$B_i(i=0,1,\ldots,N)$ are chosen so that any disturbed output variable always transfers toward the desired steady operating point, that is, the optimal predictor is always globally asymptotically stable. $\hat{X}$ is the value predicted for the next immediate time step and X can be either the terminal voltage deviation ΔV_t or speed deviation $\Delta\omega$.

In (10), it is assumed that each output variable of the optimal predictor is a linear combination of the independently predicted output variables of the dynamic system. The magnitude of the coefficients, A_i, determines the magnitude of the error signal between the identifier output and the desired response signal (or predictor) and, therefore, the magnitude of the error backpropagated to the controller to adapt its weights.

If the output $X(t)$ is bounded for $0<t<\infty$ and

$$\lim_{t\to\infty}\left(X(t)-\hat{X}(t)\right)=0 \tag{11}$$

then a predictor can be designed which forces the turbogenerator, by means of the ANN controller, back to desired set points [13]. The magnitude of the forcing signal depends on the coefficients A_i.

The conditions defined by (11) are necessary because it is not possible to damp the turbogenerator to take up the required setpoints if its outputs are unbounded. If (11) does not hold, then the outputs of the turbogenerator will not return to their set points after a disturbance. The fundamental assumption made

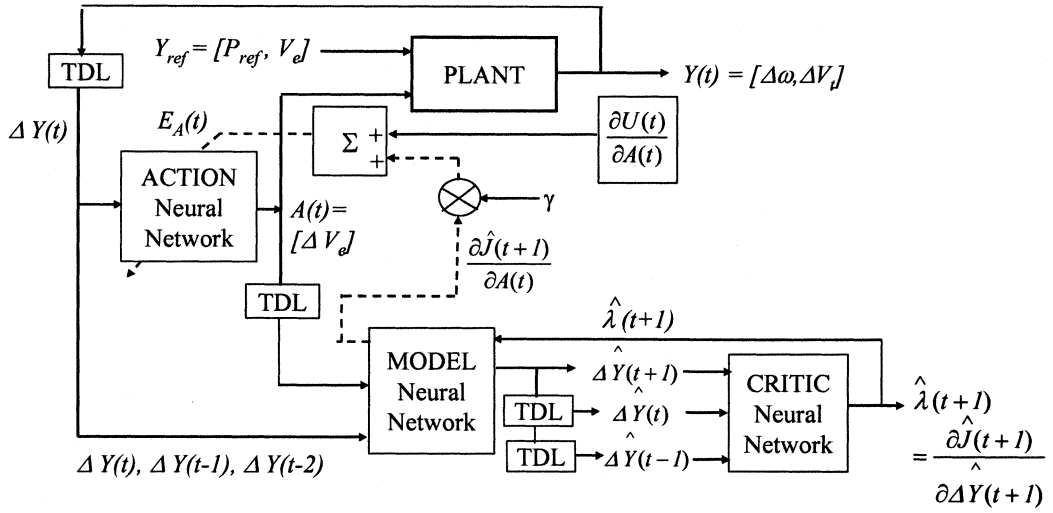

Fig. 6. DHP Action network training. Backpropagation paths shown with dashed lines. The output of the Critic $\lambda(t+1)$ at time $(t+1)$ is backpropagated through the Model from its outputs to its inputs, and the resulting vector is multiplied by discount factor γ and added to $\partial U(t)/\partial A(t)$. Then, an incremental update of the Action network weights is carried out in accordance with (9).

in this design is that it is possible for a controller to return a turbogenerator to its set points after a disturbance.

Equation (10) can be rewritten in the following way:

$$\hat{X}(k+1) = \alpha_i \hat{x}_1(k+1) + \beta_i \hat{x}_2(k+1) + \cdots + \gamma_i \hat{x}_i(k+1) \quad (12)$$

where

$$\hat{x}_i(k+1) = a_{i0}x_i(k) + a_{i1}x_i(k-1) + \cdots + a_{iN}x_i(k-N), \quad i = 1, 2, \ldots, h. \quad (13)$$

The eigenfunction polynomial of (5) is

$$Z - a_{i0} - a_{i1}Z^{-1} - a_{i2}Z^{-2} - \cdots - a_{iN}Z^{-N} = 0 \quad (14)$$

or

$$Z^{N+1} - a_{i0}Z^N - a_{i1}Z^{N-1} - \cdots - a_{iN} = (Z - S_{i0})(Z - S_{i1}) \cdots (Z - S_{iN}) = 0. \quad (15)$$

If $S_{i0}, S_{i1}, \ldots, S_{iN}$ $(i = 1, 2, \ldots, h)$ are chosen inside a unit circle, then eq. (12) will be globally asymptotically stable. It should be pointed out that $\alpha_i, \beta_i, \ldots, \gamma_i$ $(i = 1, 2, \ldots, h)$ in (12) are the qualitative coefficients, and are not relevant to the stability of the dynamic system. These coefficients describe the relationship between the desired outputs of the optimal predictor and the outputs of the dynamic system, and may be chosen according to the qualitative requirements of the controlled turbogenerator system.

An optimal predictor for the turbogenerator is designed as follows [13]:

$$\hat{x}_1(k+1) = \sum_{i=0}^{2} a_{1i}x_1(k-i) + \beta_1 \left\{ \sum_{i=0}^{2} a_{2i}x_2(k-i) \right\}$$

$$\hat{x}_2(k+1) = \sum_{i=0}^{\hat{2}} a_{2i}x_2(k-i) \quad (16)$$

where a_{1i} $(i = 0, 1, 2)$ can be obtained by

$$Z^3 - a_{10}Z^2 - a_{11}Z - a_{12} = (Z^2 - S_{11})(Z - S_{12}). \quad (17)$$

S_{11} and S_{12} are real and inside a unit circle. $a_{2i}(i = 0, 1, 2)$ can be obtained in the same way, $0 < \beta_1 < 1$.

In (16), $\hat{x}_1(k+1)$ and $\hat{x}_2(k+1)$ refer to the predicted terminal voltage and speed deviation, respectively. The predicted terminal voltage deviation depends on both the terminal voltage and speed deviation signals. The weighting of the speed deviation on the predicted terminal voltage deviation depends on the value of β_1. The inclusion of the speed deviation signal for predicting the terminal voltage deviation brings in the effects of power system stabilizers.

To find suitable values for the coefficients in (17), several simulations are carried out starting with small values for S_{11} and S_{12} and the response of the controller (to disturbances such as step changes in terminal voltage, and three-phase short circuits) is evaluated. Small values of S_{11} and S_{12} give better damped responses in turbogenerator speed and voltage. The values S_{11} and S_{12} are increased in steps until acceptable voltage and speed responses are achieved. If too large values of S_{11} and S_{12} are used, then the voltage and speed of the turbogenerator overshoot their set points.

The effect of $\beta_1 = 0.01$ in (16) is to improve the damping of the rotor swings especially after three-phase short circuits but is not too critical if set to zero. The predicted terminal voltage and speed deviation are given by (18)

$$\hat{x}_1(k+1) = 4\,\Delta V(k) + 4\,\Delta V(k-1) + 16\,\Delta V(k-2) + 0.01 \cdot \{0.4\,\Delta\omega(k) + 0.4\,\Delta\omega(k-1) + 0.16\,\Delta\omega(k-2)\}$$

$$\hat{x}_2(k+1) = 0.4\,\Delta\omega(k) + 0.4\,\Delta\omega(k-1) + 0.16\,\Delta\omega(k-2). \quad (18)$$

It can be seen from (18) that the coefficients (4, 4, and 16) used for the terminal voltage deviation fall outside the unit circle. Nevertheless, the results for the controller in the Section VI show that the large values of the coefficients used for voltage deviation (a_{1i}) do not cause instability. The reasons for this are as follows.

1) The limit in (11) applies to the terminal voltage deviations. The ANN controller creates a damping signal only when there is a difference between the turbogenerator's set-point terminal voltage and the instantaneous terminal voltage. The controller ensures that this difference becomes zero over a period of time and the output of (18) will then be zero, even with large coefficients $(a_{1i} > 1)$ for the voltage deviation terms.
2) The turbogenerator used in this study has an open-loop frequency response of about 0.3 Hz to changes in terminal voltage set points, which is considered to be slow. Therefore, the damping signal mentioned in 1) does not cause any oscillation about the set point.

The utility function $U(k)$ is taken to be the sum of squares of $\hat{x}_1(k+1)$ (with $\beta_1 = 0$) and $\hat{x}_2(k+1)$ as shown in (19). The squares are taken to ensure a positive cost when deviations exist

$$U(k) = [4\,\Delta V(k) + 4\,\Delta V(k-1) + 16\,\Delta V(k-2)]^2 + [0.4\,\Delta\omega(k) + 0.4\,\Delta\omega(k-1) + 0.16\,\Delta\omega(k-2)]^2. \quad (19)$$

V. Training of Model, Critic, and Action Neural Networks

The training procedure for the Critic and Action networks is similar to adaptive critic designs for SMIB [5]. It consists of two separate training cycles: one for the Critic (N_C) and the other for the Action (N_A). The Critic's training is done initially with a pretrained Action network [4], [14], to ensure that the whole system, consisting of the neurocontrol and the power system, remains stable. The Action network is pretrained on a linearized model of the generator. The Action is trained further while keeping the Critic network parameters fixed. This process of training the Critic and the Action one after the other is repeated until an acceptable performance is achieved. The ANN Model parameters are assumed to have converged globally during its offline training without any neurocontrollers (briefly described below) [6] and, it is not adapted concurrently with the Critic and Action networks.

A. Model Neural Network

Fig. 7 illustrates how the Model/Identifier network is trained to identify the dynamics of the plant. The Model network structure is a three layer feedforward neural network with 12 inputs, a single hidden layer with 14 sigmoidal neurons, and two linear output neurons as shown in Fig. 8.

The inputs are the *actual* deviation in the input to the exciter ΔV_e, the *actual* deviation in the input to the turbine $\Delta P_{\rm ref}$, the *actual* terminal voltage deviation ΔV_t and the *actual* speed deviation of the generator $\Delta\omega$. These four inputs are time delayed by a sample period of 10 ms and together with the eight previously delayed values form the 12 inputs to the Model network.

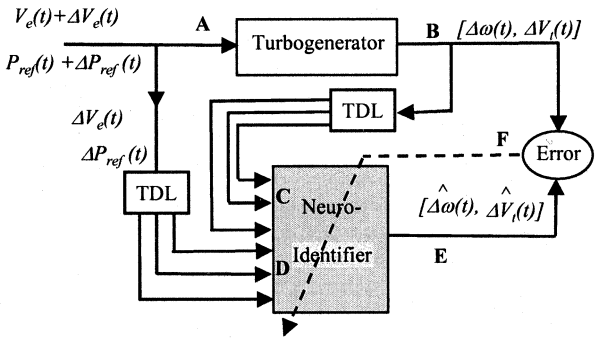

Fig. 7. Neural network modeling of the plant in Fig. 1, using the backpropagation algorithm.

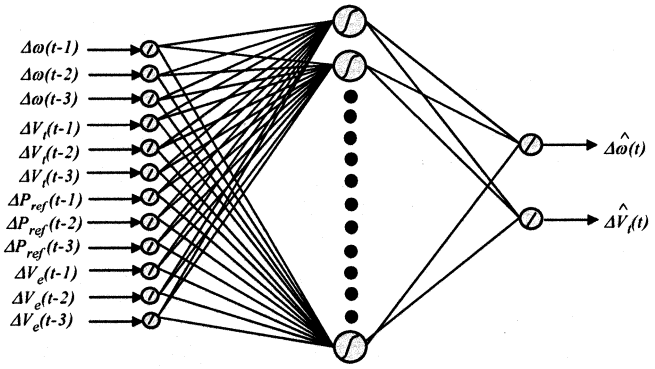

Fig. 8. Model neural network with 12 inputs, 14 sigmoidal hidden layer neurons, and two linear output neurons.

The Model network outputs are the one-step-ahead *estimated* terminal voltage deviation $\Delta\hat{V}_t$ and *estimated* speed deviation $\Delta\hat{\omega}$ of the turbogenerator.

Pseudorandom binary signals (PRBS) are applied to the exciter and the microturbine of the plant with the switches S1 and S2 in position 1 in Fig. 1, in order to train the Model/Identifier network, for a period of time at different operating conditions until satisfactory identification results are obtained. The input and output weights W_M, of the Model network are then fixed during the further development of the Critic and the Action neural networks. The backpropagation algorithm [12] is used for updating W_M in the Model network based on the error $e_m(t)$ at F in Fig. 7 given in (20)

$$e_M(t) = \left\{ \left[\Delta V_t(t) - \Delta\hat{V}_t(t) \right], [\Delta\omega(t) - \Delta\hat{\omega}(t)] \right\}. \quad (20)$$

The training is carried out to minimize (20). The change in the weights is calculated using the backpropagation algorithm based on a gradient-descent method. The Model network weight update equation is given in (21) [12]

$$E_M(t) = \tfrac{1}{2} \sum_t e_M^2(t) \quad (21)$$

$$\Delta W_M = -\eta_1 e_M(t)\, \frac{\partial e_M(t)}{\partial W_M}. \quad (22)$$

The flowchart for the Model network training appears in Fig. 9. Fig. 10 shows the PRBS signal applied to the turbine of generator G2 operating at $P = 0.1$ pu and unity power factor. A similar signal is also applied to the exciter. The training

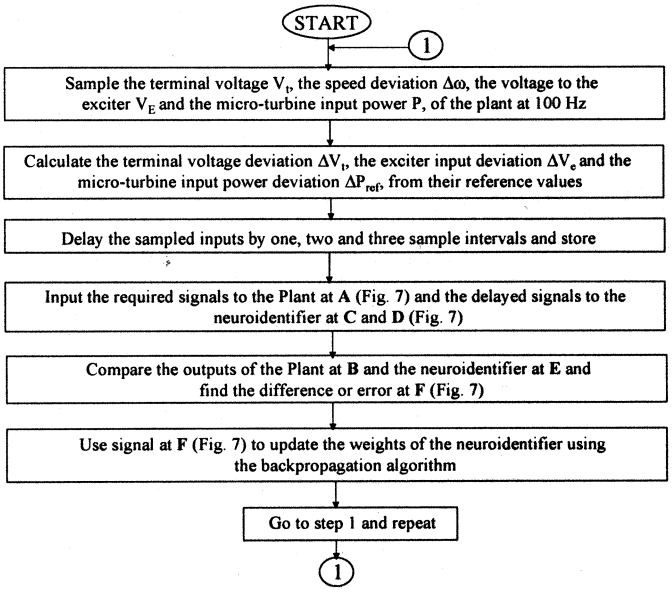

Fig. 9. Flowchart for the Model neural network training.

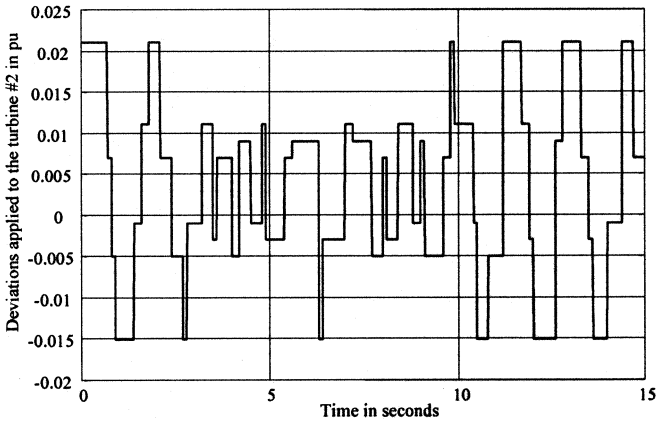

Fig. 10. PRBS training signal applied to the turbine of generator G2.

of the Model networks for generators G1 and G2 in Fig. 1 is carried out separately first and then simultaneous training is carried out applying PRBS signals to both generators. The speed and terminal voltage deviations of generator G2 appear in Figs. 11 and 12. The training of Model networks are carried out for different operating conditions of the multimachine power system until the weights of the Model networks have converged for all these operating conditions. The Critic and Action training use the globally converged Model networks to calculate the derivatives with no further training on the Model concurrently. The Model network can be further trained later to improve the performance of the Action network. More tests and results on the Model network can be found in [6].

B. Critic Neural Network

The Critic network in Fig. 4 is also a three-layer feedforward network with six inputs, 13 hidden neurons, and two outputs (Fig. 13). The inputs to the Critic network are the speed *deviation* $\Delta\omega$ and terminal voltage *deviation* ΔV_t. These inputs are time delayed by a sample period of 10 ms, and together with the

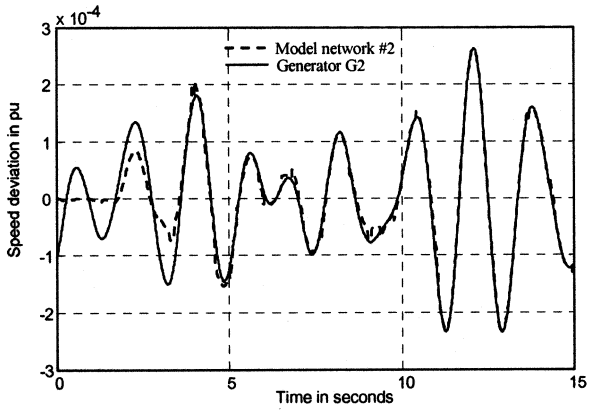

Fig. 11. Speed deviation of generator G2 and the Model network.

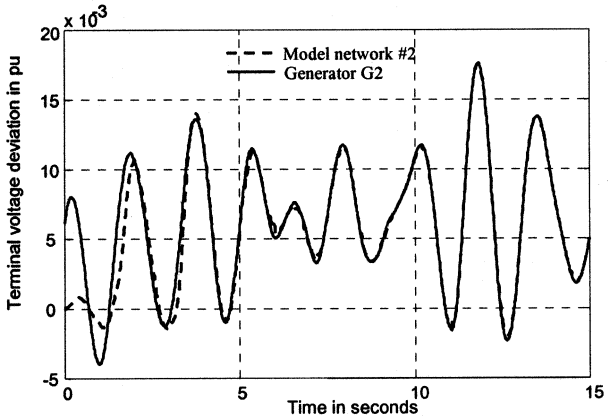

Fig. 12. Terminal voltage deviation of generator G2 and the Model network.

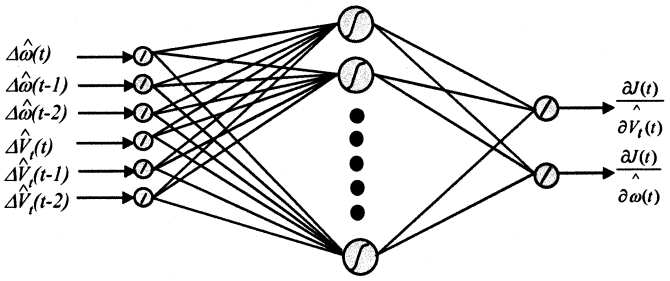

Fig. 13. DHP Critic neural network structure with six inputs, ten sigmoidal hidden layer neurons, and two linear neurons.

four previously delayed values, form the six inputs for the Critic network. The outputs of the Critic are the derivatives of the J function with respect to the output states of the generators.

In the Critic's training cycle, an incremental optimization of (3), is carried out using a suitable optimization technique such as backpropagation. The flowchart for the Critic neural network training is given in Fig. 14. The functions $f_C\,(\Delta\hat{Y}\,(t, t-1, t-2), W_C)$, $f_A\,(\Delta Y(t, t-1, t-2), W_A)$, and $f_M\,(\Delta Y(t, t-1, t-2), A(t, t-1, t-2), W_M)$ represent the Critic, the Action, and the Model neural networks with their weights W_i, respectively.

The Critic neural network's error and weight update equations are given in (23) and (24) with a discount factor γ of

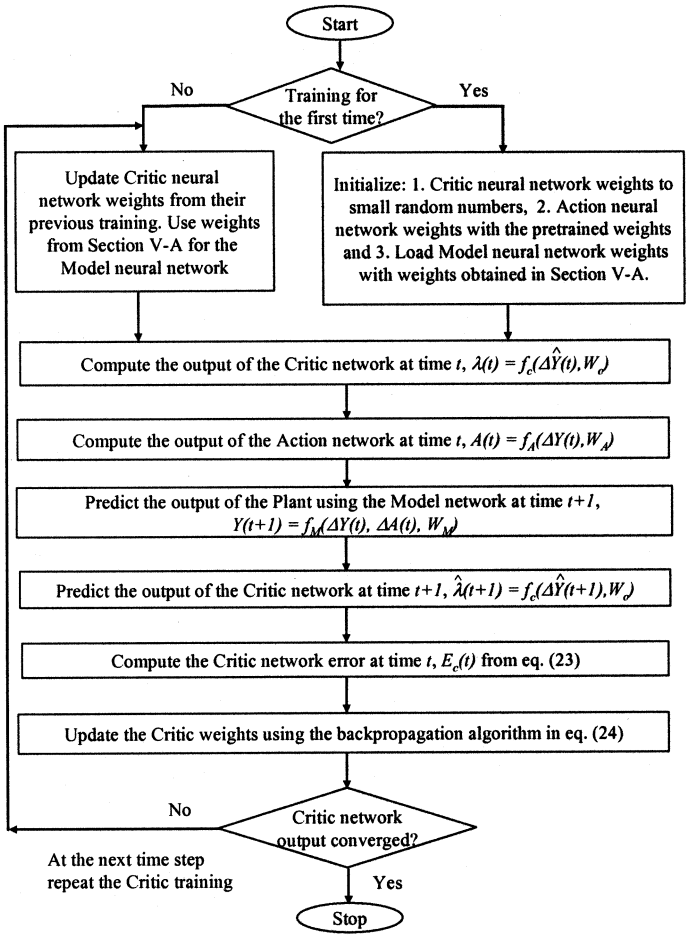

Fig. 14. Flowchart for the DHP Critic neural network training.

0.5, learning rate $\alpha = 0.03$, and the utility function given in (22). The Critic training is carried out for N_C cycles until the weights of the network have converged. W_C is initialized to small random values at beginning of the training

$$E_C(t) = \frac{\partial J\left[\Delta Y(t)\right]}{\partial \Delta Y(t)} - 0.5\,\frac{\partial \hat{J}\left[\Delta \hat{Y}(t+1)\right]}{\partial \Delta Y(t)} - \frac{\partial U(t)}{\partial \Delta Y(t)} \tag{23}$$

$$\Delta W_C = -0.03\left(\frac{\partial J\left[\Delta Y(t)\right]}{\partial \Delta Y(t)} - 0.5\,\frac{\partial \hat{J}\left[\Delta \hat{Y}(t+1)\right]}{\partial \Delta Y(t)} - \frac{\partial U(t)}{\partial \Delta Y(t)}\right) \times \frac{\partial}{\partial W_C}\left(\frac{\partial J\left[\Delta Y(t)\right]}{\partial \Delta Y(t)} - 0.5\,\frac{\partial \hat{J}\left[\Delta \hat{Y}(t+1)\right]}{\partial \Delta Y(t)} - \frac{\partial U(t)}{\partial \Delta Y(t)}\right). \tag{24}$$

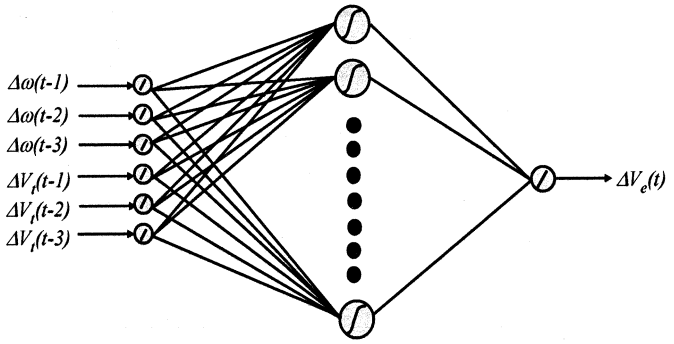

Fig. 15. DHP Action neural network structure with six inputs, ten sigmoidal hidden layer neurons, and one linear output neuron.

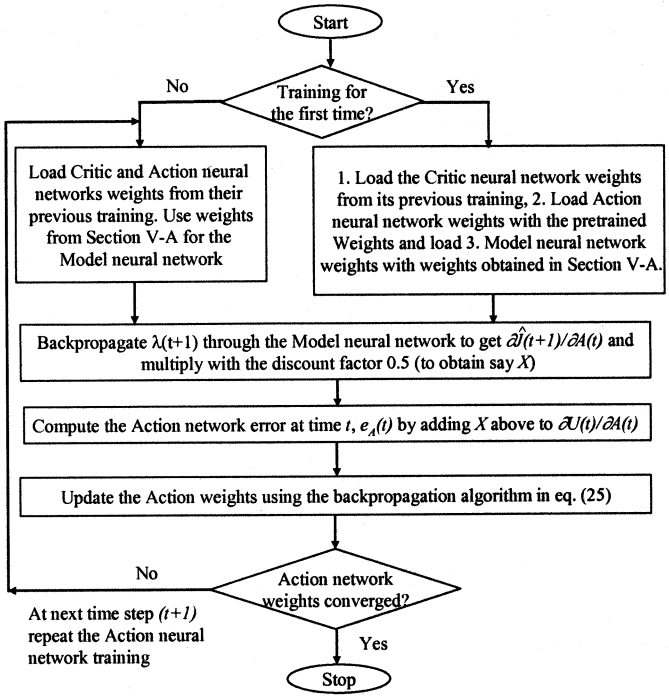

Fig. 16. Flowchart for the DHP Action neural network training.

C. Action Neural Network

The Action network (DHP neurocontroller) in Figs. 4 and 6 is also a three-layer feedforward network with six inputs, a single hidden layer with ten neurons, and a single output (Fig. 15). The inputs are the generator's *actual* speed and *actual* terminal voltage deviations, $\Delta\omega$ and ΔV_t, respectively. Each of these inputs is time delayed by 10 ms and, together with four previously delayed values, form the six inputs. The output of the Action network (DHP neurocontroller), $A(t) = [\Delta V_e(t)]$, the *deviation* in the field voltage, which augments the input to the generator's exciter.

The Action neural network weights' update expression, when applying backpropagation, is as follows:

$$\Delta W_A = -0.03\left[\frac{\partial U(t)}{\partial A(t)} + 0.5\,\frac{\partial \hat{J}(t+1)}{\partial A(t)}\right]^T \frac{\partial A(t)}{\partial W_A} \tag{25}$$

where 0.03 is the learning rate, and W_A contains the weights of the Action neural network in the DHP scheme. The flowchart for the training of the DHP Action neural network is shown in Fig. 16. The Action training is carried out for N_A cycles until

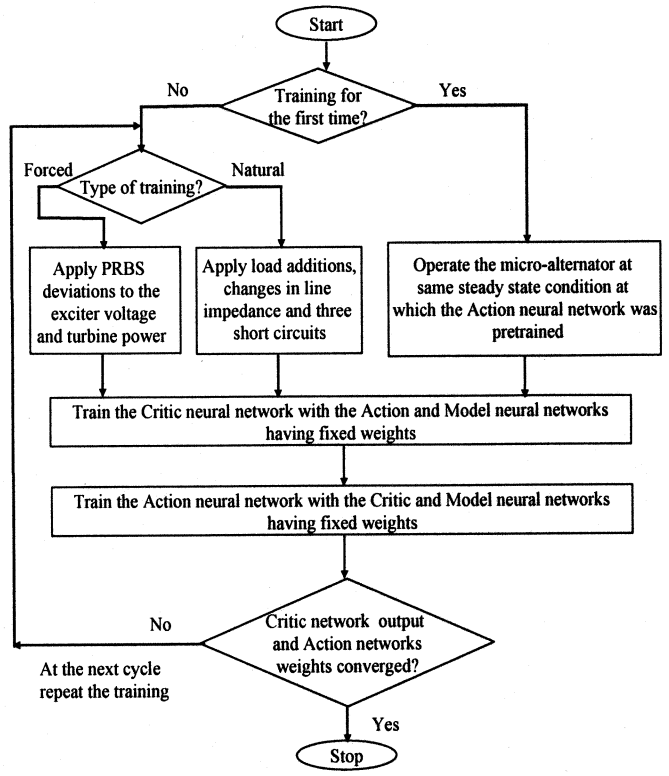

Fig. 17. Overall training steps for the DHP Critic and Action neural networks.

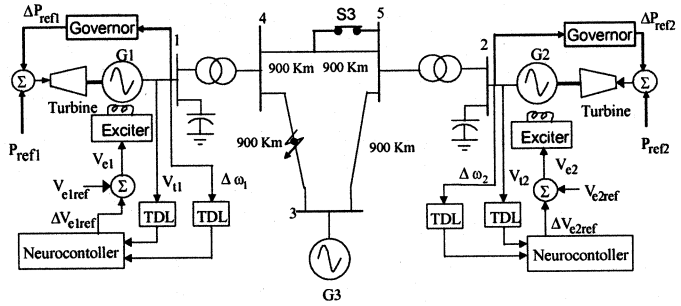

Fig. 18. Multimachine power system with trained DHP excitation neurocontrollers connected to generators G1 and G2

the weights of the network have converged. During the Action network training, the weights of the Critic network are fixed.

The overall training procedure of the DHP Critic and Action neural networks under the different types of training (forced and natural) are shown in the flowchart in Fig. 17. The training of the Critic and Action neural networks are alternated until both networks have attained training convergence over a wide range of system operating conditions and configurations. It is important that the whole system consisting of the neurocontroller and the system remains stable while both of the Critic and Action networks undergo training. Once the Critic network's and Action network's weights have converged, the Action network (neurocontroller) is connected to the generator G1 to replace the AVR (Fig. 18). A similar procedure is carried out in developing G2's DHP neurocontroller to replace its AVR.

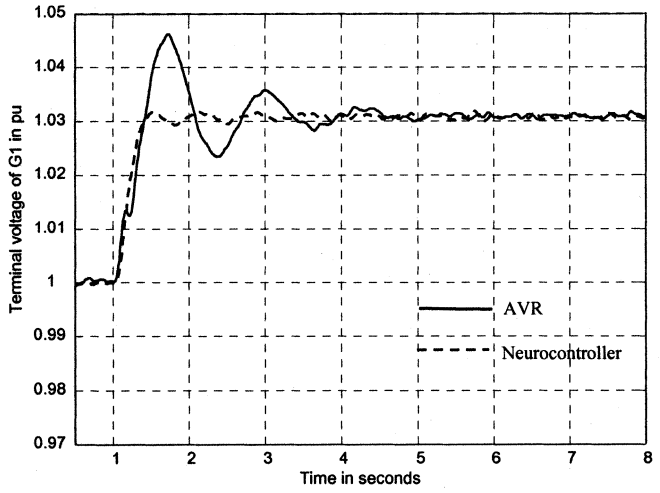

Fig. 19. Terminal voltage of generator G1 for a 3% step change in its desired terminal voltage.

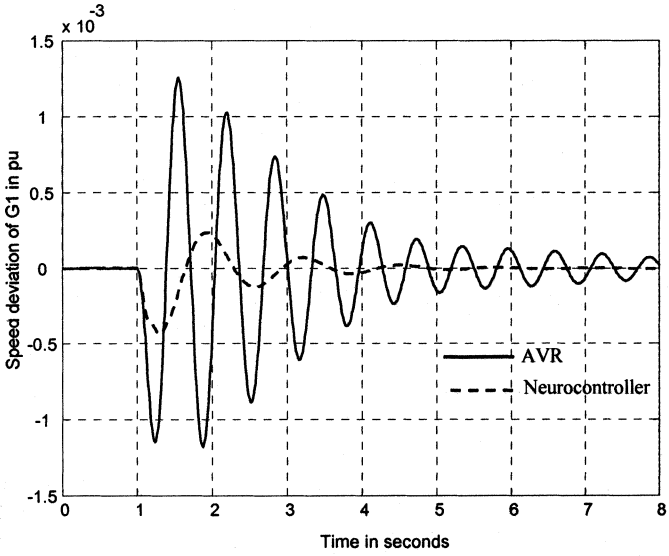

Fig. 20. Speed deviations of generator G1 for a 3% step change in its desired terminal voltage.

VI. Simulation and Experimental Results Performance With the DHP Controllers

At two different operating conditions and three different disturbances, the transient performance of the DHP neurocontrollers is compared with that of the conventional automatic voltage regulators (whose parameters are carefully tuned for the first set of operating condition given in Table IV [9]).

A. Simulation Study: 3% Step Change in $V_{e1\mathrm{ref}}$ (Fig. 18) and V_{ref1} (Fig. 1) at First Operating Condition

At the first operating condition (Table IV), a 3% step increase occurs in the desired terminal voltage of G1. Figs. 19 and 20 show that the DHP neurocontrollers ensure no overshoot on the terminal voltage and provide superior speed deviation damping unlike the AVRs, not withstanding the fact that the AVRs have been fine tuned at this operating condition.

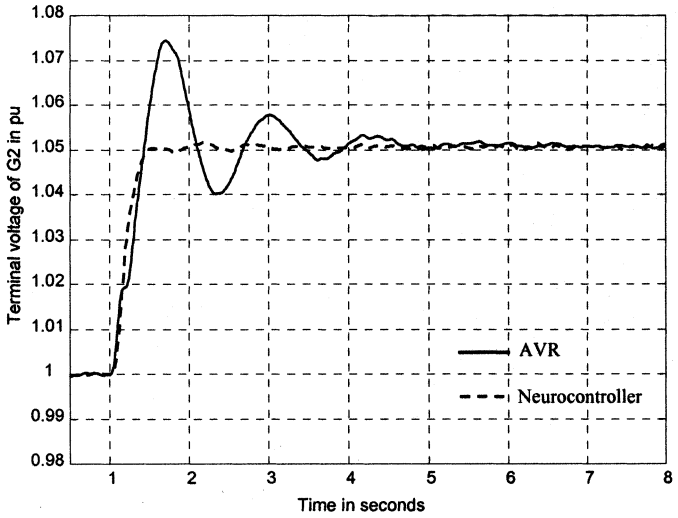

Fig. 21. Terminal voltage of generator G2 for a 5% step change in its desired terminal voltage.

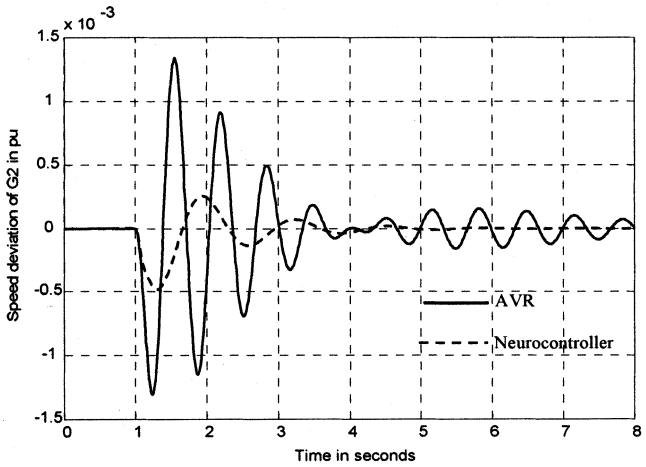

Fig. 22. Speed deviation of generator G2 for a 5% step change in its desired terminal voltage.

B. Simulation Study: 5% Step Change in $V_{e1\mathrm{ref}}$ (Fig. 18) and V_{ref1} (Fig. 1) at Second Operating Point

At the *second* operating condition (Table IV), a 5% step increase occurs in the desired terminal voltage of G2. Figs. 21 and 22 show that the DHP neurocontrollers again provide the best damping, which prove that the neurocontrollers have learned and adapted themselves to the new operating condition. In fact, Fig. 22 shows signs of an inter-area mode oscillations starting up at about 4.3 s, and the neurocontrollers are far more successful in damping this, than the conventional AVRs which were not fine tuned for this operating condition.

C. Simulation Study: Three Phase Short Circuit

At the second operating condition (Table IV), a 100–ms short circuit occurs halfway between buses 3 and 4 (Fig. 18). Figs. 23 and 24 show that the DHP neurocontrollers again have a better damping on the speed deviation and terminal voltage of G1. Although not shown, this is seen also in the speed deviation and the terminal voltage of G2.

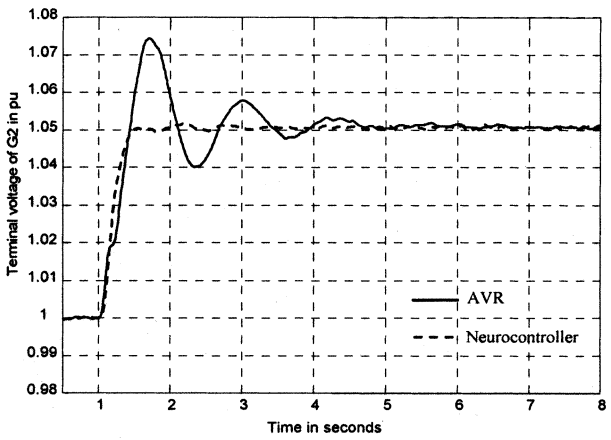

Fig. 23. Speed deviation of generator G1 for a 100-ms three-phase short circuit between buses 3 and 4.

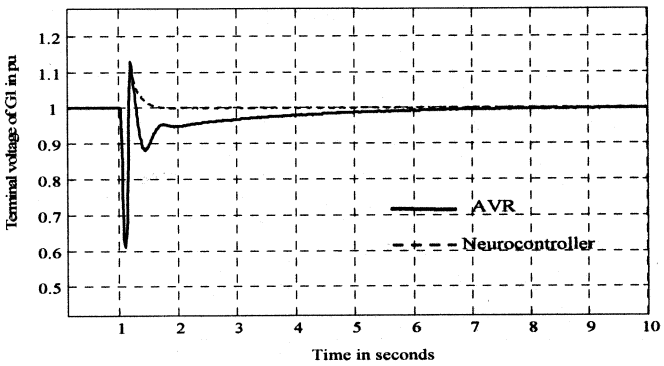

Fig. 24. Terminal voltage of generator G1 for a 100-ms three-phase short circuit between buses 3 and 4.

All these results show that at operating conditions different from the one at which the AVRs were tuned, their performance has degraded. On the other hand, the DHP neurocontrollers have given excellent performance under all the conditions tested. Many other tests, both small and large disturbances, were carried out at different power levels and power factors to confirm this.

D. Experimental Study: Increase in Transmission Line Impedance

The micromachine laboratory at the University of Natal, Durban, South Africa, has two micro-alternators, and each one represents both the electrical and mechanical aspects of a typical 1000-MW alternator. All the per-unit parameters are the same as those normally expected for 1000-MW alternators. The machine parameters have been determined by the standard IEEE methods and are given for micro-alternators in [8]. A practical laboratory three-machine power system is set up by using the two micro-alternators/turbogenerators and the infinite bus as the third machine.

At operating conditions $P = 0.2$ pu and $Q = 0$ pu for generators G1 and G2, the series transmission line impedance is increased at time $t = 10$ s from $Z = 0.0022 + j0.75$ (900 km) pu to $Z = 0.0044 + j1.50$ pu (1800 km) by opening switch S3 (Figs. 1 and 18). Figs. 25 and 26 show the rotor/load angle

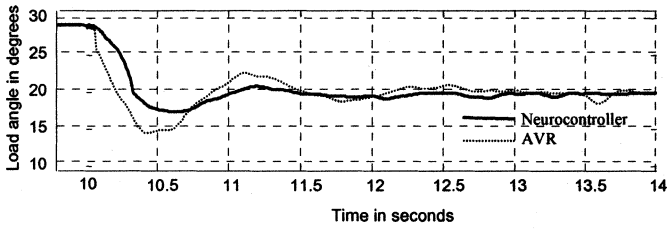

Fig. 25. Load angle response of generator G1 for increase in transmission line impedance between buses 3 and 4.

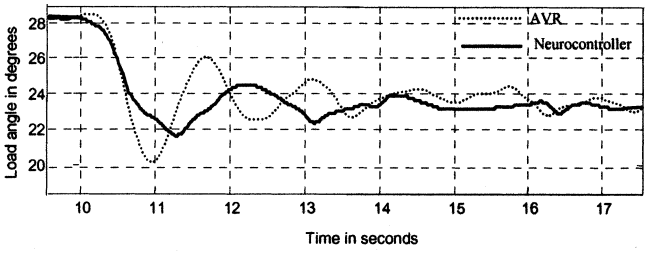

Fig. 26. Load angle response of generator G2 for increase in transmission line impedance between buses 4 and 5.

responses of generators for this test. Clearly, the DHP neurocontrollers again exhibit superior damping over the performance of the conventional controllers.

VII. Conclusions

A new design method for nonlinear optimal controllers based on derivative adaptive critics for voltage/excitation control of generators in a three-machine power system have been presented. All control variables are based on local measurements, thus, the control is decentralized. Simulations show that dynamic response of the DHP-based neurocontrolled generators are superior to the response of the conventionally controlled generators with AVRs, particularly so when operating conditions change and large disturbances are experienced. Furthermore, it has been shown both by simulation and practical implementation that it is possible to have multiple optimal neurocontrollers on a power system with no requirement for continual online training, thus avoiding the risk of system instability with neurocontrollers.

Acknowledgment

The authors acknowledge the University of Natal, Durban, South Africa, for allowing the usage of the micromachines laboratory.

References

[1] Y. Wang, D. J. Hill, L. Gao, and R. H. Middleton, "Transient stability enhancement and voltage regulation of power system," *IEEE Trans. Power Syst.*, vol. 8, pp. 620–627, May 1993.

[2] J. W. Chapman, M. D. Ilic, C. A. King, L. Eng, and H. Kaufman, "Stabilizing a multimachine power system via decentralized feedback linearizing excitation control," *IEEE Trans. Power Syst.*, vol. 8, pp. 830–839, Aug. 1993.

[3] G. K. Venayagamoorthy and R. G. Harley, "A continually online trained artificial neural network identifier for a turbogenerator," in *Proc. f IEEE IEMDC'99*, May 1999, pp. 404–406.

[4] ——, "Experimental studies with a continually online trained artificial neural network controller for a turbogenerator," in *Proc. IJCNN'99*, vol. 3, July 1999, pp. 2158–2163.

[5] 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.

[6] ——, "Adaptive neural network identifiers for effective control of turbogenerators in a multimachine power system," in *Proc. IEEE-PES Winter Meeting*, Jan. 2001, pp. 1293–1298.

[7] G. Sybille, P. Brunelle, R. Champagne, L. Dessaint, and H. Lehuy, *Power System Blockset, Version 2.0*. Natick, MA: The MathWorks Inc., 2000.

[8] D. J. Limebeer, R. G. Harley, and S. M. Schuck, "Subsychronous resonance of koeberg turbogenerators and of a laboratory micro-alternator system," *Trans. S. Afr. Inst. Elect. Eng.*, pp. 278–297, Nov. 1979.

[9] W. K. Ho, C. C. Hang, and L. S. Cao, "Tuning of PID controllers based on gain and phase margin specifications," in *Proc. 12th Triennial World Congr. Automatic Control*, 1993, pp. 199–202.

[10] P. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, White and Sofge, Eds. New York: Van Nostrand Reinhold, 1992, pp. 493–525.

[11] D. Prokhorov and D. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Networks*, vol. 8, pp. 997–1007, Sept. 1997.

[12] P. J. Werbos, *Roots of Backpropagation*. New York: Wiley, 1994.

[13] Q. H. Wu and B. W. Hogg, "Adaptive controller for a turbogenerator system," *Proc. IEE—Control Theory Applicat.*, pt. D, vol. 135, no. 1, pp. 35–42, 1988.

[14] G. K. Venayagamoorthy and R. G. Harley, "Two separate continually online trained neurocontrollers for excitation and turbine control of a turbogenerator," *IEEE Trans. Ind. Applicat.*, vol. 38, pp. 887–893, May/June 2002.

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 50 papers published in refereed journals and international conference proceedings.

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

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, and the Ph.D. degree from London University, London, U.K., in 1960, 1965, and 1969, respectively.

In 1970, he was appointed to the Chair of Electrical Machines and Power Systems, University of Natal, Durban, South Africa. He is currently with Georgia Institute of Technology, Atlanta. 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 in 1994. He has coauthored some 250 papers published in refereed journals and international conference proceedings. His research interests are in the dynamic and transient behavior of electric machines and power systems, and controlling them by the use of power electronics and modern control algorithms.

Dr. Harley was elected as a Distinguished Lecturer by the IEEE Industry Applications Society for the years 2000 and 2001. He is a Fellow of the South Africa Institute of Electrical Engineers, Institution of Electrical Engineers, U.K., Royal Society in South Africa, and University of Natal, and a Founder Member of the Academy of Science in South Africa formed in 1994. He is the recipient of several prize paper awards.

Donald C. Wunsch (SM'94) received the B.S. degree in applied mathematics from the University of New Mexico, Albuquerque, in 1984, and the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, in 1987 and 1991, respectively.

Since 1999, he has been the M. K. Finley Missouri Distinguished Professor of Computer Engineering in the Department of Electrical and Computer Engineering, University of Missouri, Rolla, where he also heads the Applied Computational Intelligence Laboratory. Previously, he was an Associate Professor at Texas Tech University, Lubbock. Prior to joining Texas Tech University in 1993, he was Senior Principal Scientist at Boeing, where he invented the first optical ART1 neural network and engaged in other applied research. He also worked for International Laser Systems and Rockwell International, and consulted for Sandia Laboratories, White Sands, and Accurate Automation Corporation. His current research includes adaptive critic designs; neural network optimization, forecasting and control; bioinformatics; and intrusion detection. He has authored over 150 publications on computational intelligence and has attracted in excess of $3 million in competitively awarded sponsored research funding since 1994, and more than $1 million since coming to the University of Missouri, Rolla.

Prof. Wunsch is an Academician in the International Academy of Technological Cybernetics, and was a recipient of the Halliburton Award for excellence and a National Science Foundation CAREER Award. He is a member of the International Neural Network Society, Association for Computing Machinery, a Life Member of the American Association for Artificial Intelligence, and previously served as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and as a voting member of the IEEE Neural Networks Council. He is General Chair of the International Joint Conference on Neural Networks (IJCNN) 2003, Portland, OR.