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

Anna Diva P. Lotufo¹, Mara Lúcia Martins Lopes¹, Carlos R. Minussi¹
¹Universidade 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 a 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 provisions 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]:

\[ M_i \ddot{\theta}_i - P_i(\theta) = 0, \quad i \in N \]  

(1)

where:

\[ P_i(\theta) = P_{mi} - P_e - (M, PCOA) / MT; \]  

(2)

\[ M_i = 2 H_i / \omega_s; \]

\[ \omega_s \Delta \text{synchronous speed of the rotor;} \]

\[ H_i \Delta \text{inertia constant (s);} \]

\[ f_0 \Delta \text{nominal frequency of system (Hz);} \]

\[ P_{mi} \Delta \text{mechanical power of input (pu);} \]
\( P_{e_i} \) = electrical power of output (pu); 
\( \theta_i \) = rotor angle of \( i \)-th synchronous machine related to CA (electrical radians); 
\( \delta \) = rotor angle of \( i \)-th synchronous machine in relation to synchronously rotating reference frame (in electrical radians); 
\( \delta_i \) = \( \sum_{j \in N} M_j \delta_j \); 
\( PCOA \) = accelerating power of CA; 
\( MT \) = \( \sum_{j \in N} M_j \); 
\( N \) = 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 effected using the security margin criterion [2], [13] and [16]:

\[ M_r = (E_{crit} - E_{te}) / E_{crit} \]  

where:

\( E_{crit} \) = total critical energy of the system;
\( E_{te} \) = total energy of the system evaluated on the instant of the fault elimination (te).

The critical energy \( E_{crit} \), and the critical time \( (t_{crit}) \), 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]:

\[ E(\theta, \omega) = E_c(\omega) + E_p(\theta) \]  

where:

\( E_c(\omega) \) = kinetics energy; 
\( = \frac{1}{2} \sum_{i \in N} M_i \omega_i^2 \);  
\( E_p(\theta) \) = potential energy; 
\( = - \sum_{i \in N} \left[ \omega_i \theta_i \right] P_i(\theta) d\theta_i \).

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]:

\[ M \geq M_{\text{min}} \]  

where:

\( M_{\text{min}} \) = minimum limit of the security margin of the system \( (M_{\text{min}} > 0) \); 
\( M \) = \( \min(M_r, r = 1, 2, \ldots, S) \).

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

\[ M_r = (M_r^0 + \Delta M_r) \geq M_{\text{min}}, r = 1, 2, \ldots, S \]  

where:

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

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

\[ \Delta M_r \equiv \left( \partial M_r / \partial X, \Delta X \right) \]  

or

\[ \Delta t_{\text{crit}} \equiv \left( \partial t_{\text{crit}} / \partial X, \Delta X \right) \]  

where:

\( \partial M_r / \partial X \) = sensitivity of the security margin in relation to the vector \( X \); 
\( \partial t_{\text{crit}} / \partial X \) = sensitivity of the critical time in relation to the vector \( X \); 
\( \Delta X \) = vector corresponding to the changing on the components of vector \( X \).

The vector \( X \), in this work, is represented by the nodal active power. The sensitivity \( \partial M_r / \partial X \) 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 \( x_i \) that are connected to the element \( i \) by the weight \( w_i \):

\[ \vartheta_i = \sum_j w_i x_j \]  

Each element can have a bias \( w_0 \) fed by an extra constant input \( x_0 = +1 \). The linear output \( \vartheta_i \) 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 \( X \in \mathbb{R}^n \) to the network, that gives an output \( Y \in \mathbb{R}^m \). 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, \( X \rightarrow Y = f(X), X \in \mathbb{R}^n \) e \( Y \in \mathbb{R}^m \).

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 \( \partial M_i/\partial X \) (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 \( X \) corresponds to the nodal active power \( (P_t) \).

Thus, consider \( X^k \) and \( Y^k \) 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:

\[
\begin{align*}
 n & = \text{dimension of input vector } X^k; \\
 m & = \text{dimension of output vector } Y^k.
\end{align*}
\]

\[\text{Figure 1. Non-recurrent neural network}\]

It is desired to obtain the partial derivative of \( y_p^k \) \((p\)-th component of vector \( Y^k \)) in relation to \( x_i^k \) \((j\)-th component of the input vector \( X^k \)). 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 \( z^k \)), in relation to \( x_i^k \), is obtained in the following way [17]:

\[
\frac{\partial y_p^k}{\partial x_i^k} = \lambda \frac{2}{(1 - \{z_i^k\}^2)} (\text{for sigmoid function (A3)) (12)}
\]

or

\[
\frac{\partial y_p^k}{\partial x_i^k} = \lambda \frac{z_i^k}{(1 - \{z_i^k\}^2)} (\text{for sigmoid function (A4)) (13)}
\]

where:

\[
x_i^k = j^{th} \text{component of input vector } X^k; \\
X^k = [x_1^k, x_2^k, \ldots, x_n^k]^T; \\
k = \text{index referred to the } k\text{-th pattern vector.}
\]

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

\[
t_p^k = [\phi^k]^T v_p, \quad p = 1, 2, \ldots, m \quad (14)
\]

Thus:

\[
\frac{\partial y_p^k}{\partial x_i^k} = \{\partial t_p^k / \partial t_{j_1}^k\} \cdot \frac{\partial t_{j_1}^k}{\partial x_i^k} \quad (15)
\]

as:

\[
\frac{\partial t_{j_1}^k}{\partial x_i^k} = [\partial z_{j_1}^k / \partial x_i^k, \partial z_{j_2}^k / \partial x_i^k, \ldots, \partial z_{j_n}^k / \partial x_i^k]^T \quad (16)
\]

where:

\[
\partial z_{j_1}^k / \partial x_i^k = [\partial z_{j_1}^k / \partial z_{j_2}^k, \partial z_{j_1}^k / \partial z_{j_3}^k, \ldots, \partial z_{j_n}^k / \partial z_{j_1}^k]^T
\]

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

\[
\frac{\partial y_p^k}{\partial x_i^k} = \lambda / 4 (1 - \{y_p^k\}^2)\{v_p\}^T b_i^k \quad (17)
\]

or

\[
\frac{\partial y_p^k}{\partial x_i^k} = \lambda / 4 (1 - \{y_p^k\}^2)\{w_p\}^T b_i^k \quad (18)
\]

where:

\[
b_i^k = [(1 - \{z_{j_1}^k\}^2) w_{j_1}, (1 - \{z_{j_2}^k\}^2) w_{j_2}, \ldots, (1 - \{z_{j_n}^k\}^2) w_{j_n}]^T \quad (\text{for sigmoid function (A3))}; (19)
\]

or

\[
b_i^k = [(1 - \{z_{j_1}^k\}^2) w_{j_1}, z_{j_2} (1 - \{z_{j_2}^k\}^2) w_{j_2}, \ldots, z_{j_n} (1 - \{z_{j_n}^k\}^2) w_{j_n}]^T \quad (\text{for sigmoid function (A4))}. (20)
\]

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% (+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 dispatches 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 \( \partial M / \partial X \) referred to the analyzed contingency. This sensibility vector \( \partial M / \partial X \) 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 \( M_{min} = 0.3 \). These results are illustrated in Table 1.

<table>
<thead>
<tr>
<th>State</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>Initial Security Margin (( M^0 ))</td>
<td>−1.416</td>
</tr>
<tr>
<td>Identification of more</td>
<td>More sensitivity generation bus</td>
<td>9</td>
</tr>
<tr>
<td>sensitivity busses</td>
<td>More sensitivity load bus</td>
<td>40</td>
</tr>
<tr>
<td>Sensibility Analysis</td>
<td>Sensibility coefficient of bus 9</td>
<td>−1.612</td>
</tr>
<tr>
<td></td>
<td>( (\partial M / \partial P_i) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensibility coefficient of bus 9</td>
<td>0.223</td>
</tr>
<tr>
<td>Control action</td>
<td>Generation reallocation at bus 9</td>
<td>−1.08</td>
</tr>
<tr>
<td></td>
<td>Load Shedding at bus 40</td>
<td>−1.08</td>
</tr>
<tr>
<td>Final</td>
<td>Final Security margin (( M ))</td>
<td>0.32</td>
</tr>
</tbody>
</table>

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 redispach, etc.).

References

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

be minimised. The sum of the instantaneous quadratic numbers [10]. The BP algorithm consists of adapting weight adjustments are formulated by [10]:

\[ w_{n} = \frac{\partial E}{\partial w_n} \]

Using the gradient descent method [8] and [10], the

\[ \frac{\partial E}{\partial w_n} = \sum \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \theta_i} \]

where:

\[ \theta_i = \text{weight corresponding to the linking of the} \]

\[ \lambda = \text{constant that determines the inclination of} \]

\[ \beta_i = \sigma \varepsilon_i \]

\[ \sigma = \text{derivative of the sigmoid function, given} \]

\[ \Gamma_i = \text{vector containing the weights of} \]

\[ h = \text{actualisation index of the adaptive process.} \]

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 \( \gamma \) determines the length of vector \( \phi(h) \). The sigmoid function is defined by [8] and [10]:

\[ y_i \Delta y_i(\lambda, \theta_i) = [1 + \exp(-\lambda \theta_i)] / [(1 + \exp(-\lambda \theta_i))] \]

\[ y_i \Delta y_i(\lambda, \theta_i) = 1 / [(1 + \exp(-\lambda \theta_i))] \]

where:

\[ \lambda = \text{parameter} \]

\[ \gamma = \text{training rate;} \]

\[ \eta = \text{moment constant (0} \leq \eta < 1 \] [10].

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]:

\[ \Pi_i(h+1) = \Pi_i(h) + \Delta \Pi_i(h) \]

\[ \Delta \Pi_i(h) = 2 \gamma (1 - \eta) \beta_i \cdot x_i + \eta \Delta \Pi_i(h - 1) \]

\[ \Pi_i = \text{weight corresponding to the linking of the} \]

\[ \gamma = \text{training rate;} \]

\[ \eta = \text{moment constant (0} \leq \eta < 1 \] [10].

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

\[ \beta_j = \sigma \varepsilon_j \]

\[ \sigma = \text{derivative of the sigmoid function, given} \]

\[ \Gamma_j(h+1) = \Gamma_j(h) + \phi(h) \]

where:

\[ \phi(h) = -\gamma \nabla_{\epsilon}(h) \]

\[ \gamma = \text{stability control parameter or training rate;} \]

\[ \nabla_{\epsilon}(h) = \text{gradient of quadratic error related to the} \]

\[ w_{i,k} = \sum_{k \in \Gamma(j)} \]

\[ \Gamma(j) = \text{set of index of elements that are on the next} \]

\[ \gamma = \gamma^* / \lambda \]

\[ \lambda = \text{parameter} \]

\[ \gamma = \text{parameter} \]

\[ \lambda = \text{parameter} \]

\[ \gamma = \text{parameter} \]

\[ \lambda = \text{parameter} \]
where:
\[ e_g = \text{global error of the neural network}; \]
\[ n_p = \text{number of pattern vectors}. \]

The global error is calculated in each iteration, and the parameter \( \gamma^p \), adjusted by an increase \( \Delta \gamma^p \) determined by fuzzy logic. The system state and the control action are defined as:
\[ E^q = [e_g^q \ \Delta e_g^q] \quad \text{and} \quad u^q = \Delta \gamma^p \]
(14)

where:
\[ q = \text{index of the current iteration}. \]

For a very large pattern input \( X \), \( e_g \) and \( \Delta e_g \) 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:
\[ \Delta \gamma^p \cdot q = \exp (- \alpha q \Delta \psi^q) \quad (15) \]
where:
\[ \alpha = \text{a positive arbitrary number}; \]
\[ \Delta \psi^q = \text{variation of the fuzzy controller on the instant } q. \]

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

![Diagram of Power System]

( ) Transmission line number.

Figure 2. Representation of test systems