Anticipatory fuzzy control of power systems

P.K. Dash
A.C. Liew

Indexing terms: Fuzzy logic, Kalman filters, Neural networks, Power systems, Stability, Transient disturbances

Abstract: The paper presents an anticipatory fuzzy control to improve the stability of electric power systems. This differs from the traditional fuzzy control in that, once the fuzzy-control rules have been used to generate a control value, a predictive routine built into the controller is called for anticipating its effect on the system output and hence updating the rule base or input–output membership functions in the event of unsatisfactory performance. The effectiveness of the anticipatory and traditional PI fuzzy controllers is demonstrated by simulation studies on a single-machine infinite-bus and multimachine power system subjected to a variety of transient disturbances for different operating conditions. The anticipatory fuzzy control, however, requires a neural-network prediction routine using modified-Kalman-filter-based fast-learning algorithm.

1 Introduction

Many modern power systems have long transmission distances and remote sources of generation. Such systems have high series impedance which reduces system stability. Conventional stabilisers of fixed structure and constant parameters are tuned for one operating point and can give optimal performance for that condition. As the characteristics of power system elements are nonlinear, conventional stabilisers are not capable of providing optimal performance for all operating conditions.

With the widespread use of static excitation systems, the transient-stability limit of a synchronous machine has been significantly improved by the features of fast response and high ceiling voltage inherent in a static-excitation system. However, during small disturbances, the damping of synchronous-generator oscillations deteriorates and this necessitates the use of an auxiliary controller. Amongst the various control schemes proposed earlier, a supplementary excitation controller which can generate a damping signal in the excitation system has attracted widespread interest. In the past decade, a considerable amount of research has been undertaken to develop a self-tuning power-system stabiliser [1, 2] for generating the desired supplementary stabilising signal.

Recently, alternative control schemes based on rule-based stabiliser and fuzzy logic decision system have been proposed. Of these schemes, fuzzy control appears to be the most suitable owing to its low computational burden and ease of implementation using a microcomputer [3–5]. The fuzzy-logic-based controller overcomes system ambiguities and parameter variations by modelling the control objective in terms of a human operator's response to various system scenarios, thereby eliminating the need for an explicit mathematical model of the system dynamics. The fuzzy controller for synchronous-generator stabilisation relates significant and observable variables such as generator speed and its rate to an auxiliary control signal for the power-system stabiliser using fuzzy-membership functions. These variables evaluate control rules using the compositional rule of inference.

In this paper, a new control strategy, called anticipatory fuzzy control, is proposed for supplementary stabilisation of synchronous machines in a power system. This control differs from the traditional fuzzy control in that, once fuzzy-control rules have been used to generate a control value, a predictive routine built into the controller is called to anticipate the effect of the proposed control on the system output. If using the current control value will result in system behaviour which is in some way unacceptable, additional rules are called. This method may be used to nest as many sets of rules as the designer desires. The predictive routine uses a neural network based on a modified backpropagation learning technique [7]. The anticipating fuzzy controller based on a neural-network system-identification routine [8, 9] produces significant damping of the electromechanical oscillations of a single-machine infinite-bus and multimachine power system for a variety of transient conditions. However, the implementation of this type of controller requires one-step-ahead prediction of machine speed resulting in increased computational overhead comparison with a conventional fuzzy controller.

2 Fuzzy control of synchronous-generator excitation

This Section details the design of the rule-based fuzzy-logic controller used to achieve the desired transient performance of a synchronous generator connected to a large power system (Fig. 1*a*). The idea of a fuzzy set introduced by Zadeh [6] allows imprecise and quantitative information to be expressed in an exact way and, as the name implies, is the generalisation of the characteristic function which can take values between 0 and 1. In the fuzzy-control algorithm, the state variables are the fuzzy

---

Paper 1585C (P11), first received 7th March and in revised form 23rd August 1994

P.K. Dash is with the Centre of Applied Artificial Intelligence, Regional Engineering College, Rourkela, 769008, India

A.C. Liew is with the Department of Electrical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 0511, Republic of Singapore

sets associated with $\Delta\omega$ and $\Delta\dot{\omega}$, where $\Delta\omega$ is the generator-speed-deviation signal used as a supplementary stabilising signal for the PSS (shown in Fig. 2) and $\Delta\dot{\omega}$ is the acceleration (derivative of $\Delta\omega$) signal.

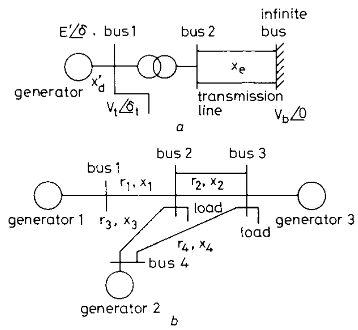

Fig. 1 *Single- and multi-machine configurations*
a Single-machine-infinity-bus power system
b Multimachine-power-system configuration

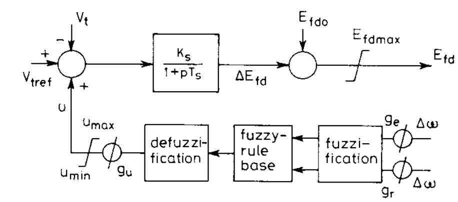

Fig. 2 *Supplementary fuzzy controller for PSS*

The inputs to the fuzzy controller are

$$\left.\begin{aligned}\Delta\omega(nT) &= \omega(nT) - \omega_0 \\ \Delta\dot{\omega}(nT) &= \{\Delta\omega(nT) - \Delta\omega(nT - T)\}/T\end{aligned}\right\} \tag{1}$$

where T is the sampling period and n is a positive integer.

The fuzzified input and output from the fuzzy controller are

$$\begin{aligned}e(nT) &= F[g_e\,\Delta\omega(nT)] \\ ce(nT) &= F[g_r\,\Delta\dot{\omega}(nT)] \\ du(nT) &= DF[dU(nT)] \\ u(nT) &= u(nT - T) + g_u\,\Delta u(nT)\end{aligned} \tag{2}$$

where $F[\;]$, and $DF[\;]$ stand for fuzzification and defuzzification, respectively, and $\Delta u(nT)$ denotes the incremental output after defuzzification at sampling time nT, and g_e, g_r and g_u are gains for error (e), its rate (ce) and the control output (dU), respectively.

2.1 Fuzzification algorithm and fuzzy-control rules

A simple fuzzification algorithm for scaled errors and rate is shown in Fig. 3, where L denotes either the maximum error $\Delta\omega$ or maximum rate $\Delta\dot{\omega}$ multiplied by the gains g_e and g_r, respectively. The fuzzy-set error e has two members, i.e. error positive e_p and error negative e_n, and fuzzy-set rate ce has two members, i.e. rate positive r_p and rate negative r_n. The output fuzzy-set control dU has three members, i.e. output positive o_p, output negative o_n, and zero output o_z.

The following fuzzy-control rules are used in this program

Rule 1: IF e is e_p AND ce is r_n THEN dU is o_z

Rule 2: IF e is e_p AND ce is r_p THEN dU is o_p

Rule 3: IF e is e_n AND ce is r_n THEN dU is o_n

Rule 4: IF e is e_n AND ce is r_p THEN dU is o_z

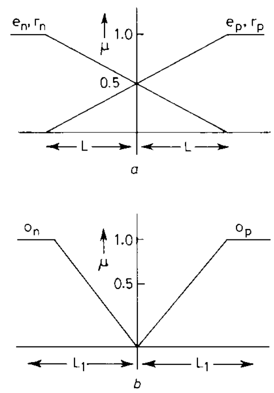

Fig. 3 *Fuzzification*
a Fuzzification of error and rate
b Fuzzification of control output dU

Thus the rule based considered above belongs to the simplest fuzzy controller. Extra rules can be added by including more fuzzy classifiers such as e_{sp} (error small positive), e_{sn} (error small negative), e_{lp} (error large positive) and e_{ln} (error large negative). Similar classifiers for rate and control output are used. The total number of rules in the rule base will thus be 16.

The membership functions for the error e and its rate ce are obtained from Fig. 3 as

$$\left.\begin{aligned}\mu_{ep}(e) &= \{L + g_e e\}/2L \\ \mu_{en}(e) &= \{L - g_e e\}/2L \\ \mu_{rp}(e) &= \{L + g_r ce\}/2L \\ \mu_{rn}(e) &= \{L - g_r ce\}/2L\end{aligned}\right\} \tag{3}$$

The membership functions for the control output are given by

$$\left.\begin{aligned}\mu_{op}(dU) &= dU/L_1 \\ \mu_{on}(dU) &= -dU/L_1 \\ \mu_{oz}(dU) &= 0\end{aligned}\right\} \tag{4}$$

To enable the fuzzy controller to operate for any given input, use is made of the compositional rule of interference as $dU = (e \times ce) \cdot R$.

Normally $\cdot$ denotes the max–min product. In our case, Zadeh AND and OR rules are used for evaluating the control. Zadeh AND or OR rules are given by

AND rule:

$$\mu_{a \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \tag{5}$$

OR rule:

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \tag{6}$$

A and B are the linguistic sets for error $\Delta\omega$ and the rate $\Delta\dot{\omega}$, respectively. The inference engine of the fuzzy-logic-based controller matches the preconditions of rules in the fuzzy-rule base with input-state linguistic terms and performance implications. For example, for a given error and its rate, the firing strengths α_1, α_2, α_3 and α_4 of rules 1–4, are obtained as

$$\begin{aligned} \alpha_1 &= \mu_{ep}(e) \wedge \mu_{rn}(ce) \\ \alpha_2 &= \mu_{ep}(e) \wedge \mu_{rp}(ce) \\ \alpha_3 &= \mu_{en}(e) \wedge \mu_{rn}(ce) \\ \alpha_4 &= \mu_{en}(e) \wedge \mu_{rp}(ce) \end{aligned} \tag{7}$$

where $\wedge$ stands for fuzzy AND operation.

Taking the min operator

$$\left.\begin{aligned} \alpha_1 &= \{L - g_r ce\}/2L \\ \alpha_2 &= \{L + g_r ce\}/2L \\ \alpha_3 &= \{L + g_e e\}/2L \\ \alpha_4 &= \{L - g_e e\}/2L \end{aligned}\right\} \tag{8}$$

the control output of any rule is calculated by matching strength of its precondition on its conclusion. Since rule 1 and rule 4 have the same output set, the Zadeh OR rule is used to evaluate the output decision α_0 as

$$\alpha_0 = \alpha_1 \vee \alpha_4 \tag{9}$$

whose $\vee$ stands for the max operation and α_0 is

$$\alpha_0 = \{L - g_r ce\}/2L \tag{10}$$

Using a nonlinear defuzzification, the control output dU is given by

$$dU = (\alpha_2\, dU_2 + \alpha_3\, dU_3)/(\alpha_2 + \alpha_3 + \alpha_0) \tag{11}$$

where dU_2 and dU_3 are the values of the control output for which membership values of $\mu_{op}(dU)$ and $\mu_{on}(dU)$ are each equal to 1. The nonlinear defuzzification algorithm, after some simplification, yields the control Δu as

$$\Delta u = L_1 g_u[g_e e + g_r ce]/(3L - g_e|e|) \tag{12}$$

for $g_r|ce| \leqslant g_e|e| \leqslant L$
and

$$\Delta u = L_1 gu[g_e e + g_r ce]/(3L - g_r|ce|) \tag{13}$$

The scaling factor g_u is chosen as -1 and the values of g_e and g_r are obtained by minimising a performance index of the form

$$J = \int_0^t (t\ \Delta\omega)^2\, dt \tag{14}$$

If, on the other hand, the weighted-average method is used for defuzzification, the controller output will be given by

$$\Delta u = 2L_1\{(L + g_r ce)^2 - (L - g_e e)^2\}/(2L - g_e|e|) \tag{15}$$

for $g_r|ce| \leqslant g_e|ce| \leqslant L$ and

$$\Delta u = 2L_1\{(L + g_r ce)^2 - (L - g_e e)^2\}/(2L - g_r|e|) \tag{16}$$

for $g_e|e| \leqslant g_r|ce| \leqslant L$

A new control strategy, called anticipatory fuzzy control, differs from traditional fuzzy control in that, once fuzzy rules have been used to generate a control value, a predictive routine built into the controller is called to anticipate the effect of the proposed control on the system output. If using the current control value will result in system behaviour which is some way unacceptable, additional rules are called. This method may be used to nest as many sets of rules as the desigure desires.

The advantages compared with standard fuzzy controllers are

(i) nesting rules allow the use of only as many rules as are necessary to achieve the desired system performance, saving computer time; and

(ii) by predicting system performance, controls which would result in unstable or unacceptable system performance can be discarded.

The simplest type of anticipatory control has a single additional rule of the form: 'If the current control u_c will cause the difference between the current and anticipated values of rate of change of speed deviation $\Delta\omega$ to be large, then the new control u is

$$u = u_c[1 - \beta\mu\{\hat{\Delta}\dot{\omega}(t + T) - \hat{\Delta}\dot{\omega}(t)\}] \tag{17}$$

and

$$\hat{\Delta}\dot{\omega} = \{\hat{\Delta}\omega(t + 1) - \hat{\Delta}\omega(t)\}/T$$

where $\mu(\hat{\Delta}\dot{\omega})$ is the membership function of the predicted acceleration and the value of β varies between 0 and 1 and can be determined heuristically'.

In this paper an optimum value of β is found by minimising quadratic-performance index of the form

$$J = \int_0^t (t\ \Delta\omega)^2\, dt \tag{18}$$

However, a fuzzy supervisor can be used to determine the value of β by computing a pseudodamping rate

$$d_r = r(t + T)/r(t)$$

where

$$r(t) = \Delta\dot{\omega}(t)^2 + \beta_1(\Delta\dot{\omega}(t)^2 \quad \beta_1 > 0 \tag{19}$$

The value of d_r will determine the value of β using the logic of approximate reasoning. If d_r is small, the system state will move towards the desired state.

3 Neural network for prediction

The implementation of an anticipatory fuzzy controller necessitates the prediction of the $\Delta\omega(nT + T)$ one step ahead of the input signal and thus a neural network [8] can be used for system identification and one-step-ahead prediction. For a power system, the input–output relationship can be written in the form

$$y(t + T) = f\{y(t), y(t - T), \ldots, y(t - nT), u(t), \ldots, u(t - nT)\} \tag{20}$$

where the function f represents a memoryless nonlinear function. As shown in Fig. 4*a*, the neural net can be substituted for the memoryless nonlinear function f to construct the neural-net model for the plant. As shown in the Figure the neural net monitors the current and past plant inputs and outputs and is adjusted, or trained, so that it can predict the next output of the plant. The block diagram for the anticipatory fuzzy control using a neural net for one-step-ahead prediction is shown in Fig. 4b.

From the Figure, it is seen that the output from the fuzzy controller $u(t)$ goes to the neural net along with past inputs and outputs to predict the one-step-ahead output. However, if the neural net-predicted one-step-ahead output deviates significantly from the actual system output, the weight adjustment of the previously learned neural-net model is carried out.

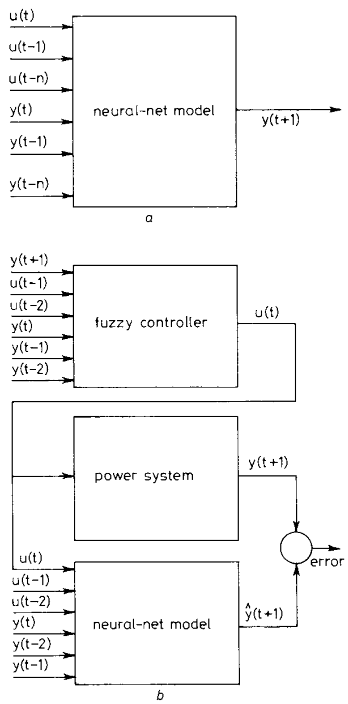

Fig. 4 *Neural-network models*
a Neural-network model of the power system
Input to net: measurements $u(t), \ldots, u(t-n), y(t), \ldots, y(t-n)$. output: prediction $y(t+1)$
b Neural-network prediction and fuzzy-control model of the power system

In this paper a modified backpropagation neural network with a fast learning algorithm [9] is used to predict the output one step ahead. The algorithm minimises the mean-squared error between the desired output and the actual output with respect to summation output (inputs to nonlinearities). This is in contrast to the standard backpropagation algorithm which minimises the mean-squared error with respect to weights. Error signals, generated by the backpropagation algorithm, are used to estimate values at summation outputs which will improve the total network error. These estimates, along with the input vectors to the respective nodes, are used to produce an updated set of weights through a system of linear equations at each node. These linear equations are solved using a Kalman filter at each layer (Figs. 5A–C) and training patterns are run through each layer until the convergence is reached. An autotuning procedure and Kalman-gain variation as a function of the output error E are used to achieve very fast convergence to the earlier algorithm. The details are given in Appendix 8.1.

The neural network used for the training comprises six input neurons, one hidden layer with 15 neurons and one output neuron. The inputs to the net are $\Delta\omega(t)$, $\Delta\omega(t-T)$, $\Delta\omega(t-2T)$, and $\Delta E_{fd}(t)$, $\Delta E_{fd}(t-T)$, $\Delta E_{fd}(t-2T)$ and ΔE_{fd} is the excitation voltage change to be applied to the generator. The training patterns consist of the data samples generated under normal and a variety of abnormal conditions to predict the one-step-

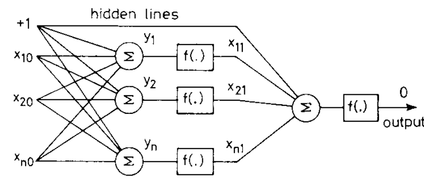

Fig. 5A *Feedforward neural network used for prediction*

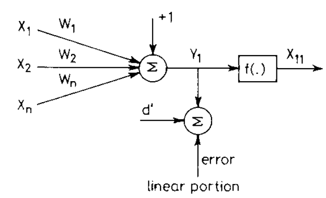

Fig. 5B *Linear and error portion of the hidden layer*

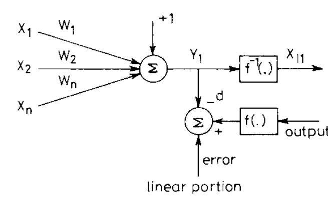

Fig. 5C *Linear and output portion of the output layer*

ahead output, i.e. $\Delta\omega(t+T)$. The parameters for the training algorithm are

Initial weights: -0.8 to 0.8
Number of iterations: 80
$\bar{\mu}$ = step size = 40
a = sigmoid slope = 0.28
c = Kalman initialisation constant = 1.0
b_0 = forgetting factor (starting value) = 1.0
b = forgetting factor = 0.98

As seen from the above parameters, the number of iterations required to achieve convergence is very small compared with the normal backpropagation (which takes nearly 10 000 iterations) and compared with the modified backpropagation proposed earlier and without autotuning features.

4 Power system studied

Case 1: Single-machine infinite-but system
The power system shown in Fig. 1*a* consists of a synchronous generator connected via high-voltage double-circuit transmission lines, represented by a lumped reactance, to a large power system, represented by an infinite bus.

The generator is equipped with a governor and a fast static-excitation system and the excitation control loop includes a conventional AVR and, in addition, an auxiliary fuzzy controller. The fuzzy controller is activated from the generator speed and acceleration signals. The synchronous generator is represented by a third-order model and its equations and parameters are given in Appendix 8.2. The following fuzzy-controller parameters are used for optimum results: $g_e = 2.0$, $g_r = 0.4$ and $\Delta\Omega(\max) = 5$ rad/s, $\Delta\dot{\omega}(\max) = 25$ rad/s. The tuning parameter for the anticipatory fuzzy control is initially fixed at 0.4. Its variation greatly affects the transient performance of the studied system. The following disturbances are considered in these simulation studies.

4.1 Step change in input power

The performance of the anticipatory fuzzy controller for a step change in input power ($P = 0.8$, $Q = 0.2$) is shown in Fig. 6. The input power is changed by 20% and kept at that value for a duration of 0.5 s. The anticipatory fuzzy controller β is fixed at 0.8. The choice of this parameter is very critical in providing optimal transient performance.

From Fig. 6 it is seen that the first swings of the rotor-angle displacement δ are considerably reduced compared with those of the conventional controller. Further, the damping provided by the anticipatory fuzzy controller for both speed and rotor-angle oscillations is found to be superior to that provided by the traditional fuzzy controller.

4.2 Small changes in the infinite-bus voltage

Performance results for a small change (2%) in the reference terminal-voltage magnitude are shown in Fig. 7 for both types of fuzzy controller. The small-oscillation performance of the anticipatory fuzzy controller is found to be better than that of the fuzzy PI controler. The initial P and Q loadings are fixed at $P = 0.8$, $Q = 0.2$. The first swings in both rotor angle and speed oscillations are significantly reduced.

4.3 Three-phase fault at generator bus

The performance of the anticipatory fuzzy controller for a 3-phase short circuit on the generator bus and cleared in 0.1 s is shown in Fig. 8. The Figure also presents the results for the fuzzy PI controller for comparison. The first swings of the rotor angle are considerably reduced using the new approach.

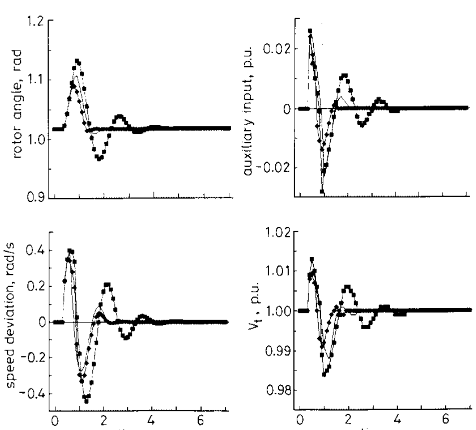

Fig. 6 *Response to 20% increase in input torque $P = 0.8$, $Q = 0.2$*
—■— conventional
—◆— anticipatory
—— fuzzy

The generation speed deviation and controller outputs show a similar trend, indicating the superiority of the new controller over the traditional fuzzy controller.

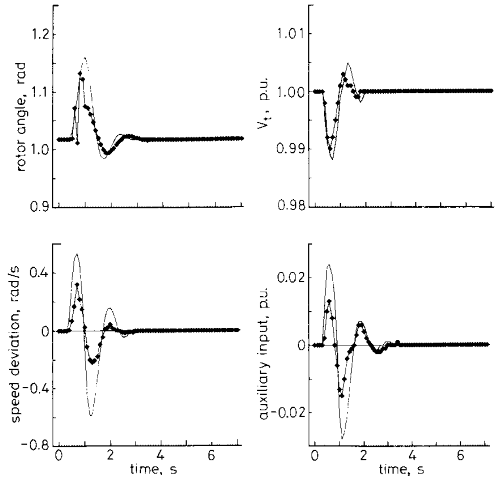

Fig. 7 *Response to 2% decrease in terminal voltage reference magnitude*
$P = 0.8$, $Q = 0.2$ —— fuzzy —◆— anticipatory

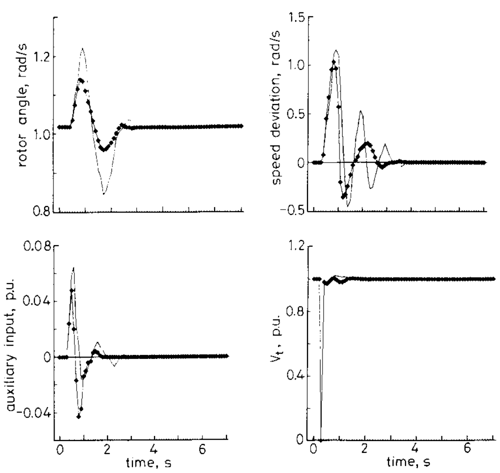

Fig. 8 *Response to a three-phase fault at the generator bus*
—— fuzzy
—◆— anticipatory

4.4 Effect of variation of β

The effect of variation of β on the transient performance of the generator connected to an infinite bus is shown in Fig. 9. The operating point of the generator is fixed at $P = 1.2$, $Q = 0.4$. A three-phase short circuit is created on the infinite bus and is cleared 0.1 s after its initiation. In addition, one of the transmission lines is removed immediately after the fault. From the transient-response curves it is seen that, for low or high values of β, the overshoot and settling time are both higher (Fig. 9*a*). An optimum value of β is found to be 0.8 for the case considered in this paper. The speed deviation and auxiliary-controller output for this fault are shown in Fig. 9*b* using conventional, fuzzy and anticipatory fuzzy ($\beta = 0.8$) control.

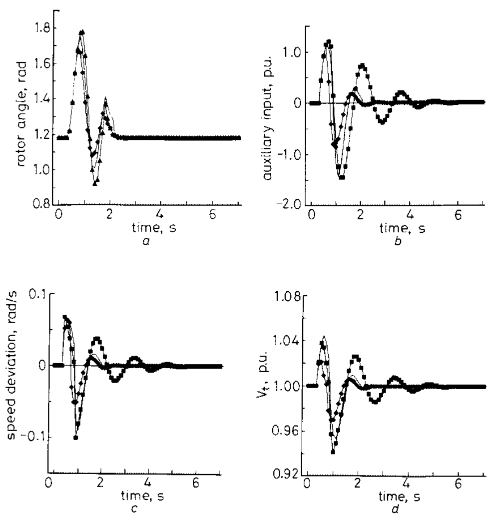

Fig. 9 *Effect of three-phase fault at infinite bus and removal of one line*
a Effect of variation of β on the dynamic performance
—— $\beta = 0.4$
—◆— $\beta = 0.8$
—▲— $\beta = 1.0$
b Effect on speed
—■— conventional
—▲— anticipatory
—— fuzzy PI
c Effect on auxiliary input
—■— conventional
—▲— anticipatory
—— fuzzy PI
d Effect on V_t
—■— conventional
—▲— anticipatory
—— fuzzy PI

Case 2: Multimachine power system
The multimachine system used in the studies is shown in Fig. 1*b*. This system consists of three generators with transmission lines and associated loads. Each system is equipped with a simple governor, an AVR and thyristor exciter for fast response. In addition, generators 1 and 2 are provided with auxiliary fuzzy controllers for the PSS loop, and these are activated by speed and acceleration error signals. Generator 3 is provided with conventional PSS.

The parameters of the generator governors and transmission lines are given in Appendix 8.3. The conventional stabiliser uses the same parameters as in the single-machine infinite-bus case.

4.5 Three-phase fault at busbar 1

Fig. 10 shows the system response to a three-phase short circuit to earth at busbar 2 which is cleared after 100 ms, and illustrates the response for fuzzy and neurofuzzy (anticipatory fuzzy) controllers. The swings of the rotor-angle displacement δ_{12} and δ_{13} are considerably reduced with the anticipatory fuzzy controller in comparison with the traditional fuzzy controller. The oscillations settle out in about 2 s. The oscillations in the rotor speeds $\Delta\omega_2$ and $\Delta\omega_3$ of generators 2 and 3, respectively, also show a similar trend and the damping is found to be significantly improved using the combined fuzzy and neural-network control.

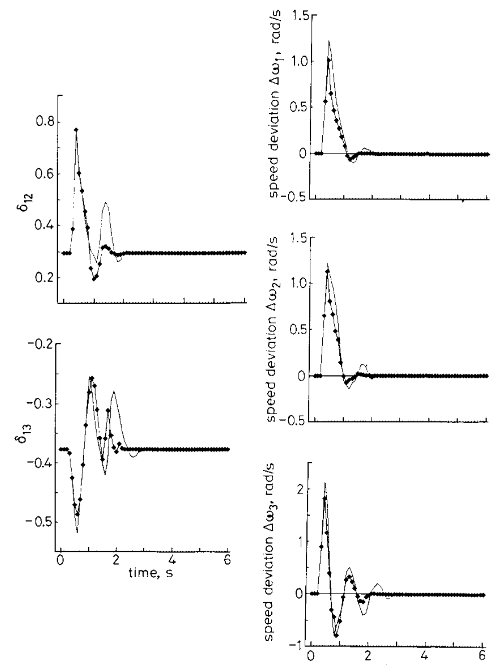

Fig. 10 *Response to a three-phase short circuit at bus 2, multimachine power system*
—— fuzzy
—◆— neuro-fuzzy anticipatory

5 Discussion

From the results presented in Figs. 6–10, it can be seen that the anticipatory fuzzy controller produces excellent damping for a variety of transient distrubances in single- and multimachine power systems and the constant β is an important parameter for optimising the transient performance of this controller. A performance index can be used to choose the optimum value of β for the best damping. The fuzzy PI controller also produces excellent damping but has slightly more initial overshoot and a greater settling time than the proposed fuzzy control. Further, the value of β can be chosen using a fuzzy supervisor.

6 Conclusions

The paper presents the design of an anticipatory fuzzy controller for the auxiliary control loop of the PSS of a single-machine infinite-bus and mulimachine power system. Both the fuzzy and anticipatory fuzzy controller produce excellent damping compared with the conventional controller for a variety of transient disturbances on the power system. The anticipatory fuzzy controller, however, requires a neural-network prediction routine for an accurate one-step-ahead prediction of the generator acceleration. The proposed anticipatory fuzzy stabiliser is very simple for practical implementation since the decentralised output-feedback control law developed in this paper requires only local measurement at each generating unit.

7 References

1 CHANG, S.J., CHOW, Y.S., MALIK, O.P., and HOPE, G.S.: 'An adaptive synchronous machine stabiliser', *IEEE Trans.*, 1986, **PWRS-1**, (3), pp. 101–109

2 GHANDAKLY, A.A., and FARHOUD, A.M.: 'A parametrically optimized self-tuning regulator for power system stabilizers', *IEEE Trans.*, 1992, **PWRS-7**, (3), pp. 1245–1251

3 HIYAMA, T., and LIM, C.M.: 'Application of fuzzy logic control scheme for stability enhancement of a power system'. IFAC symposium on Power systems and power plant control, Singapore, August 1989

4 HASSAN, M.A.M., MALIK, O.P., and HOPE, G.S.: 'A fuzzy logic based stabiliser for synchronous machines', *IEEE Trans.*, 1991, **EC-6**, (3), pp. 407–414

5 HSU, Y., and CHANG, C.: 'Design of fuzzy power system stabilisers for multimachine power systems', *IEE Proc. C*, 1990, **137**, (3), pp. 233–238

6 ZADEH, L.A.: 'Fuzzy sets', *Inf. & Control*, 1965, **8**, pp. 338–353

7 BUCKLEY, J.J., and YING, H.: 'Fuzzy controller theory limit theorems for linear fuzzy control', *Automatica*, 1988, **25**, pp. 472–496

8 PAO, Y.H., PHILLIPS, S.M., and SOBAJIC, D.J.: 'Neural-net computing and the intelligent control systems', *Int. J. Control*, 1992, **56**, (2), pp. 263–291

9 SCALERO, R.S., and TEPEDELENLIOGLUE, N.: 'A fast new algorithm for training feedforward neural networks', *IEEE Trans.*, 1992, **ASSP-40**, (1), pp. 202–210

8 Appendixes

8.1 Kalman-filter-based learning algorithm

Referring to Fig. 5, the nonlinear problem of system identification is reduced to a linear one with known y'_p and x'_p (outputs and inputs). Here, the output and inputs to the network are known during training and so the desired output $[d_p = f^{-1}(o_p)]$ is known but all other inputs are unknown and need to be estimated. If the estimates are designated by the usual known symbols with a prime, so that the estimate of x_p is $x'_p = f(d'_p)$, then d'_p can be calculated by using

$$d'_p = y_p + u_p e_p \tag{21}$$

where μ_p is the learning rate and e_p is the error calculated in the conventional backpropagation algorithm.

The total mean-squared error E is given by

$$E = \sum_{p=1}^{M} (d_p - y_p)^2 \tag{22}$$

where M is the number of training patterns, and y_p and d_p are the actual and desired summation outputs for the pth training pattern. The weight vector $\boldsymbol{W}$ [9] can be calculated by solving the equation

$$\boldsymbol{W} = \boldsymbol{R}^{-1}\boldsymbol{C} \tag{23}$$

where $\boldsymbol{R}$ is the autocorrelation matrix

$$= \sum_{p=1}^{M} x_p x_p^T \tag{24}$$

$\boldsymbol{C}$ is the crosscorrelation matrix

$$= \sum_{p=1}^{M} d_p x_p \tag{25}$$

where M is the number of patterns.

The weights are updated by solving these sets of linear equations at each node. This can be achieved by each layer with a variable forgetting factor.

Further, to increase the rate of convergence and reduce the number of iterations, the slope a of the sigmoidal nonlinearity, the learning rate $\bar{\mu}$ and the Kalman gain are updated at each iteration. This reduces the number of iterations drastically from 300 or so (proposed in the earlier algorithm) to 100 to achieve convergence.

The modified algorithm (compared with that presented in Reference 9) is implemented in the following steps:

Step 1: Randomly select training pattern.

Step 2: For each layer j, $j = 1, \ldots, L$, calculate, for every node k, the summation output

$$y_{jk} = \sum_{i=0}^{N} (x_{j-1,i} w_{jki}) \tag{26}$$

and function output

$$x_{jk} = f(y_{jk}) = [\{1 - \exp(-ay_{jk})\}/\{1 + \exp(-ay_{jk})\}] \tag{27}$$

where N is the number of inputs (including the offset) to a node, and a is the sigmoid slope.

Step 3: For each layer j from 1 to L, calculate the Kalman gain

$$k_j = R_j^{-1} x_{j-1}/(b_j + x_{j-1}^T R_j^{-1} x_{j-1}) + \gamma_j \, \partial E^j/\partial k_j \tag{28}$$

The gradient $\partial E^j/\partial k_j$ is calculated as

$$\frac{\partial E^j}{\partial k_j} = \{E^j(t) - E^j(t-2)\}/\{k_j(t-1) - k_j(t-2)\} \tag{29}$$

The forgetting factor is updated as

$$b_j = b_0 b_j + (1 - b_0) \tag{30}$$

and, $\boldsymbol{R}_j^{-1} = (\boldsymbol{R}_j^{-1} - k_j x_{j-1}^T \boldsymbol{R}_j^{-1}) b_j^{-1}$ where $\boldsymbol{R}_j^{-1}$ is the inverse autocorrelation matrix.

Step 4: The derivatives of $f(y_{jk})$ are calculated as

$$f'(y_{jk}) = 2a\{\exp(-ay_{jk})\}/\{1 + \exp(-ay_j)\}^2 \tag{31}$$

where the prime denotes the derivative.

Further, the sigmoidal slope is changed at every iteration as

$$a(t) = a(t-1) - \gamma_a \frac{\partial E}{\partial a} = a(t-1) + \gamma_a\{E(t-1) - E(t-2)\}/\{a(t-1) - a(t-2)\} \tag{32}$$

and γ_a is chosen to be positive for the choice of the sigmoidal parameter in the proper direction of convergence.

The output error signal at $j = 1$ is

$$e_{LK} = f'(y_{LK})(O_K - x_{LK}) \tag{33}$$

and the hidden-layer error signal is ($j = 1 - 1, \ldots, 1$)

$$e_{jk} = f'(y_{jk}) \sum (e_{j+1,i} w_{j+1,i,k}) \tag{34}$$

Step 5: The desired summation output at the output layer is obtained as

$$d_k = (1/a) \ln \{(1 + o_k)/(1 - o_k)\} \tag{35}$$

for every kth node at the output layer.

Step 6: The output-layer weight $j = 1$ is updated as

$$w_{LK} = w_{LK} + k_L(d_k - y_{Lk}) \tag{36}$$

and the hidden-layer weights are updated as

$$w_{jk} = w_{jk} + k_j e_{jk} \mu_j \quad (37)$$

for $j = 1, \ldots, 1$.

Initialisation: Node offsets are fixed at $+1$. Network weights are initialised to random values and initialise the $\boldsymbol{R}^{-1}$ matrix. In the above algorithm, j stands for layer, k for neuron and i for inputs to the kth neuron.

8.2 Synchronous-machine equations

$$\dot{\delta} = \Delta\omega$$

$$\Delta\dot{\omega} = (T_m - E_q i_q - D\,\Delta\omega)/M$$

$$\dot{E}'_q = (E_{fd} - (x_d - x'_d)i_d - E'_q)/\tau'_{do}$$

$$E_q = E'_q + (x_q - x'_d)i_d$$

$$v_d = x_q i_q$$

$$v_q = E'_q - x'_d i_d$$

$$E_{fd} = E_{fdo} + \Delta E_{fd}$$

$$v_t = \sqrt{(v_d^2 + v_q^2)}$$

i_d, i_q = d-axis and q-axis currents

v_d, v_q = d-axis and q-axis voltages

AVR and exciter:

$$\Delta\dot{E}_{fd} = \{k_e(v_{ref} - v_t) - \Delta E_{fd}\}/T_e + k_e u/\tau_e$$

Governor:

$$\dot{u}_g = \{k_g\,\Delta\omega - u_g\}/T_g$$

and

$$\dot{T}_m = (F_{np} u_g - T_m)/\tau_c$$

The conventional stabiliser used in this simulation is of the form

$$u = -k_s \frac{sT_q}{1 + sT_q}\left(\frac{1 + sT_1}{1 + sT_2}\right)\Delta\omega$$

8.3 System parameters

8.3.1 Single-machine infinite-bus system

$x_d = 2.0 \quad x_q = 2.0 \quad M = 0.032 \quad x'_d = 0.271$

$x_e = 0.2 \quad \tau'_{do} = 4.955 \quad v_t = 1.0$

AVR, PSS and governor parameters:

$T_e = 0.1$ s $\quad k_e = 50.0 \quad k_g = 2.5 \quad T_g = 0.1$ s

$F_{np} = 1.0 \quad \tau_c = 0.1$ s $\quad k_s = 0.1 \quad T_1 = 0.3$ s

$T_2 = 0.04$ s $\quad T_q = 2.5$ s

Limitation on control signals:

$E_{fd} \in [-6.0, 6.0]$

$u_g \in [0, 1.15]$

$u \in [-0.1, 0.1]$

8.3.2 Data for three-machine power system

Generators 1 and 2:

$x_d = 2.0 \quad x_q = 2.0 \quad M = 0.032 \quad x'_d = 0.271$

$\tau'_{do} = 4.955$ s

Generator 3:

$x_d = 1.537 \quad x_q = 1.476 \quad M = 0.0276 \quad x'_d = 0.249$

$\tau'_{do} = 4.629$ s

Line parameters:

$r_{12} = 0.2 \quad x_{12} = 0.2 \quad r_{13} = 0.015 \quad x_{13} = 0.15$

$r_{24} = 0.02 \quad x_{24} = 0.2 \quad r_{34} = 0.02 \quad x_{34} = 0.2$

AVR and governor parameters and limitation on control signals are chosen to be same as for the single-machine infinite-bus power system:

Loads	*P*	*Q*
bus 2	1.2	0.2
bus 3	1.2	0.5