Pergamon

Neural Networks, Vol 7, No 1, pp 183–194, 1994

0893-6080/94 $6 00 + 00

CONTRIBUTED ARTICLE

Application of Neural Networks to Load-Frequency Control in Power Systems

FRANÇOISE BEAUFAYS,[1] YOUSSEF ABDEL-MAGID,[2] AND BERNARD WIDROW[1]

[1]Stanford University and [2]King Fahad University of Petroleum and Minerals

(*Received* 8 *June* 1992; *revised and accepted* 21 *June* 1993)

Abstract—*This paper describes an application of layered neural networks to nonlinear power systems control. A single generator unit feeds a power line to various users whose power demand can vary over time. As a consequence of load variations, the frequency of the generator changes over time A feedforward neural network is trained to control the steam admission valve of the turbine that drives the generator, thereby restoring the frequency to its nominal value. Frequency transients are minimized and zero steady-state error is obtained The same technique is then applied to control a system composed of two single units tied together through a power line Electric load variations can happen independently in both units. Both neural controllers are trained with the back propagation-through-time algorithm Use of a neural network to model the dynamic system is avoided by introducing the Jacobian matrices of the system in the back propagation chain used in controller training*

Keywords—Power system, Load-frequency control, Feedforward neural network, Back propagation-through-time.

1. INTRODUCTION

Control and stability enhancement of synchronous generators is of major importance in power systems. Different types of controllers based on classical linear control theory have been developed in the past (Elgerd, 1982; Anderson & Fouad, 1977; Debs, 1988; Wood & Wollenberg, 1984). Because of the inherent nonlinearity of synchronous machines, neural network techniques can be considered to build nonlinear controllers with improved performances.

In this paper, we first consider a single, isolated, generator unit connected to a power line or *electric bus* that serves different users. Variations in the power demand of the users cause the electric load on the bus to change over time. As the load varies, the frequency of the generator unit varies. To bring the steady-state frequency back to its nominal value after a given load variation, a control system is designed that acts on the setting of the steam admission valve of the unit turbine. It is of great importance to eliminate frequency transients as rapidly as possible. Most load-frequency control systems are primarily composed of an integral controller. The integrator gain is set to a level that compromises between fast transient recovery and low overshoot in the dynamic response of the overall system (Elgerd, 1982). This type of controller is slow and does not allow the control designer to take into account possible nonlinearities in the generator unit.

We propose in this paper a neural network load-frequency controller. The neural network makes use of a piece of information that is not used in conventional controllers: an estimate of the electric load perturbation (i.e., an estimate of the change in electric load when such a change occurs on the bus). This load perturbation estimate could be obtained either by a linear estimator or by a nonlinear neural network estimator. In certain situations, it could also be measured directly from the bus. We will show by simulation that when a load estimate is available, the neural network can achieve extremely good dynamic response.

The same neural network technique is then extended to control a two-area system (i.e., two generator units linked together by a tie-line) where electric load perturbations can happen independently in both areas. The two neural controllers are adapted using the back propagation-through-time algorithm (Nguyen & Widrow, 1989, 1990; Werbos, 1990). The system to be controlled, the *dynamic plant,* is represented by its state space equations. An error signal is defined at the output of the dynamic plant as the difference between the state vector of the plant and a desired state vector. Back propagation of the error vector through the plant state space equations is effected by means of a multiplication

Requests for reprints should be sent to Bernard Widrow, Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305-4055.

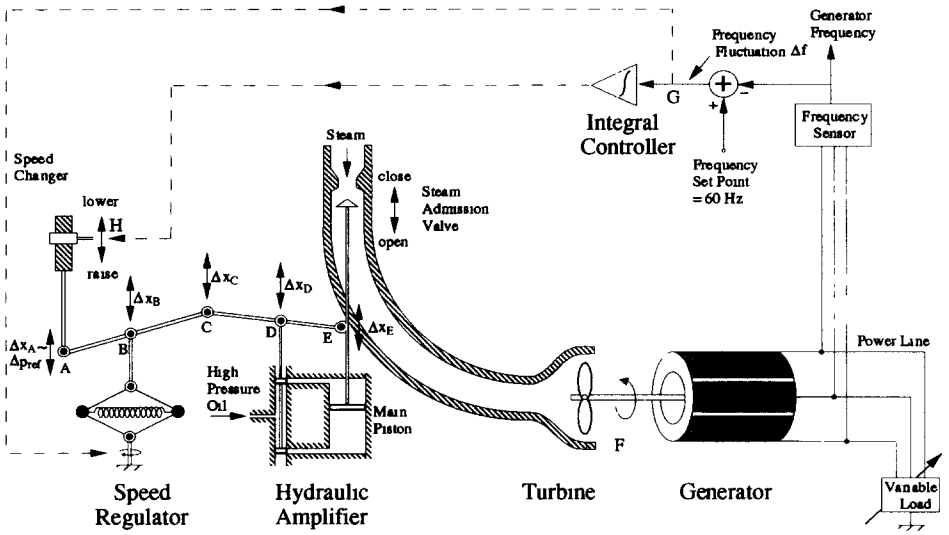

FIGURE 1. Conventional single-area system: simplified functional diagram.

of the error vector with the Jacobian matrix of the plant (a matrix containing the derivatives of the elements of the state transition matrix with respect to the state variables of the plant) (Piche & Widrow, 1991). The resulting signal is then back propagated through the neural network controller, and the adaptive weights of the controller are adjusted according to the back propagation algorithm. Introducing Jacobian matrices in the back propagation chain can be done whenever the state space equations of the plant are known *a priori*, and this avoids the introduction and training of a neural network plant model.

The next section of this paper describes the systems under study and shows how conventional integral controllers are used to control such plants. Following sections describe the neural networks used to control the plants and present simulation results. Models, equations, and orders of magnitude of parameters are in accord with examples in the classic book by O. Elgerd, *Electric Energy Systems Theory* (Elgerd, 1982, chap. 9, sections 9.3 and 9.4).

2. PLANT MODELS AND CONTROL BY CONVENTIONAL MEANS

2.1. Automatic Load-Frequency Control in Single-Area Systems

In a single-area system,[1] mechanical power is produced by a turbine and delivered to a synchronous generator serving different users. The frequency of the current and voltage waveforms at the output of the generator is mainly determined by the turbine steam flow. It is also affected by changes in user power demands that appear, therefore, as electric perturbations. If, for example, the electric load on the bus suddenly increases, the generator shaft slows down, and the frequency of the generator decreases. The control system must immediately detect the load variation and command the steam admission valve to open more so that the turbine increases its mechanical power production, counteracts the load increase, and brings the shaft speed and hence the generator frequency back to its nominal value.

A simplified functional diagram of a conventionally controlled power system is shown in Figure 1. A brief explanation of the diagram follows (a more detailed description can be found in Elgerd, 1982).

Steam enters the turbine through a pipe that is partially obstructed by a steam admission valve. In steady-state, the opening of the valve is determined by the position of a device called the *speed changer* (upper left corner in Figure 1). Its setting (points H or A in Figure 1) fixes the position of the steam valve through two rigid rods ABC and CDE. The reference value, or set-point, of the turbine power in steady-state is called the *reference power* and is denoted by p_{ref}. When the load on the bus suddenly changes, the shaft speed is modified, and a device called the *speed regulator* acts through the rigid rods to move the steam valve. Note that a similar effect could be producted by temporarily modifying the reference power (which justifies the name speed changer). In practice, both control schemes are used simultaneously. Amplifying stages (generally hydraulic) are introduced to magnify the outputs of the controllers and produce the forces necessary to actually move the steam valve.

The speed regulator is a proportional controller of gain $1/R$ (that is, the deflection in B, Δx_B, is proportional to $1/R$ times the frequency fluctuation Δf). In

[1] For sake of clarity, we restrict the concept of *area* to pertain to a system containing one single generator although, in practice, the word area generally refers to a system containing many parallel-working generators (Elgerd, 1982)

conventional systems, an integral controller of gain K_I sums the frequency fluctuations Δf (point G) and uses the result (point H) as a control signal to the speed changer to raise or lower the reference power. By combining these two control loops, we get a parallel PI (proportional-integral) controller capable of driving frequency fluctuations to zero whenever a step-load perturbation is applied to the system (Elgerd, 1982).

Because most devices in power systems are extremely nonlinear, one usually likes to linearize the plant and to think of different variables in terms of their fluctuations about given operating points. Nonlinearities are then modeled by making the parameters of the linearized system functions of the operating point. The resulting small signal models consist of linear operators having variable parameters whose values depend upon the state of the system. The last step in modeling consists of replacing all small signals by their Laplace transform and to represent the linearized devices by transfer functions.

Before going any further, let us define the notation. Δ of some variable represents the difference between the variable and its nominal value. Lowercases are used for time signals, and uppercases for their Laplace transforms. For example, $\Delta f(t)$ represents the generator frequency relative to its nominal value, that is, $\Delta f(t) = f(t) - f_{\text{nominal}} = f(t) - 60$ Hz, and $\Delta F(s)$ represents the Laplace transform of $\Delta f(t)$.

A Laplace domain small signal model of the single-area system is given in Figure 2. Points A, B, C, D, E, F and G of Figure 1 are also shown here to help relate the figures. Starting from point A in Figure 1, the fluctuation in reference power Δp_{ref} (i.e., the output of the speed changer) is added to the output of the speed regulator (point B) to produce a global control signal (point C), which is instantaneously transmitted to the input of the hydraulic amplifier (point D). For simplicity, the hydraulic amplifier is modeled by a first-order transfer function whose output is the fluctuation in hydraulic amplifier power $\Delta P_H(s)$. The turbine is also modeled by a first-order transfer function. Its input is the hydraulic amplifier output $\Delta P_H(s)$, its output is the turbine mechanical power $\Delta P_T(s)$. The change in variable load (lower right corner in Figure 1) is symbolized by an electric perturbation, $\Delta P_E(s)$, and can be modeled as an input perturbation to the generator

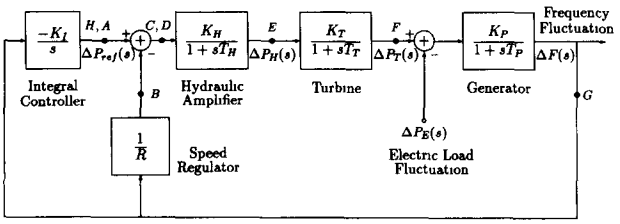

FIGURE 2. Conventional single-area system: small-signal model, simplified block diagram.

(Elgerd, 1982). The input to the generator model in Figure 2 is thus the sum of the turbine output power and the electric perturbation. The generator is modeled by a first-order transfer function. Its output, the frequency fluctuation $\Delta F(s)$ (point G), is used to drive the speed regulator and the integral controller.

To simulate the dynamic plant in a C-programming environment, we derive its discrete-time state space equations. Referring to Figure 2 and expressing the outputs of the generator, turbine, and amplifier as functions of their inputs, inverting the Laplace transforms, and discretizing the time functions, we obtain the following discrete-time state space equations:

$$\Delta f(nT_s + T_s) = \Delta f(nT_s) + \frac{T_s}{T_P}[K_P\,\Delta p_T(nT_s) - K_P\,\Delta p_E(nT_s) - \Delta f(nT_s)] \quad (1)$$

$$\Delta p_T(nT_s + T_s) = \Delta p_T(nT_s) + \frac{T_s}{T_T}[K_T\,\Delta p_H(nT_s) - \Delta p_T(nT_s)] \quad (2)$$

$$\Delta p_H(nT_s + T_s) = \Delta p_H(nT_s) + \frac{T_s}{T_H}[K_H\,\Delta p_{\text{ref}}(nT_s) - \frac{K_H}{R}\Delta f(nT_s) - \Delta p_H(nT_s)] \quad (3)$$

with

$$\Delta p_{\text{ref}}(nT_s) = \Delta p_{\text{ref}}(nT_s - T_s) - K_I\,\Delta f(nT_s). \quad (4)$$

T_s is the sampling period and n is the discrete-time index.

Typical orders of magnitude in large systems ($\approx$1000 MW) are: for the gains of the turbine, hydraulic amplifier, and generator, $K_H = K_T = 1.0$, $K_P = 120$ Hz/pu MW [2]; for the corresponding time constants, $T_H = 80$ ms, $T_T = 0.3$ s, $T_P = 20$ s; for the regulator gain, $R = 2.4$ Hz/pu MW.

For any nonnegative value of the integrator gain K_I, and assuming that the perturbation $\Delta P_E(s)$ is a step function of amplitude ΔP_E, the controlled plant is stable; that is, its state vector $\mathbf{x}(nT_s) = [\Delta f(nT_s)\ \Delta p_T(nT_s)\ \Delta p_H(nT_s)]^T$ converges to a finite steady-state value. It is easy to see from eqns (1)–(4) that the steady-state frequency fluctuation $\Delta f(nT_s)$ converges to zero. Therefore (cf. Figure 2), the output turbine power $\Delta p_T(nT_s)$ converges towards ΔP_E, and the amplifier output power $\Delta p_H(nT_s)$ converges towards $\Delta P_E/K_T$. The steady-state state vector is thus:

$$\mathbf{x}_{\text{steady-state}} = \left[0\ \ \Delta P_E\ \ \frac{\Delta P_E}{K_T}\right]^T. \quad (5)$$

Figure 3 shows, for different values of the integrator gain K_I, the dynamic responses of a single-area system

[2] In the *per unit* (pu) system, variables are scaled by their nominal value and become thereby dimensionless

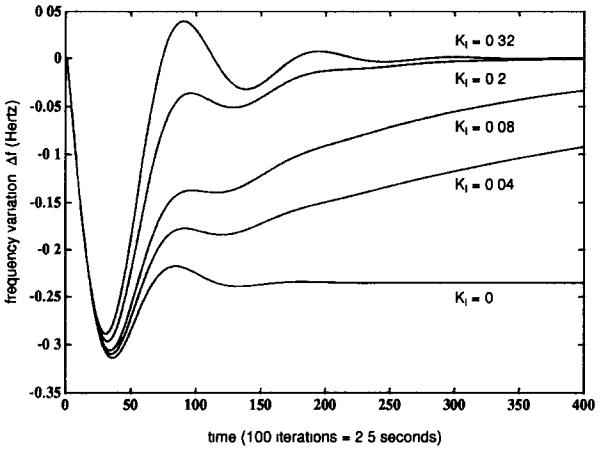

FIGURE 3. Dynamic responses of a conventional single-area system subject to a 10% step-load perturbation, for different values of the integrator gain.

subject to a 10% step-load increase. With no integral control ($K_I = 0$), the frequency fluctuations $\Delta f(nT_s)$ converge to some non-zero steady-state value. To achieve zero steady-state error, the integrator must have a strictly positive gain. The higher the gain, the faster the convergence will be, but high gains tend to produce ringing in the step response, and this should be avoided for stability reasons. In practice, K_I will be chosen to be the *critical gain,* that is, the highest gain that yields no overshoot ($K_I = 0.2$ in Figure 3).

2.2. Automatic Load-Frequency Control in Two-Area Systems

A two-area system consists of two single-area systems connected through a power line called the *tie-line*: each area feeds its user pool, and the tie-line allows electric power to flow between the areas. Because both areas are tied together, a load perturbation in one area affects the output frequencies of both areas as well as the power flow on the tie-line. For the same reason, the control system of each area needs information about the transient situation in both areas to bring the local frequency back to its steady-state value. Information about the local area is found in the output frequency fluctuation of that area. Information about the other area is found

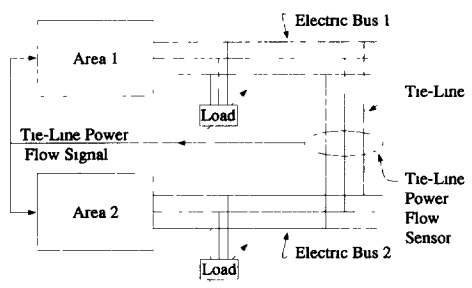

FIGURE 4. Conventional two-area system: basic block diagram.

in the tie-line power fluctuations. Therefore, the tie-line power is sensed, and the resulting tie-line power signal is fed back into both areas. This basic scheme is illustrated in Figure 4. A more complete diagram is given in Figure 5. One recognizes the two single-area block diagrams (dashed boxes) and the tie-line. A few additional elements are introduced whose function is next described.

In steady-state, each area outputs a frequency of 60 Hz. A load perturbation occurring in either area affects the frequencies in both areas as well as the tie-line power flow. In fact, it can be shown (Elgerd, 1982) that, with small signal approximation, the fluctuation in power exchanged on the tie-line, $\Delta p_{1,2}(t)$, is proportional to the difference between the instantaneous shaft angle variations in both areas, $\Delta\theta_1(t)$ and $\Delta\theta_2(t)$ (Elgerd, 1982). These shaft angle variations are equal to 2π times the integral of the corresponding frequency variations, $\Delta f_1(t)$ and $\Delta f_2(t)$. In the Laplace domain,

$$\Delta P_{1,2}(s) = T^0[\Delta\Theta_1(s) - \Delta\Theta_2(s)]$$
$$= \frac{2\pi T^0}{s}[\Delta F_1(s) - \Delta F_2(s)], \qquad (6)$$

where T^0 is a constant called the tie-line synchronizing coefficient (typically 0.0707 MW/rad). This operation is illustrated in Figure 5 (right-hand part of the tie-line).

If, for example, the electric load increases in area 2 ($\Delta P_{E,2}(s) > 0$), the frequency in area 2 decreases ($\Delta F_2(s) < 0$), and more power is transmitted from area 1 to area 2 [$\Delta P_{1,2} > 0$, which is in accord with eqn (6)]. In area 1, this increase in tie-line power is perceived exactly the same way as an increase in power demand from the users of area 1, that is, an increase of δp in $P_{1,2}(s)$ or an increase of δp in $P_{E,1}(s)$ has the same effect on the frequency $F_1(s)$. Therefore, in our model, $\Delta P_{1,2}(s)$ should be added to the same node as $\Delta P_{E,1}$, and with the same sign (which is a minus sign). By symmetry, $\Delta P_{2,1} = -\Delta P_{1,2}$, the power going from area 2 to area 1, is added to the same node as $\Delta P_{E,2}$, also with a minus sign.

Let us now examine how two-area systems are controlled. In conventional systems, the turbine reference power of each area is set by an integral controller. Because a perturbation in either area affects the frequency in both areas and a perturbation in one area is perceived by the other through a change in tie-line power, the controller of each area should take as input not only the local frequency variations, but also the tie-line power variations. Because an integral controller has just one input, these two contributions (local frequency variation and tie-line power variation) must be combined into a single signal that can be inputted in the controller. The easiest way of doing this is to combine them linearly, that is, the input of the integrator in area 1 is $\Delta P_{1,2} + B_1\Delta F_1$, and the input of the integrator in

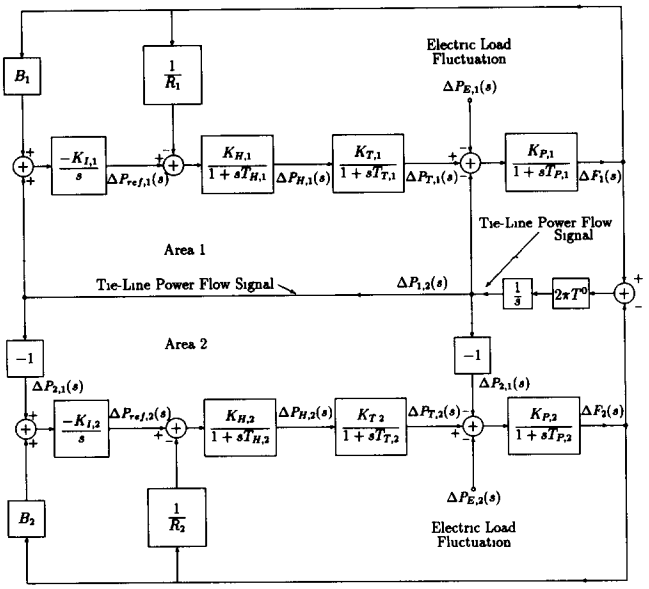

FIGURE 5. Conventional two-area system: small signal model, simplified block diagram.

area 2 is $\Delta P_{2,1} + B_2 \Delta F_2$ (see Figure 5). The coefficients B_1, B_2 are usually set to $1/K_P + 1/R$ (see Elgerd, 1982, for more details). With the orders of magnitude previously mentioned, $B_1 = B_2 = 0.425$ pu MW/Hz.

Following the same procedure as in the one-area case, we derive the discrete-state space equations of a two-area system. Assume for sake of simplicity that the hydraulic amplifier and turbine time constants are negligible compared to the generator time constants. Choose as discrete state variables the frequency in area 1, $\Delta f_1(nT_s)$, the frequency in area 2, $\Delta f_2(nT_s)$, and the tie-line power, $\Delta p_{1,2}(nT_s)$. The equations are:

$$\Delta f_1(nT_s + T_s) = \Delta f_1(nT_s) + \frac{T_s}{T_{P,1}}\left[K_{P,1}\Delta p_{\text{ref},1}(nT_s) - \left(\frac{K_{P,1}}{R_1} + 1\right)\Delta f_1(nT_s) - K_{P,1}\Delta p_{E,1}(nT_s)\right], \quad (7)$$

$$\Delta f_2(nT_s + T_s) = \Delta f_2(nT_s) + \frac{T_s}{T_{P,2}}\left[K_{P,2}\Delta p_{\text{ref},2}(nT_s) - \left(\frac{K_{P,2}}{R_2} + 1\right)\Delta f_2(nT_s) - K_{P,2}\Delta p_{E,2}(nT_s)\right], \quad (8)$$

$$\Delta p_{1,2}(nT_s + T_s) = \Delta p_{1,2}(nT_s) + T_s[2\pi T^0(\Delta f_1(nT_s) - \Delta f_2(nT_s))]. \quad (9)$$

Once again, T_s is the sampling rate. For large systems ($\approx$1000 MW), typical parameter values are: for the generator gains, $K_{P,1} = K_{P,2} = 120$ Hz/pu MW; for the generator time constants, $T_{P,1} = T_{P,2} = 20$ s; for the regulation parameters, $R_1 = R_2 = 2.4$ Hz/pu MW.

For any nonnegative integrator gains $K_{I,1}, K_{I,2}$, when a step-load perturbation has occurred in area 1 and/or in area 2, and after transients have died out, the frequency variations in both areas converge to zero, and so does the tie-line power [see eqn (6)]. The plant state vector $\mathbf{x}(nT_s) = [\Delta f_1(nT_s)\ \Delta f_2(nT_s)\ \Delta p_{1,2}(nT_s)]^T$ converges thus to a steady-state value equal to:

$$\mathbf{x}_{\text{steady-state}} = [0\ 0\ 0]^T. \quad (10)$$

Figure 6 shows the dynamic response of a conventional two-area system subject to a 10% step-load increase in area 2. Figure 6A shows the frequency transients in both areas and Figure 6B shows the tie-line power transients. We may observe here that zero integrator gains (dashed lines) yield non-zero steady-state errors, while positive integrator gains (plain lines) do achieve convergence of the state variables to zero. In this specific plot, the gains were chosen equal to the critical gains ($K_{I,1} = K_{I,2} = K_{\text{critical}} \simeq 0.05$), which ensured the fastest transient recovery with no overshoot in the step response.

Before leaving our discussion of conventional control systems and beginning a development of neural controls, we make an important remark. As explained in Section 2.1, the plant models used so far were linearized models. We were assuming that the operating point of the plant did not change much when a step-load perturbation occurred on the bus, and that, therefore, all the plant parameters could be kept constant. In practice though, the constant characterizing the speed regulator R depends in a highly nonlinear way upon the turbine power p_T, and this occurs as well in a single-area system as in each area of a two-area system (Cohn, 1986). The presence of this nonlinearity and the slowness of traditional integral controllers is what motivates our substituting a neural network controller for the integrator(s). To emphasize the difference in plant dy-

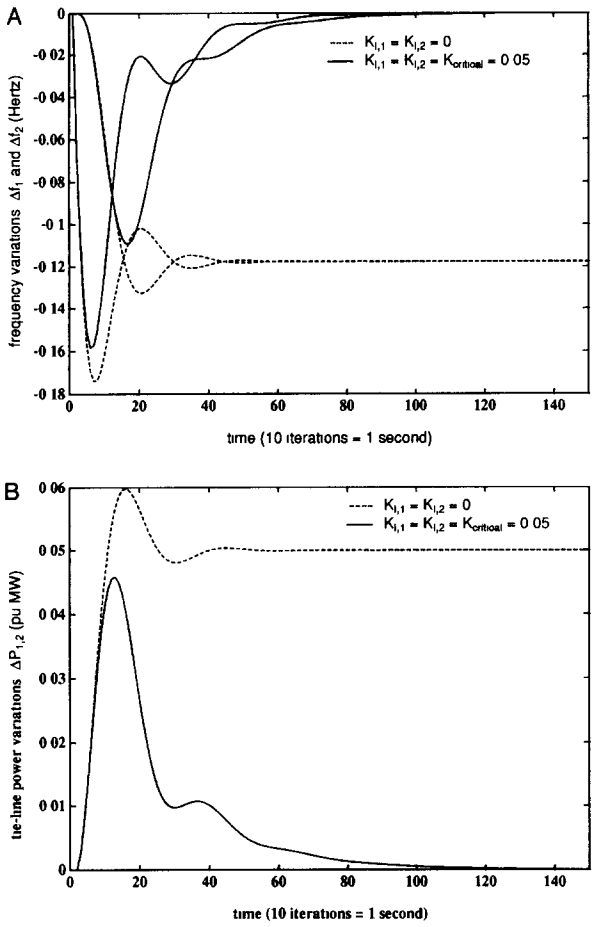

FIGURE 6. Dynamic response of a two-area system subject to a 10% step-load increase in area 2. (A) Frequency transients in both areas. (B) Tie-line power transients.

namics for various values of the regulation parameter R, we plotted in Figure 7 the step responses of an uncontrolled two-area system (integrator gains equal to zero) for two different values of R (chosen within a reasonable physical range). Figure 7A shows the frequency transients in both areas when a 10% step perturbation hits area 2 of a system having as regulation parameters $R_1 = R_2 = 2.4$ Hz/pu MW. Figure 7B shows the frequency transients for a system having $R_1 = R_2 = 6.0$ Hz/pu MW. Apart from the fact that the steady-state frequencies are different for different values of R, we notice that a higher regulation parameter creates more ringing in the system: the fact that R is not a constant profoundly affects the dynamic response of both generators.

3. NONLINEAR CONTROL BY MEANS OF NEURAL NETWORKS

3.1. Neural Network Control of a Single-Area System

We have seen in the previous section that the slowness and lack of efficiency of conventional controllers in handling system nonlinearities suggested their substitution by nonlinear neural network controllers. A natural choice of neural network architecture for a dynamic controller is the feedforward multilayer structure. Such an architecture can be adapted with the back propagation-through-time algorithm (Nguyen & Widrow, 1989, 1990; Werbos, 1990), an extension of the well-known back propagation algorithm (Werbos, 1974; Rumelhart & McClelland, 1986).

Let us first briefly summarize the back propagation algorithm.

3.1.1. *Back Propagation Algorithm* A feedforward multilayer neural network is shown in Figure 8A. A single neuron extracted from the lth layer of a L-layer neural network is represented in Figure 8B. The inputs x_i^l are multiplied by the adaptive weights $w_{i,j}^l$; the output x_j^{l+1} is obtained by passing the sum of the weighted inputs through a sigmoidal function $s_j^l(\cdot)$ (i.e., hyperbolic tangent)

$$x_j^{l+1} = s_j^l\left(\sum_i w_{i,j}^l x_i^l\right) \tag{11}$$

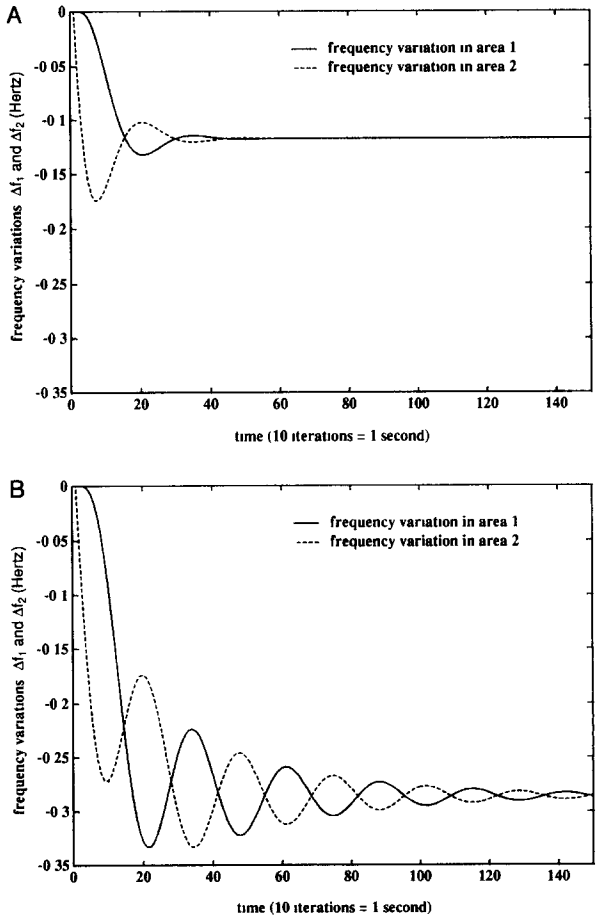

FIGURE 7. Dynamic response of an uncontrolled two-area system subject to a 10% step-load increase in area 2, for different values of R. (A) $R_1 = R_2 = 2.4$ Hz/pu MW. (B) $R_1 = R_2 = 6.0$ Hz/pu MW.

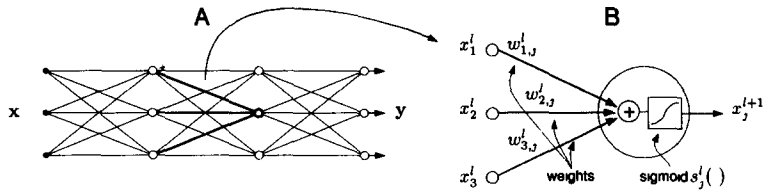

FIGURE 8. Feedforward multilayer neural network. (A) L-layer neural network. (B) jth neuron extracted from the lth layer.

Initially set to small random values, the weights are adjusted after each presentation of a new input pattern. The adaptation rule is given by:

$$\Delta w_{i,j}^{l} = -\mu \frac{d(\mathbf{e}^T\mathbf{e})}{dw_{i,j}^{l}} \tag{12}$$

where $w_{i,j}^{l}$ is the weight connecting neuron i in layer l with neuron j in the next layer, μ is the learning rate, $\mathbf{e}$ is the error vector, that is, the difference between the actual and desired outputs. It can easily be shown (see Rumelhart & McClelland, 1986; Widrow & Lehr, 1990, for a detailed derivation) that eqn (12) is equivalent to:

$$\Delta w_{i,j}^{l} = -\mu \delta_j^{l+1} \cdot x_i^{l} \tag{13}$$

where x_i^{l} is the output of neuron i in layer l. The error gradients δ_j in a L-layer network are evaluated with the following recursive formula:

$$\delta_j^{l} = \begin{cases} -2e_j \cdot s_j'^{L} & l = L \\ s_j' \cdot \sum\limits_m \delta_m^{l+1} \cdot w_{j,m}^{l} & 1 \le l \le L-1 \\ \sum\limits_m \delta_m^{1} \cdot w_{j,m}^{0} & l = 0 \end{cases} \tag{14}$$

where e_j is the error at the output node j, and $s_j'^{l} \equiv s_j'^{l}(x_j^{l})$ is the derivative of the sigmoid function $s_j^{l}(\cdot)$ for node j in layer l. Equation (14) can be interpreted as follows: the error gradient δ associated with a given neuron is obtained by back propagating the δ's of the next layer through connecting weights. This process is illustrated in Figure 9A and B.

3.1.2. *Adaptation of the Neural Network Controller.* The feedforward neural network studied in the previous section was a static structure in the sense that no time dependency existed between inputs and desired outputs. The situation is different when the network to be adapted is embedded in a dynamic structure. In our application, for example, a neural network is used to control a dynamic plant. Initially in steady-state, the plant is suddenly hit by a step perturbation. Transients ensue. It has been shown above that the discrete state vector $\mathbf{x}(n) = [\Delta f(n)\ \Delta p_T(n)\ \Delta p_H(n)]^T$ of a single-area system[3] controlled by an integral controller eventually converged to a steady-state value equal to $\mathbf{x}_{\text{steady-state}} \equiv [0\ \Delta P_E\ (\Delta P_E/K_T)]^T$ [eqn (5)] but that this convergence was slow (see Figure 6). The neural network controller that replaces the integral controller should make the plant converge to the same $\mathbf{x}_{\text{steady-state}}$ vector, while limiting the duration and magnitude of the transients. Such an operation cannot be performed instantaneously. In addition, the value of the desired control signal (i.e., the desired output of the neural network) is not known a priori; only the desired steady-state of the plant is known. The simple, static, back propagation algorithm is not directly applicable; it must be generalized to the present dynamic structure. The result of this generalization is referred to as the back propagation-through-time algorithm.

The dynamic controller-plant structure is shown in Figure 10. The plant model (dashed box) is characterized, at any instant of time, by its discrete state space vector $\mathbf{x}(n) = [\Delta f(n)\ \Delta p_T(n)\ \Delta p_H(n)]^T$. The box labeled plant state equations contains the discrete state space equations previously derived to model the one-area system, eqns (1)–(3). The neural network controller replaces the integral controller used in conventional systems. Its output, the fluctuation in reference power, $\Delta p_{\text{ref}}(n)$, is used as a control signal to drive the plant model. For simplicity, it is referred to as $u(n)$.

To evaluate $u(n) = \Delta p_{\text{ref}}(n)$, the neural network controller makes use of a piece of information that is not used in conventional control: an estimate of the load perturbation, $\Delta \widehat{p_E}(n)$. In general, the load perturbation of a large system is not directly measurable. It must therefore be estimated by a linear estimator or by a nonlinear neural network estimator if the nonlinearities in the system justify it. Such an estimator takes as inputs a series of K samples of the frequency fluctuation at the output of the generator $[\Delta f(n)\ \Delta f(n-1) \ldots \Delta f(n-K+1)]^T$, and estimates the instantaneous value of the load perturbation $\Delta p_E(n)$ based on this input vector. The estimate $\Delta \widehat{p_E}(n)$ is then used to drive the plant controller. The implementation of such an estimator is beyond the scope of this paper. Here, we assume that the load estimate $\Delta \widehat{p_E}(n)$ is

[3] To shorten the notation, we set the sampling period T_s to 1 so that $\mathbf{x}(nT_s)$ is replaced by $\mathbf{x}(n)$, $\Delta f(nT_s)$ by $\Delta f(n)$, etc.

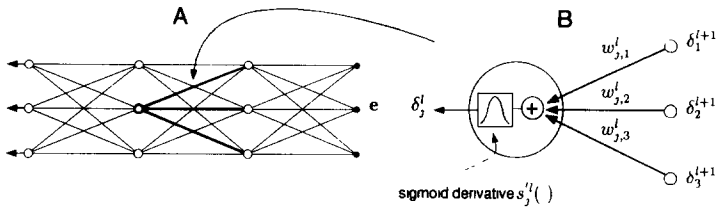

FIGURE 9. Feedforward multilayer neural network. (A) Back propagation through a L-layer neural network. (B) Back propagation through the *j*th neuron of the *l*th layer.

available and that, for adaptation purposes, this estimate is perfect, that is, $\Delta \widehat{p_E}(n) \equiv \Delta p_E(n)$. We also assume that the electric perturbation is a step function of amplitude ΔP_E: $\Delta p_E(n) = \Delta P_E \cdot H(n)$ where $H(n) = 1$ for $n \geq 0$ and 0 for $n < 0$.

The internal structure of the neural network controller is detailed in Figure 11. It contains two layers of neurons: a hidden layer and a single neuron output layer. No sigmoid is used at the output of the network. Bias terms (i.e., constant inputs equal to 1) are included in all the hidden units and in the output unit.

Going back to Figure 10, we see that at each time step, a new state vector $\mathbf{x}(n+1)$ is evaluated, based on the magnitude of the step load perturbation ΔP_E, the control signal $u(n)$, and the current state vector $\mathbf{x}(n)$. This is better understood if we redraw Figure 10, unravelling it in time. The resulting block diagram is shown in Figure 12. For simplicity, the neural network controller is denoted by C, and the plant state equations are denoted by P. At time $n = 0$ (defined as the instant when the step-load perturbation hits the system), the state of the plant is $\mathbf{x}(0)$. The neural network controller C takes as inputs the state vector $\mathbf{x}(0)$ and the amplitude of the step-perturbation ΔP_E, and quasi-instantaneously evaluates the control signal $u(0)$. The plant equations P are then used to compute the next state vector $\mathbf{x}(1)$, which is used by the controller to evaluate the next control signal $u(1)$, etc. This way, the controller and plant model successively compute new control signals and new state vectors until some time N is reached. N is chosen large enough to allow plant transients to die out. At time N, the plant state vector $\mathbf{x}(N)$ should have converged to $\mathbf{x}_{\text{steady-state}}$, which appears thus for the learning algorithm as the *desired* state vector, and is

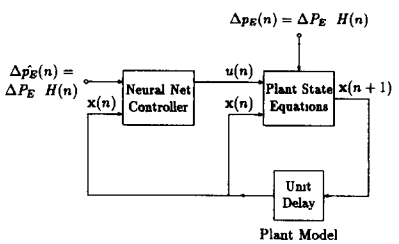

FIGURE 10. Neural network controller and plant model of a single-area system: block diagram.

denoted by $\mathbf{d}(N)$. The difference between the desired state vector $\mathbf{d}(N)$ and the final state vector $\mathbf{x}(N)$, the error vector $\mathbf{e}(N)$, is used to adjust the adaptive weights of the controller according to the back propagation-through-time algorithm. Although we are controlling the plant continuously over time, the control law is derived as if the process were only ongoing over the last N time steps, with N large. The weights are adjusted to reduce the error at time N.

3.1.3. *Back Propagation-Through-Time Algorithm.* Figure 12 is essentially composed of a layered arrangement of controller and plant equations blocks. The controller is a neural network. If the block containing the plant equations, P, were replaced by a neural network copy $\hat{P}$ of P[4], the unravelled system of Figure 12 would become a giant layered neural network with inputs $\mathbf{x}(0)$ and ΔP_E, output $\mathbf{x}(N)$, and desired output $\mathbf{d}(N)$. The back propagation algorithm could then be applied to train such a network. By doing so, the error gradient defined at the output of the network are back propagated through the $\hat{P}$ and C blocks, from $\mathbf{x}(N)$ back to $\mathbf{x}(0)$; hence, the name back propagation-through-time. This approach was first introduced by Nguyen and Widrow (1989), and was successfully applied to number of applications in the area of nonlinear neural control (see for example Nguyen & Widrow, 1990; Wu, Hogg, & Irwin, 1992).

In this paper, we adopt a slightly different approach by which we avoid the introduction and training of a neural network copy of the plant equations (Piche & Widrow, 1991). The basic idea is that instead of building a neural network copy or emulator $\hat{P}$ of P to back propagate error gradients through it, it is possible to directly back propagate the error gradients through the plant equations P.

Let us consider a two-layer neural network that emulates a set of continuous nonlinear functions P. The inputs to the neural network are denoted by x_i^0, and the outputs by x_k^2. Applying eqn (11) recursively for $l = 0$, 1, and 2, and for all i, k, we get:

[4] It can be shown that any continuous nonlinear function can be approximated to an arbitrary degree of precision by a two-layer neural network (Cybenko, 1988; Hornik, Stinchcombe, & White, 1989; Irie & Miyake, 1988)

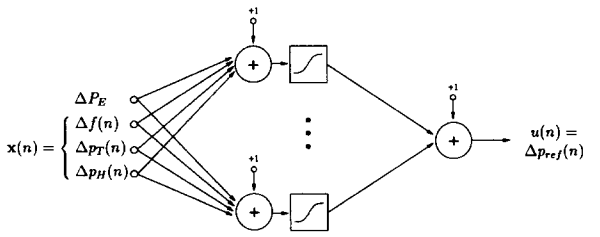

FIGURE 11. Neural network controller for a single-area system: internal structure.

$$x_k^2 = s_k^2\left[\sum_j w_{j,k}^1 \cdot s_j^1\left(\sum_i w_{i,j}^0 \cdot x_i^0\right)\right]. \tag{15}$$

Let us now evaluate the derivative of the kth output with respect to the ith input:

$$\frac{dx_k^2}{dx_i^0} = s_k'^2 \cdot \sum_j w_{j,k}^1 \cdot s_j'^1 \cdot w_{i,j}^0. \tag{16}$$

Let us also evaluate the error gradients according to eqn (14):

$$\delta_k^2 = -2e_k \cdot s_k'^2 \tag{17}$$

$$\delta_j^1 = s_j'^1 \cdot \sum_k \delta_k^2 \cdot w_{j,k}^1 \tag{18}$$

$$\delta_i^0 = \sum_j \delta_j^1 \cdot w_{i,j}^0 \tag{19}$$

Substituting eqn (17) in eqn (18) and eqn (18) in eqn (19), and comparing with eqn (16), we get:

$$\delta_i^0 = -2e_k \cdot \frac{dx_k^2}{dx_i^0}. \tag{20}$$

Equation (20) implies that the kth term of the error gradient vector δ_i^0 obtained by back propagating δ_k^2 through the neural network is proportional to the derivative of the kth output of the network with respect to its ith input. The matrix containing the derivatives of the outputs of P with respect to its inputs is called the *Jacobian matrix*

Building a neural network emulator of the plant and back propagating error gradients through it is nothing other than approximating the true Jacobian matrix of the plant using a neural network technique. Whenever the equations of the plant are known beforehand, they can be used to compute, analytically or numerically, the elements of the Jacobian matrix. The error gradient at the input of the plant is then obtained by multiplying the output error gradient by the Jacobian matrix.

This approach avoids the introduction and training of a neural network emulator, which brings a substantial saving in development time. In addition, the true derivatives being more precise than the one obtained approximately with a neural network emulator, the controller training is faster and more precise. One disadvantage of this method, however, is that a neural network controller included in such a structure cannot track changes in the plant (such as parameter variations due to wearing of certain parts, etc.). If such considerations are an issue, the neural network emulator method is a better choice. Again, the Jacobian approach is only applicable if an analytical description of the plant is known beforehand. Otherwise, a plant neural network emulator can be used to identify the plant and to back propagate the error gradients.

3.2. Neural Network Control of a Two-Area System

The neural network control scheme for a two-area system is basically the same as for a one-area system. Referring back to Section 2.2, the discrete state vector of the two-area system was chosen equal to $\mathbf{x}(n) = [\Delta f_1(n)\ \Delta f_2(n)\ \Delta p_{1,2}(n)]^T$. After a step-load perturbation has occurred in one area (or simultaneously in both areas), and after transients have died out, the state of the plant controlled by a neural network is expected to converge to the same steady-state value as it did with a conventional controller. We have seen in Section 2.2 that this steady-state vector was equal to $\mathbf{x}_{\text{steady-state}} = [0\ 0\ 0]^T$ [Eqn (10)]. This will thus be the desired response for the two-area system, once unravelled in time. Generalizing upon the control process for the single-area system, the inputs of the two-area neural network controller are: the state vector $\mathbf{x}(n)$ and the magnitude of the step-load perturbations, $\Delta P_{E,1}$ and $\Delta P_{E,2}$. The outputs are the reference power in area 1, $\Delta p_{\text{ref},1}(n)$, and the reference power in area 2, $\Delta p_{\text{ref},2}(n)$.

The two-area controller-plant structure is unravelled in time and trained the same way as the one-area system, using back propagation-through-time.

4. SIMULATIONS

Computer simulations have been conducted to illustrate the behavior of single-area and two-area systems subject to step load perturbations. Both neural networks have been adapted using back propagation-through-time. No

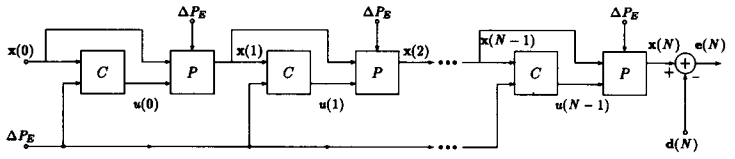

FIGURE 12. Neural network controller and plant model of a single-area system: block diagram unfolded in time.

plant emulator was used; instead, the error gradients were directly back propagated through the plant equations.

As explained in Section 2, the regulation parameters R of single-area and two-area systems are nonlinear functions of the power Δp_T produced in the areas (Cohn, 1986). The effect of this nonlinearity on the system is generally represented in the following way. In steady-state, a simple relationship can be written between the frequency Δf_0 and the turbine output power $\Delta p_{T,0}$ (see Figure 2): $\Delta p_{T,0} = \Delta p_{\text{ref},0} - (1/R)\,\Delta f_0$, or:

$$\Delta f_0 = R(\Delta p_{\text{ref},0} - \Delta p_{T,0}). \qquad (21)$$

The subscript 0 denotes steady-state values. If R is a constant, Δf_0 vs. $\Delta p_{T,0}$ represents a family of straight lines with parameter $\Delta p_{\text{ref},0}$. If instead the regulation R is a function of the turbine power, eqn (21) represents a family of nonlinear curves. Nonlinearities encountered in practical systems produce $\Delta f_0(\Delta p_{T,0})$ curves such as represented in Figure 13 (solid lines). For reference, the corresponding curves for R = constant = 2.4 pu Hz/MW are also shown (dashed lines).

4.1. Single-Area System Simulations

Such a nonlinearity was introduced in the discrete plant equations [eqns (1)–(3)] of a single-area system. A feedforward neural network containing one hidden layer with 20 hidden units was adapted to control this nonlinear plant. The neural network was trained to compensate for any perturbation ranging from minus to plus 100%.

Figure 14 shows the frequency fluctuations at the output of the power unit when a 10% step-load increase occurs on the bus. Two control schemes are compared: (a) the plant is controlled by a conventional integral controller with critical gain, and (b) the plant is controlled by a neural network controller. The neural network clearly achieves faster transient recovery while

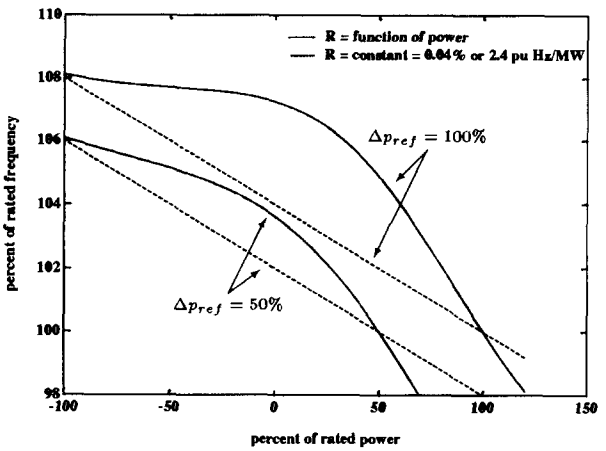

FIGURE 13. Static frequency-power response (linear case: R = 0.04% or 2.4 pu Hz/MW).

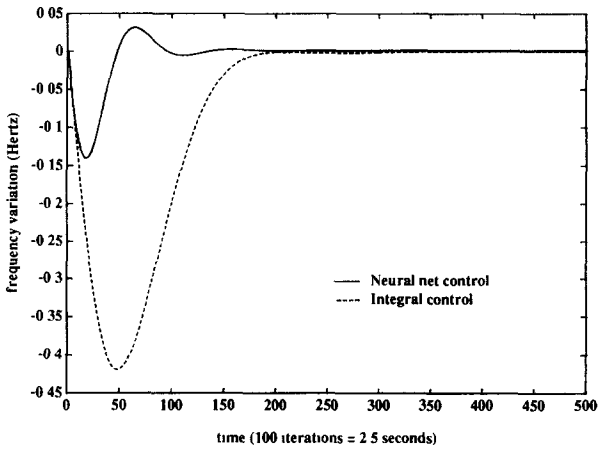

FIGURE 14. Step response of a single-area system subject to a 10% load increase.

still limiting the overshoot. This result is not unexpected. The neural network controller makes use of the estimate of the load-perturbation that is provided to it as an input signal to immediately counteract the effect of the perturbation hitting the system. The integral controller, on the other hand, proceeds by building up the frequency error and using it to drive the plant. Therefore, it has to wait until the error has built up before significantly effecting the system response.

In a sense, the comparison between the neural network and the PI controller may seem unfair because the neural network is provided with a piece of information that is not available to the other controller. On the other hand, the PI controller is the most widely used in practice and therefore constitutes the standard comparison tool for new control methods. The point that the authors would like to illustrate is that a judicious use of some additional information can produce a tremendous improvement in the system dynamic behavior.

4.2. Two-Area System Simulations

A one-hidden-layer neural network controller with 20 hidden units was then trained to control a nonlinear two-area system. The same nonlinearity as in the single-area system was used for both areas of this system.

Figure 15 compares the dynamic response of a two-area system subject to a 1% step-load increase when controlled (a) by a neural network (solid lines), or (b) by two conventional integral controllers with critical gains (dotted lines). Figure 15A shows the frequency variations in both areas; Figure 15B shows tie-line power variations.

As explained previously, the neural network controller takes as inputs the estimates of the load-perturbations in both areas. These estimates are evaluated by a special device that needs a short, but non-zero, lapse of time to output accurate estimates of the perturbations. For this reason, a 500-ms delay was allowed before

the neural net started controlling the plant. During these first 500 ms, the control signals $\Delta p_{ref,1}$ and $\Delta p_{ref,2}$ were set to zero. This explains that the solid and dotted lines in Figure 15 exactly coincide for the first five iterations (0.5 s) of the simulation. The use of a neural network controller has greatly reduced the frequency transient recovery period. The neural network achieves zero steady-state error and the uncontrolled system has a large steady-state error.

Figure 16 shows the frequency recovery of a two-area system subject to a 1% step-load increase in area 1 and, at the same time, a 2% step-load decrease in area 2. Dynamic responses obtained with a neural network and with integral controllers are compared.

5. CONCLUSION

Layered neural networks have been successfully applied to control the turbine reference power of a computer-simulated generator unit. The same principle has then been applied to a simulated two-area system. Both neural networks have been adapted using back propagation-through-time. In this paper, the frequency variations in both areas of the two-area system were inputted into the neural network controller. The neural network thus implements a global control scheme. In the past, nonneural network local controllers have been developed (Aly & Abdel-Magid, 1981; Abdel-Magid et al., 1984). A subject of future research consists in building independent neural network controllers for the different areas of a multiarea system. Each neural network controller receives only local information about the system (frequency in that specific area and powers in the tie-lines connected to that area). Such an architecture decentralizes the control of the overall system and reduces the amount of information to be exchanged between different nodes of the power grid. Performance with local control only will be compared with that obtained with global control.

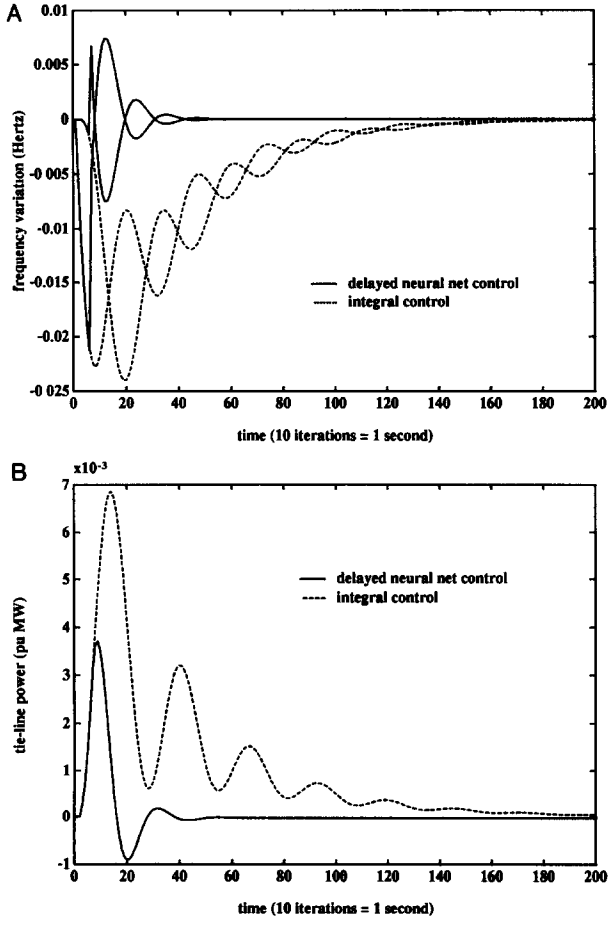

FIGURE 15. Dynamic response of a two-area system subject to a 1% step-load increase in area 2. (A) Frequency transients in both areas. (B) Tie-line power variations.

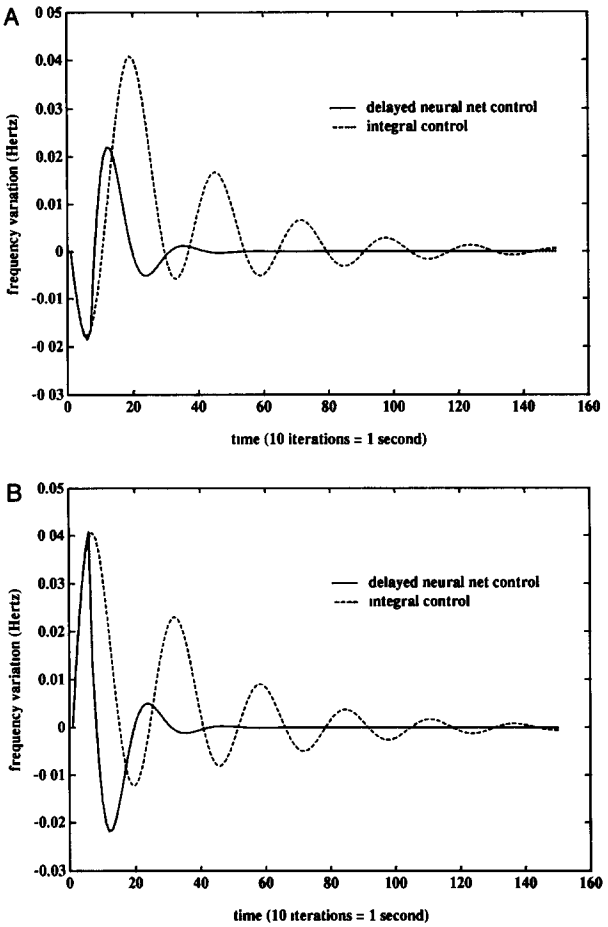

FIGURE 16. Dynamic response of a two-area system subject to a 1% step-load increase in area 1 and a 2% step-load decrease in area 2. (A) Frequency variations in area 1. (B) Frequency variations in area 2.

REFERENCES

Abdel-Magid, Y L., Aly, G , & Cherid, A. A. (1984) Robust decentralized load frequency regulator for interconnected power systems. *Eurocon* 84, (pp 69–73). 6th conference on electrotechnics-computers in communication and control. Brighton, UK

Aly, G , & Abdel-Magid, Y. L. (1981). Two level load-frequency

control of interconnected power systems. *IECI Proceedings* (pp 213–218). San Francisco, CA.

Anderson, P M, & Fouad, A A. (1977). *Power system control and stability* Ames: Iowa State University Press

Cohn, N (1986) *Control of generation and power flow on interconnected systems* New York: John Wiley.

Cybenko, G. (1988). Continuous valued neural networks with two hidden layers are sufficient. Department of Computer Science, Tufts University

Debs, A. S. (1988) *Modern power systems control and operation* Norwell, MA: Kluwer–Nijhoff Publishers.

Elgerd, O E. (1982). *Electric energy systems theory* (2nd ed) New York. McGraw–Hill, Inc

Hornik, K., Stinchcombe, M, & White, H (1989) Multilayer feedforward networks are universal approximators *Neural Networks*, **2**, 359–366.

Irie, B, & Miyake, S (1988). Capabilities of three-layered perceptrons. (pp 641–647) *Proceedings of the IEEE Second International Conference on Neural Networks* (vol I, pp 641–647). San Diego, CA

Nguyen, D, & Widrow, B (1989) The truck backer-upper: An example of self-learning in neural networks. *Proceedings of the International Joint Conference on Neural Networks* (vol II, pp 357–363) Washington, DC

Nguyen, D, & Widrow, B. (1990) Neural networks for self-learning control systems *IEEE Control Systems Magazine*, pp 18–23

Piche, S W., & Widrow, B (1991) *First-order gradient descent training of adaptive discrete-time dynamic networks* (Tech Rep RL-TR-91-62) Rome, NY Griffiss Air Force Base, Rome Laboratory, Air Force Systems Command.

Rumelhart, D. E, & McClelland, J L. (1986) *Parallel distributed processing* Cambridge, MA The MIT Press.

Werbos, P J. (1974). *Beyond regression New tools for prediction and analysis in the behavioral sciences* PhD thesis, Harvard University, Cambridge, MA

Werbos, P J. (1990) Backpropagation through time: What it does and how to do it [special issue on neural networks]. *Proceedings of IEEE*, **2**, 1550–1560

Widrow, B, & Lehr, M. A. (1990) 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation *Proceedings of IEEE*, **78**(9), 1415–1442

Wood, A J., & Wollenberg, B F. (1984). *Power generation, operation and control* New York John Wiley & Sons

Wu, Q. H., Hogg, B W, & Irwin, G. W (1992) A neural network regulator for turbogenerators. *IEEE Transactions on Neural Networks*, **3**(1), 95–100

NOMENCLATURE

1. Power Systems Quantities

1.1. *Single-Area System*

$\Delta f(t)$	generator frequency variation
$\Delta F(s)$	Laplace transform of $\Delta f(t)$
R	regulation parameter
$1/R$	speed regulator gain
K_I	integral controller gain
p_{ref}	turbine reference power in steady-state
Δp_{ref}	variation in reference power
$\Delta P_{\text{ref}}(s)$	Laplace transform of $\Delta p_{\text{ref}}(t)$
$\Delta p_H(t)$	fluctuation in hydraulic amplifier output power
$\Delta P_H(s)$	Laplace transform of $\Delta p_H(t)$
$\Delta p_T(t)$	fluctuation in turbine output power
$\Delta P_T(s)$	Laplace transform of $\Delta p_T(t)$
$\Delta p_E(t)$	electrical perturbation (i.e., load variation)
$\Delta P_E(s)$	Laplace transform of $\Delta p_E(t)$
$\Delta \widehat{p_E}(t)$	electrical perturbation estimate
K_H, K_T, K_P	transfer function gains for the hydraulic amplifier, the turbine, and the generator
T_H, T_T, T_P	transfer function time constants for the hydraulic amplifier, the turbine, and the generator
T_s	sampling period in discretized dynamic equations
n	discrete time index
$\mathbf{x}(nT_s)$	discrete state vector of the dynamic system

1.2. *Two-Area System* (in addition to above symbols with additional underscore 1 or 2 to specify the area concerned)

$\Delta p_{1,2}(t)$	fluctuation in tie-line power transmitted from area 1 to area 2
$\Delta P_{1,2}(s)$	Laplace transform of $\Delta p_{1,2}(t)$
$\Delta p_{2,1}(t)$	fluctuation in tie-line power transmitted from area 2 to area 1
$\Delta P_{2,1}(s)$	Laplace transform of $\Delta p_{2,1}(t)$
$\Delta\theta_1(t)$, $\Delta\theta_2(t)$	fluctuation in shaft angle in area 1 or 2
$\Delta\Theta_1(s)$, $\Delta\Theta_2(s)$	Laplace transform of $\Delta\theta_1(t)$, $\Delta\theta_2(t)$
T^0	tie-line synchronizing coefficient
B_1, B_2	proportionality coefficients

2. Neural Networks and Back Propagation Quantities

$w_{i,j}^l$	weight connecting neuron i in layer l with neuron j in next layer
x_i^l	ith input to neuron in layer l
$s_j^l(\cdot)$	sigmoid function in neuron j of layer l
$s_j'^l(\cdot)$	derivative of $s_j^l(\cdot)$ with respect to its argument
δ_j^l	error gradient produced in neuron j of layer l
$H(n)$	discrete step function of amplitude 1.0
$u(n)$	neural controller output
$\mathbf{e}$	neural network error vector
$\mathbf{d}$	desired output vector