Proceedings of the VI Brazilian Conference on Neural Networks – VI Congresso Brasileiro de Redes Neurais
pp. 91-96, June 2-5, 2003 – Centro Universitário da FEI, São Paulo – SP - Brazil

A Neural Network for Transient Stability Analysis and Preventive Control of Electric Energy Systems

Anna Diva P. Lotufo[1], Mara Lúcia Martins Lopes[1], Carlos R. Minussi[1]
[1]Universidade Estadual Paulista - UNESP - Departamento de Engenharia Elétrica -
Câmpus de Ilha Solteira - 15385-000, Ilha Solteira, São Paulo, Brasil
E-mails: annadiva@dee.feis.unesp.br, mara@dee.feis.unesp.br, minussi@dee.feis.unesp.br

Abstract

This work presents a procedure for transient stability analysis and for preventive control of electric power systems, formulated by multilayer feedforward neural network. The neural network training is realised using the Backpropagation algorithm with fuzzy controller. The fuzzy controller is used to provide a faster convergence and more precise results, if compared to the traditional Backpropagation algorithm. The adapting of the training rate, is effectuated using the information of the global error, and global error variation. After finishing the training, the neural network is capable to estimate the security margin, and the sensitivity analysis. With these information it is possible to develop a method for the realisation of the security correction (preventive control) for levels considered appropriate to the system, based on generation reallocation and load shedding. To illustrate the proposed methodology it is presented an application considering a multi-machine power system.

1. Introduction

Transient stability analysis is one of the principal studies used in EPS (Electric Power Systems). It is a procedure to evaluate the effects provoked by perturbations that cause great excursions on the angles of the synchronous machines, e.g., short-circuit, operation outage/input of electric equipment. In this case, the model of the system is described by a set of algebraic and non-linear differential equations. On the unstable cases and/or being violated the capacity limit of the equipment, it is necessary to adopt providences that can lead the system to a secure state, known as security control. The methods for dynamical preventive control have appeared recently and the publications available in the literature are not enough [1], [2], [3], [4], [5], [6] and [7] among others.

Therefore, this work develops a methodology based on neural networks [8] to analyse the transient stability – considering short-circuit faults with transmission line outages – and, principally for the sensitivity analysis of EPS, that represent the necessary instruments to do the preventive control. Neural networks are important resources to treat the preventive control problem, considering that once the training is finished (off-line activity), the analysis can be concluded almost without computational effort (basically the calculus with the input and output of the neural network), and can be used for applications in real time. It is emphasised that the sensitivity calculus is effectuated without computational effort. It is also emphasised that, to obtain the sensitivity model by conventional procedures, it involves a great quantity of complex calculus of matrices, consuming much time, principally for applications in large systems.

The neural network used is a non-recurrent multilayer one with training by BP (Backpropagation) algorithm [9] and [10]. The BP algorithm training rate is adjusted by a fuzzy controller [11] and [12], monitoring the global error and the global error variation during the training. It is an optimal mechanism that reduces the convergence time and improves the precision of the results, as observed in [12]. The variables used on the training are causal variables of a problem of transient stability analysis (active and reactive nodal electric power) (input neural network stimulus) and the security margins (output neural network stimulus) generated using the PEBS (Potential Energy Boundary Surface) iterative method [13], microcomputer version. The security margin expressed in function of total energy, can be interpreted as being a measure of the distance in relation to the condition of the instability of the system. The sensitivity model is referred to the relation with the security margin and the nodal electric power. Thus, it can be evaluated the generation reallocation and load cut necessary for obtaining a secure state of the system, this is, a security level considered adequate for transient stability. For testing the proposed methodology it is presented an application considering a multi-machine system.

2. System Model

Considering an Electrical Power System composed of *ns* synchronous machines, the dynamical behavior of the *i*-th machine, related to CA (Center of Angles), is described by the following differential equation (classical model) [14] and [15]:

$$M_i \ \ddot{\theta}_i - P_i(\theta) = 0, \ i \in N \tag{1}$$

where:

$P_i(\theta) \ = \ Pm_i - Pe_i - (M_i \ PCOA) / MT;$ (2)

$M_i \ = \ 2 H_i / \omega s;$

$\omega s \ \triangleq$ synchronous speed of the rotor;

$= 2\pi f_0;$

H_i = inertia constant (s);

f_0 = nominal frequency of system (Hz);

Pm_i = mechanical power of input (pu);

Pe_i = electrical power of output (pu);

θ_i $\triangleq$ rotor angle of i-th synchronous machine related to CA (electrical radians);
= $\delta_i - \delta_0$;

δ_i = rotor angle of i-th synchronous machine in relation to synchronously rotating reference frame (in electrical radians);

δ_0 = $\sum_{j \in N} M_j \, \delta_j$;

$PCOA$ $\triangleq$ accelerating power of CA;
= $\sum_{j \in N} (Pm_j - Pe_j)$;

MT = $\sum_{j \in N} M_j$;

N $\triangleq$ index set of synchronous machines that comprise the system;
= $\{1, 2, \ldots, ns\}$;

ns = number of electrical synchronous machines.

3. Transient Stability Analysis

The transient stability analysis of EPS, considering a contingency of index r, is effectuated using the security margin criterion [2], [13], [15] and [16]:

$$M_r = (Ecrit_r - Ee_r) / Ecrit_r \tag{3}$$

where:

$Ecrit_r$ = total critical energy of the system;

Ee_r = total energy of the system evaluated on the instant of the fault elimination (te).

The critical energy ($Ecrit$), and the critical time ($tcrit$), is determined by the iterative PEBS method [11] and [13], or another procedure that presents a similar result, principally in relation to precision. The total energy, related to system (1), is given by [11], [13] and [15]:

$$E(\theta, \omega) = Ec(\omega) + Ep(\theta) \tag{4}$$

where:

$Ec(\omega)$ = kinetics energy;

$$= 1/2 \sum_{i \in N} M_i \, \omega_i^{\,2} ; \tag{5}$$

$Ep(\theta)$ = potential energy;

$$= - \sum_{i \in N} \int_{\theta_i^p}^{\theta_i} P_i(\theta) \, d\theta_i \, . \tag{6}$$

Then, the transient stability for the r-th contingency is evaluated by the security margin on the following way [2] and [13]:

- if $M_r \geq 0$, the system is considered *stable*, for transient stability;
- if $M_r < 0$, the system is considered *unstable*, for transient stability.

4. Dynamic Preventive Control

Considering a list composed of S contingencies, the security margin of the system must satisfy the following relation [2], [4] and [17]:

$$M \geq Mmin \tag{7}$$

where:

$Mmin$ = minimum limit of the security margin of the system ($Mmin > 0$);

M $\triangleq$ min (M_r , $r = 1, 2, \ldots, S$).

The control actions must cause modifications on the security margins such as, the following relations must be satisfied [2], [4] and [17]:

$$M_r = (M_r^{\,0} + \Delta M_r) \geq Mmin \, , \; r = 1, 2, \ldots, S \tag{8}$$

where:

M_r = security margin referred to the r-th contingency.

The necessary changing (ΔM_r) to correct the security margin – in function of a vector $\boldsymbol{X}$ – is estimated by the sensitivity theory, of first order, according to [2], [4] and [17]:

$$\Delta M_r \;\cong\; \langle \, \partial M_r / \partial \boldsymbol{X}, \Delta \boldsymbol{X} \, \rangle \tag{9}$$

or

$$\Delta tcrit_r \cong \langle \, \partial \, tcrit_r / \partial \boldsymbol{X}, \Delta \boldsymbol{X} \, \rangle \tag{10}$$

where:

$\partial M_r / \partial \boldsymbol{X}$ = sensitivity of the security margin in relation to the vector $\boldsymbol{X}$;

$\partial tcrit_r / \partial \boldsymbol{X}$ = sensitivity of the critical time in relation to the vector $\boldsymbol{X}$;

$\Delta \boldsymbol{X}$ = vector corresponding to the changing on the components of vector $\boldsymbol{X}$.

The vector $\boldsymbol{X}$, in this work, is represented by the nodal active power. The sensitivity $\partial M_r / \partial \boldsymbol{X}$ is developed in Section 6 by neural networks.

5. Neural Network Structure

The i-th output element (neuron) [8] is a linear combination of the element inputs x_j that are connected to the element i by the weight w_{ij}:

$$\vartheta_i = \sum_{j} w_{ij} \, x_j \tag{11}$$

Each element can have a bias w_0 fed by an extra constant input $x_0 = +1$. The linear output ϑ_i is finally converted in a nonlinearity, as a sigmoid and relay [10], etc. The relay functions are appropriated for binary systems, while the sigmoid functions can be employed for both continuous and binary systems.

The training of this neural network is realized as shown in the Appendix A.

6. Sensitivity Analysis by Neural Networks

The BP algorithm is initialised presenting a pattern $\boldsymbol{X} \in \Re^n$ to the network, that gives an output $\boldsymbol{Y} \in \Re^m$. In the sequence it is calculated an error in each output (the difference with the desired value and the output).

Next step is to determine the error propagated in inverse way by the network associated to the partial derivative of the quadratic error of each element related to the weights, and finally to adjust the weights in each element. Then, a new pattern is presented, and the pattern must be repeated until convergence ($|$error$|$ $\leq$ predefined tolerance). Once concluded this step, the training mechanism do not actuate, including the fuzzy controller. This way, the network is able to generalise, this is, applying any input pattern vector, propagating the signal on the straight sense (input to output), it results on the output an evaluation of the analysis (diagnosis), providing a mapping, $\boldsymbol{X} \to \boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X})$, $\boldsymbol{X} \in \Re^n$ e $\boldsymbol{Y} \in \Re^m$.

Using this idea, it is estimated the derivatives of the output variables (sensitivity analysis) in relation to the input vector components, using a neural network structure trained as described as follows. The sensitivity analysis, by neural networks, is used in this work to obtain $\partial M_r/\partial \boldsymbol{X}$ (problem defined by equation (9)). Taking into account that, for the solution of the preventive control problem, it is adopted the generation reallocation and load shedding (according to the formulation described in Section 4), the vector $\boldsymbol{X}$ corresponds to the nodal active power ($\boldsymbol{P}$).

Thus, consider $\boldsymbol{X}^k$ and $\boldsymbol{Y}^k$ as being the k-th pair of input and output vector of the neural network. Consider, too, the non-recurrent neural network shown in Figure 1. It is the representation of a network composed of three layers, where the variables on the principal points of the network and the weight matrices are defined. The input and output layers have n and m neurons, respectively, where:

n = dimension of input vector $\boldsymbol{X}^k$;
m = dimension of output vector $\boldsymbol{Y}^k$.

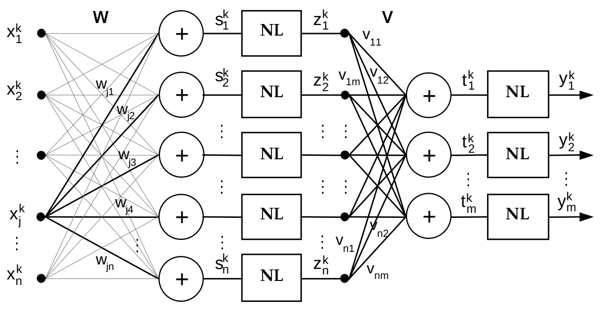


Figure 1. Non-recurrent neural network

It is desired to obtain the partial derivative of y_p^k (p-th component of vector $\boldsymbol{Y}^k$) in relation to x_j^k (j-th component of the input vector $\boldsymbol{X}^k$). To obtain these partial derivatives, it is necessary to obtain the intermediate partial derivatives (on the output of the neurons of the hidden layers) of the neural network. Thus, the calculus of partial derivatives of z_i^k (i-th component of output vector $\boldsymbol{z}^k$), in relation to x_j^k, is obtained in the following way [17]:

$$\partial z_i^k/\partial x_j^k = \lambda/2(1-\{z_i^k\}^2) \text{ (for sigmoid function (A3))} \quad (12)$$

or

$$\partial z_i^k/\partial x_j^k = \lambda\, z_i^k(1-z_i^k) \quad \text{(for sigmoid function (A4))} \quad (13)$$

where:
x_j^k = j^{th} component of input vector $\boldsymbol{X}^k$;
$\boldsymbol{X}^k$ = $[\,x_1^k\ x_2^k\ \ldots\ x_n^k\,]^{\mathrm{T}}$;
k = index referred to the k-th pattern vector.

The p-th intermediate output (input of the sigmoid function) of the output layer of the neural network is expressed by:

$$t_p^k = \{\boldsymbol{z}^k\}^{\mathrm{T}}\, \boldsymbol{v}_p\,, \quad p = 1,2,\ldots,m \quad (14)$$

where:
$\boldsymbol{z}^k$ = $[z_1^k\ z_2^k\ \ldots\ z_n^k]^{\mathrm{T}}$;
$\boldsymbol{v}_p$ = $[v_{1p}\ v_{2p}\ \ldots\ v_{np}]^{\mathrm{T}}$.

Thus:

$$\partial y_p^k / \partial x_j^k = \{\partial y_p^k(\lambda, t_p^k)/\partial t_p^k\}\ \partial t_p^k / \partial x_j^k \quad (15)$$

as:

$$\partial t_p^k/\partial x_j^k = \{\boldsymbol{v}_p^k\}^{\mathrm{T}}\, \partial \boldsymbol{z}^k/\partial x_j^k \quad (16)$$

where:

$$\partial \boldsymbol{z}^k/\partial x_j^k = [\partial z_1^k/\partial x_j^k\ \partial z_2^k/\partial x_j^k\ \ldots\ \partial z_n^k/\partial x_j^k]^{\mathrm{T}}$$

Then, substituting equation (A3), or (A4) on equation (15), it is obtained:

$$\partial y_p^k/\partial x_j^k = \lambda^2/4(1-\{y_p^k\}^2)\{\boldsymbol{v}_p\}^{\mathrm{T}}\, \boldsymbol{b}_j^k \text{ (for sigmoid function (A3))} \quad (17)$$

or

$$\partial y_p^k/\partial x_j^k = \lambda^2\, y_p^k\,(1-y_p^k)\ \{\boldsymbol{v}_p\}^{\mathrm{T}}\, \boldsymbol{b}_j^k \text{ (for sigmoid function (A4))} \quad (18)$$

where:

$$\boldsymbol{b}_j^k = [(1-\{z_1^k\}^2)\, w_{j1}\quad (1-\{z_2^k\}^2)\, w_{j2}\quad \ldots\quad (1-\{z_n^k\}^2)\, w_{jn}]^{\mathrm{T}} \text{ (for sigmoid function (A3))}; \quad (19)$$

or

$$\boldsymbol{b}_j^k = [z_1^k\,(1-z_1^k)\, w_{j1}\quad z_{2k}\,(1-z_1^k)\, w_{j2}\quad \ldots\quad z_n^k\,(1-z_n^k)\, w_{jn}]^{\mathrm{T}} \text{ (for sigmoid function (A4))}. \quad (20)$$

7. Application

Faults like three-phase short-circuit with time of fault elimination equal to 0.15s (9 cycles considering a 60Hz operation), followed by the outage of the transmission line are considered. The one-line diagram system is shown in Figure 2 (Appendix B). This system is composed of 10 synchronous machines, 73 transmission lines and 45 buses, based on the configuration of a southern Brazilian system.

The neural network training was effectuated considering a set of 158 generation and load profiles and respective security margin. Each profile corresponds to a generation redispatch in relation to a base case in a random way to attend the demand, also fixed in a random way in each bus. The universe of the load variation is between 80 and 120% ($\pm$20%), in relation to the nominal load of the system. Therefore, each profile is generated considering a variation percentile around the nominal state (base case) and a respective seed to the random sequence generation process. Thus, to a same percentile, different generation seeds generates different generation dispatchs of different load profiles. This proceeding generates an adequate set of patterns to the training phase.

The contingency, adopted as an example, corresponds to a three phase short-circuit at bus 39, with outage of transmission line between busses 39 and 40. This contingency was chosen due to be the most critical one among the possible faults. Nevertheless, other contingencies can be included, only leading to a calculation increase.

It is emphasized that the neural network, not only effectuates the stability analysis (margin security evaluation), but gives the sensibility analysis model $(\partial M_r/\partial \mathbf{X})$ referred to the analyzed contingency. This sensibility vector $\partial M_r/\partial \mathbf{X}$ is used to define the generation reallocation, and the load shedding, necessary to correct the security margin to predefined levels, i.e. for levels considered secure, considering the transitory stability. It is considered the minimum security margin $Mmin$ = 0.3. These results are illustrated in Table 1.

Table 1. Analyzed results before and after the control action (reallocation generation / load shedding)

State	Item	Value
Initial	Initial Security Margin (M^0)	–1.416
Identification of more sensitivity busses	More sensitivity generation bus	9
	More sensitivity load bus	40
Sensibility Analysis	Sensibility coefficient of bus 9 ($\partial M/\partial P_9$)	–1.612
	Sensibility coefficient of bus 40 ($\partial M_r/\partial P_{40}$)	0.223
Control action	Generation reallocation at bus 9	–1.08
	Load Shedding at bus 40	–1.08
Final	Final Security margin (M^f)	0.32

Suppose it is desired to correct the security margin from M^0 = –1.416 to $M \geq Mmin$ = 0.3. Then, the control action (generation redispatch / load shedding) provides a displacement of the security margin $\Delta M \geq$ 1.716. This goal, by sensibility analysis, is obtained, for example, by a reduction on the generated power of synchronous machine number 9, and a reduction of the load demand associated to bus 40. This reduction correspondent to 1.08 pu, produces a final security margin of 0.32, that is a good approximation of the goal to be reached ($M \geq Mmin$).

8. Conclusion

It was proposed in this work, a procedure to analyse the transient stability and preventive control of EPS formulated by non-recurrent neural networks. The neural network training was done using the BP algorithm with fuzzy controller. The fuzzy controller gives a faster convergence and more precise results, [12], when compared to the traditional BP, by adjusting the training rate, using the information of the error, and global variation error, Once finished the training, the network is able to evaluate the security margin and sensitivity analysis. With this information, it was possible to develop a procedure to do the correction of the security (preventive control) for levels considered adequate for the system. The approach presented is a preliminary result, which is a beginning point to more elaborated preventive control approaches (stability analysis considering a set of contingencies, optimal generation redispatch, etc.).

References

[01] I. C. Decker and J. E. G. Castro. Dynamical Security Improvement of Electric Power Systems Using Optimization. *XI CBA*, São Paulo - SP, Vol. 1, 1996, pp. 191–196. (In Portuguese).

[02] L. G. S. Fonseca and C. R. Minussi. Reallocation Generation For Dynamical Security Control of Electric Power System by Sensitivity Analysis. *VIII CBA*, Belém - PA, 1990, pp.735–740. (In Portuguese).

[03] A. A. Fouad and T. Jianzhong. Stability Constrained Optimal Rescheduling of Generation. *IEEE Transactions on Power Systems*, Vol. 8, No. 1, 1993, pp. 105–112.

[04] C. R. Minussi and W. Freitas Filho. Sensitivity Analysis For Transient Stability. *IEE Proceedings on Generation, Transmission And Distribution*, Vol. 145, No. 6, 1998, pp. 669-674.

[05] C. R. Minussi, L. M. Ramos, S. C. Marchiori, M. L. M. Lopes, and A. D. P. Lotufo. Neural Networks to Transient Stability Analysis of Electrical Power Systems. *Proceedings of the V Brazilian Conference on Neural Networks*, Rio de Janeiro-RJ, Brazil, April-2001, pp. 31-36.

[06] V. Vittal, E. Z. Zhou, C. Hwang, and A. A. Fouad. Derivation on Stability Limits Using Analytical Sensitivity of The Transient Energy Margin. *IEEE Transactions on Power Systems*, Vol. 4, No. 2, 1989, pp. 1363–1372.

[07] Y. Xue, Th. Van Cutsem, and M. Ribbens-Pavella. Real-time Analytic Sensitivity Method For Fast Transient Stability Assessment of Large Power Systems. *IEE Proceedings*, Vol. 135, Pt. C. No. 2, 1988, pp. 107–116.

[08] S. V. Kartalopoulos. *Understanding Neural Net-woks and Fuzzy Logic.* IEEE Press, New York, 1996.

[09] P. J. Werbos. *Beyond Regression: New Tools For Prediction And Analysis in The Behavioural Sciences.* Master Thesis, Harvard University, Cambridge, MA, U.S.A., 1974.

[10] B. Widrow and M. A. Lehr. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, And

Backpropagation. *Proceeding of the IEEE*, Vol. 78, No. 9, 1990, pp. 1415-1442.
[11] P. Arabshahi, J. J. Choi, R. J. Marks II, and T. P. Caudell. Fuzzy Parameter Adaptation in Optimization. *IEEE Computational Science & Engineering*, Spring 1996, pp. 57-65.
[12] M. L. M. Lopes, C. R. Minussi, and A. D. P. Lotufo. A Fast Electric Load Forecasting Using Neural Networks. *Proceedings of 43rd Midwest Symposium on Circuits and Systems*, Lansing, MI, U.S.A. 2000. vol. 1, pp.1–4.
[13] L. G. S. Fonseca and I. C. Decker. Iterative Algorithm For Critical Energy Determination in Transient Stability of Power System. *IFAC – Symposium Planning & Operation in Electric Energy System*, Rio de Janeiro, RJ, Brazil, 1985, pp. 483-489.
[14] P. M. Anderson and A. Fouad. *Power System Control And Stability.* IOWA State University Press, USA, 1977.
[15] M. A. Pai. *Power System Stability.* North Holland Control Series, 1981.
[16] T. Athay, V. R. Sherket, R. Podmore, S. Virmani, and C. Puech. Transient Energy Stability Analysis. *Conference on 'System Engineering For Power'*, Davos, Switzerland, 1979.
[17] A. D. P. Lotufo, M. L. M. Lopes, and C. R. Minussi. Dynamic Preventive Control of Electric Power Systems By Neural Networks. *V Induscon*, Salvador-BA, 2002, Paper Code:144.
[18] T. Terano, K. Asai, and M. Sugeno. *Fuzzy Systems Theory And its Application.* Academic Press, N. York, USA, 1987.

Appendix A
Neural Network Training

The initial weights are general adopted as random numbers [10]. The BP algorithm consists of adapting the weights such as the quadratic error of the network to be minimised. The sum of the instantaneous quadratic error of each neuron on the last layer (network output) is given by [10]:

$$\varepsilon^2 = \sum_{i=1}^{no} \varepsilon_i^{\,2} \qquad \text{(A1)}$$

where:

$\varepsilon_i = d_i - y_i$;
d_i = desired output of the i-th element of the last layer;
y_i = output of the i-th element of the last layer;
no = number of neurons of the last layer.

Considering the i-th neuron of the network and using the gradient descendent method [8] and [10], the weight adjustments are formulated by [10]:

$$\Gamma_i(h+1) = \Gamma_i(h) + \phi_i(h) \qquad \text{(A2)}$$

where:

$\phi_i(h)$ = $-\gamma[\nabla_i(h)]$;
γ = stability control parameter or training rate;
$\nabla_i(h)$ = gradient of quadratic error related to the weights of neuron i;
Γ_i Δ vector containing the weights of i-th neuron;
= $[\, w_{0i} \;\; w_{1i} \;\; w_{2i} \; . \; . \; . \; w_{ni} \,]^T$;
h = actualisation index of the adaptive process.

The direction adopted in equation (A2) to minimise the objective function of the quadratic error corresponds to the opposite direction of the gradient. The parameter γ determines the length of vector $\phi_i(h)$. The sigmoid function is defined by [8] and [10]:

$$y_i \,\underline{\Delta}\, y_i(\lambda, \vartheta_i) = \{(1+exp(-\lambda\,\vartheta_i)\}/\,\{(1+exp(-\lambda\,\vartheta_i)\} \qquad \text{(A3)}$$

or

$$y_i \,\underline{\Delta}\, y_i(\lambda, \vartheta_i) = 1\,/\,\{(1 + exp\,(-\lambda\,\vartheta_i)\} \qquad \text{(A4)}$$

where:

λ = constant that determines the inclination of function y_i.

The variation of equations (A3) and (A4) is, respectively, (–1,+1) and (0,+1).

Then, calculating the gradient as shown on equation (A2), considering the sigmoid function defined by equations (A3) or (A4) and the moment term [10], it is obtained the following schema of the adaptation of the weights [10]:

$$\Pi_{ij}(h+1) = \Pi_{ij}(h) + \Delta\Pi_{ij}(h) \qquad \text{(A5)}$$

where:

$\Delta\Pi_{ij}(h)$ = $2\,\gamma(1-\eta)\,\beta_j\,x_i + \eta\,\Delta\Pi_{ij}(h-1)$; (A6)
Π_{ij} = weight corresponding to the linking of the i-th and the j-th neuron;
γ = training rate;
η = moment constant ($0 \leq \eta < 1$) [10].

If the j-th element is on the last layer, then:

$$\beta_j = \sigma_j \;\, \varepsilon_j \qquad \text{(A7)}$$

where:

σ_j $\underline{\Delta}$ derivative of the sigmoid function, given by equation (A3) or (A4), respectively, related to ϑ_j;
= $\lambda/2\,(1 - y_j^{\,2})$; (A8)
= $\lambda\; y_j\,(1 - y_j^{\,2})$. (A9)

If the j-th element is on the other layers, then:

$$\beta_j = \sigma_j \sum_{k \in \Gamma(j)} w_{jk}\,\beta_k \qquad \text{(A10)}$$

where:

$\Gamma(j)$ = set of index of elements that are on the next layer of the j-th element layer and are linked to the j-th element.

The parameter γ, that is used as a stability controller of the iterative process is dependent of λ. The network weights are randomly initiated, considering the interval [0,1]. By convenience, the parameter γ (training rate) is redefined as follows:

$$\gamma = \gamma^* \,/\, \lambda \qquad \text{(A11)}$$

Substituting equation (A11) in equation (A6), it is "cancelled" the dependency of the amplitude of σ related to λ. The amplitude σ is maintained constant for any λ. This alternative is important, considering that λ only actuates on the left and right tails of σ. Then, equation (A6) is written as follows:

$$\Delta\Pi_{ij}(h) = \{2\ \gamma^*(1-\eta)\ \beta_j/\lambda\}\ x_i + \eta\ \Delta\Pi_{ij}(h-1). \quad \text{(A12)}$$

The BP algorithm is considered in the technical literature a benchmark in precision, although its convergence is very slow. Thus, this work proposes the adjustment of the training rate γ^* during the convergence process, to reduce the training time during the execution. The adjustment γ^* is effectuated by a procedure based on a fuzzy controller [18]. The basic idea of the methodology consists in determining the system state, defined as global error εg and the variation of the global error $\Delta\varepsilon g$, taking as an objective a control structure that leads the error to zero, with few iterations, when compared to the conventional procedures. In this work, the control is formulated using the concept of fuzzy logic [18]. The global error εg and its variation $\Delta\varepsilon g$ are the components of the system state, and $\Delta\gamma^*$ is the control action that must be executed on the system. Initially, the global error is defined as:

$$\varepsilon g = \sum_{j=1}^{np} \sum_{i=1}^{no} \varepsilon_i^{\,2} \quad \text{(A13)}$$

where:

εg = global error of the neural network;
np = number of pattern vectors.

The global error is calculated in each iteration, and the parameter γ^*, adjusted by an increase $\Delta\gamma^*$ determined by fuzzy logic. The system state and the control action are defined as:

$$E^q = [\varepsilon g^q \ \ \Delta\varepsilon g^q]^T, \quad \text{and} \quad u^q = \Delta\gamma^{*q} \quad \text{(A14)}$$

where:

q = index of the current iteration.

For a very large pattern input $\boldsymbol{X}$, εg and $\Delta\varepsilon g$ can saturate. Then, the adaptive control is effectuated using an exponential decreasing function applied to the response of the fuzzy controller. This way, the adaptive controller is given by:

$$\Delta\gamma^{*q} = exp\,(-\alpha\, q)\ \Delta\psi^q \quad \text{(A15)}$$

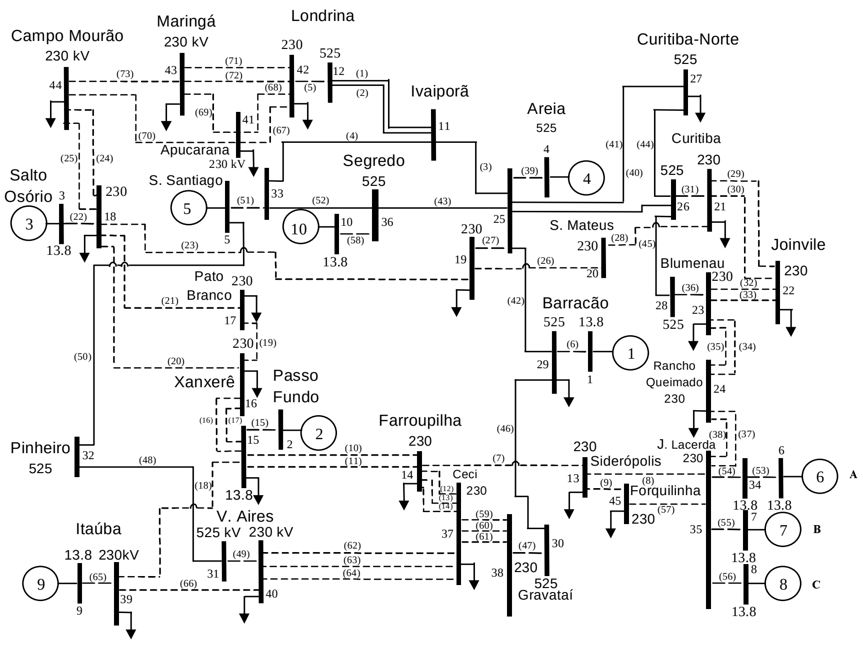
where:

α = a positive arbitrary number;
$\Delta\psi^q$ = variation of the fuzzy controller on the instant q.

This parameter is used to adjust the set of the network weights referred to the subsequent iteration. The process must be repeated until the training be concluded.

Appendix B

One-line Diagram of Power System



( ) Transmission line number.

Figure 2. Representation of test systems