

Avaliação e Melhoria da Margem de Colapso de Tensão Utilizando Rede Neural Artificial

Maury M. Gouvêa Jr.¹, Luiz Danilo B. Terra²

¹Centro Federal de Educação Tecnológica de MG, Av. Amazonas 5253, 30.480-000, Brasil

²Pontifícia Universidade Católica – MG, Av. Dom José Gaspar 500, 30.535-610, Brasil

E-mails: mauryjr@digitus.com.br, siselet@pucminas.br

Abstract

This paper deals with static security assessment and enhancement of power systems. An artificial neural network, which yields the voltage collapse margin for a critical bus is incorporated to an objective function that is optimized through a gradient search procedure. Simulations using the IEEE-30 bus network illustrate the method. The advantages and enhancement in the voltage collapse margin are investigated by varying key parameters during the optimization process.

1. Introdução

O planejamento adequado de sistemas elétricos de potência (SEP) deve garantir uma operação segura e econômica no caso base e em todas as contingências. A instabilidade da tensão, que pode ser causada pela escassez de reservas de potência reativa, é um dos maiores responsáveis pela ocorrência de colapso de tensão em sistemas de potência. Essa evidência é importante para se encontrar uma metodologia que gerencie a potência reativa de forma a proporcionar uma operação segura, sem riscos de colapsos.

Muitas pesquisas têm proposto definir a margem de segurança como a proximidade do ponto de operação ao ponto de colapso do sistema. Existem vários métodos que encontram a margem de colapso de tensão (MCT) de um barramento. Tiranuchit *et al.* [1] propôs a utilização do valor singular mínimo da matriz *Jacobiana* das equações do fluxo de potência como monitoramento da segurança. Ejebe *et al.* [2] propôs um novo procedimento, denominado método da curva ajustada, para calcular a margem de colapso de tensão. Esses métodos são, normalmente, utilizados para análise de estabilidade de tensão, seleção (*screening*) e classificação (*ranking*) de contingências. Quando, num processo de otimização, o objetivo é maximizar as reservas de potência reativa ou melhorar o perfil de tensão do sistema, indicadores de segurança, como a própria MCT, são beneficiados. No entanto, indicadores econômicos, como a perda total de potência ativa, devem ser monitorados, pois aumentos excessivos em fatores de segurança podem ser desnecessários e dispendiosos. A principal desvantagem dos métodos,

anteriormente citados, que calculam a MCT de um barramento é a necessidade de um volume muito grande de cálculos, o que pode torná-los inadequados para uso *on-line* nos centros de controle. Utilizando uma única RNA, tem-se a capacidade de lidar com a complexidade dos SEP com um volume de cálculos muito baixo. Outro problema está no fato de serem utilizados objetivos múltiplos, pois o estado ótimo para um pode não representar o ótimo para outro(s). Deve-se, portanto, encontrar um ponto que satisfaça todos os objetivos ou ponderá-los conforme suas relevâncias.

As exigências com a segurança e o crescimento dos sistemas de potência exigem métodos mais rápidos para análise *on-line*. A tecnologia das redes neurais artificiais (RNA's) representa um desses métodos, tendo sido fortemente pesquisada e aplicada em problemas de controle e identificação a partir do final da década passada. Especificamente em sistemas de potência, as RNA's são utilizadas na avaliação de segurança [3], diagnóstico de falhas [4] e previsão de carga [5]. Sua grande vantagem é trabalhar como uma caixa preta que se adapta aos exemplos de comportamentos (já conhecidos) de um sistema, podendo, posteriormente, prever situações que não fizeram parte dos exemplos apresentados com considerável precisão.

Este trabalho apresenta uma metodologia que utiliza uma RNA apta a ser executada em modo *on-line* na fase de operação de sistemas de potência para avaliação da estabilidade; assim como em seu planejamento, para auxiliar o processo de otimização que tem como objetivo melhorar o perfil de tensão, monitorar e aumentar a margem de colapso em sistemas de potência. A rede neural é direcionada a fornecer a MCT do sistema elétrico e influenciar as ações de controle no decorrer do processo de otimização. Os objetivos – distintos e conflitantes – são tratados por graus de satisfações ponderados por termos lingüísticos. A função objetiva torna-se uma somatória ponderada de funções de pertinência que descrevem as satisfações dos objetivos. A razão desse tratamento é a dificuldade em se encontrar pesos numéricos coerentes que proporcionem um resultado ótimo. Para dar significado aos termos lingüísticos, utiliza-se conjuntos fuzzy [6]. O método é testado e analisado em simulações feitas no sistema IEEE-30 barras [7]. Os efeitos e as vantagens

da melhoria da MCT para o sistema são investigados variando parâmetros decisivos no processo de otimização.

2. Redes Neurais Artificiais

2.1. Estrutura das Redes Neurais Artificiais

As redes neurais artificiais (RNA's) são uma analogia às redes biológicas, simulando seu comportamento em computadores digitais ou em placas de *hardware*. Existem vários modelos de redes neurais que são utilizados conforme a aplicação. Neste trabalho, utiliza-se o modelo *feedforward*, introduzido por Rosenblatt [8] em 1958, composto por neurônios artificiais ligados em camadas sucessivas de forma maciça. Os sinais que são aplicados nas entradas propagam-se camada a camada até a(s) saída(s) passando por cada neurônio.

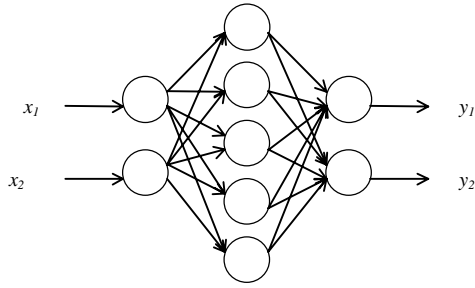


Figura 1: Modelo de Redes *Feedforward*

Os sinais que chegam em cada neurônio são ponderados por coeficientes, denominados *pesos sinápticos*, e somados no corpo principal do neurônio. A saída, definida por uma função não linear, é ativada se a soma ponderada das entradas for maior que um valor denominado limiar, caso contrário a saída não é ativada. Assim, a saída de um neurônio é determinada por:

$$y = f(v) \quad (1)$$

sendo

$$v = \sum_{i=1}^n W_i x_i - \theta \quad (2)$$

sendo W_i o peso sináptico, x_i a entrada e θ o limiar do neurônio com n entradas.

2.2. Treinamento de Redes Neurais Artificiais

As RNA só foram efetivamente aplicáveis depois que Rumelhart *et al* [9] introduziram o algoritmo de treinamento, denominado *Backpropagation*, e após o grande desenvolvimento computacional.

O treinamento ou aprendizagem de uma RNA é feito por exemplos de comportamento (padrões de

treinamento) constituídos por pares entrada/saída que são apresentados nessa fase. No processo de aprendizagem, a RNA se adapta aos estímulos, ajustando os pesos sinápticos para produzir um desempenho para o qual foi projetada. O conjunto de padrões é definido por

$$\Psi = \{(X_l^d, Y_l^d)\}_{l=1}^P \quad (3)$$

onde cada padrão é igual ao par (X_l^d, Y_l^d) , sendo X_l^d o vetor de entrada (estímulo de entrada) e Y_l^d o vetor desejado na saída. Para que a RNA seja treinada, um sinal é aplicado e uma saída é obtida; se essa saída não coincidir com a desejada existe um erro e , portanto, os pesos devem ser ajustados. O algoritmo *Backpropagation* trata o treinamento como um problema de otimização global sem restrições, onde a função erro quadrático, a ser minimizada, é definida por

$$E_p = \frac{1}{2} \sum_{i=1}^m (y_i^d - y_i)^2 \quad (4)$$

sendo m o número de saídas da RNA. Os exemplos são apresentados sequencialmente até que a diferença entre os valores gerados na(s) saída(s) da RNA e os desejados dos P exemplos seja menor ou igual a uma tolerância ϵ . O ajuste do peso j do neurônio i é definido por:

$$\Delta W_{ij} = -\eta \delta_i x_j \quad (5)$$

sendo

$$\delta_i = -(y_i^d - y_i) f'(v_i) \quad (6)$$

para a camada de saída e

$$\delta_i = f'(v_i) \sum_{j=1}^m \delta_j \frac{\partial v_j}{\partial y_i} \quad (7)$$

para as camadas ocultas. O coeficiente η , denominado taxa de aprendizagem, determina o comprimento do passo sobre a superfície da função erro quadrático.

3. Margem de Colapso de Tensão

Alguns indicadores podem apresentar o estado de um sistema elétrico de potência em relação a critérios econômicos ou de segurança. O custo operacional instantâneo das usinas geradoras e a perda total de potência ativa são indicadores para critérios econômicos. Para critérios de segurança, a margem de colapso de tensão pode ser utilizada para determinar o grau de segurança do caso base ou selecionar as piores contingências.

A margem de colapso de tensão (MCT) estabelece a quantidade de carga que pode ser adicionada do ponto de operação até o ponto de colapso. Para transformar a margem de colapso de tensão em um indicador de segurança, deve ser considerado a média das margens de todos os barramentos de carga ou a margem do barramento mais crítico do sistema. Para encontrar o barramento mais crítico, utiliza-se o método da Decomposição do Valor Singular (*Singular Value Decomposition*), descrito em [10].

A MCT é, normalmente, avaliada em função das variações das cargas de potência ativa e reativa [1], que proporcionam mudanças no perfil de tensão do sistema e na MCT. Neste trabalho, a margem de colapso de tensão é avaliada no barramento mais crítico em função das variações dos elementos de controle. Dessa forma, as ações de controle, causadoras de variações nos despachos de potência reativa, alteram o perfil de tensão do sistema e, por conseqüência, a MCT. Para calcular a MCT e formar o conjunto de exemplos de comportamentos, é utilizado o *Método da Curva Ajustada*, proposto por Ejebe *et al* [2].

Para que uma RNA possa apresentar em sua saída a margem de colapso de tensão do barramento mais crítico do sistema, um conjunto de padrões de treinamento é construído para ser apresentado na fase de treinamento. As entradas são os módulos das tensões dos barramentos de geração e a saída desejada é a margem de colapso de tensão do barramento.

4. Formulação do Problema

Neste trabalho, são fixados os valores das tensões dos barramentos de carga, pois, desta forma, os componentes do sistema trabalham melhor e o fluxo de potência reativa é limitado. Como foco principal, deseja-se preparar uma RNA para avaliar a segurança de um sistema de potência. Dessa forma, pode-se utilizá-la em modo *on-line*, na operação do sistema, e em modo *off-line*, na fase de planejamento – melhorando a segurança do sistema no processo de otimização. No último, tem-se um problema multiobjetivo (multicritério), pois é necessário minimizar o desvio entre a tensão fixada e a atual nos barramentos de carga e maximizar a MCT, respeitando os despachos de potência reativa dos geradores. Na *Programação Multiobjetiva Fuzzy* [11], todos os objetivos e restrições são tratados por graus de satisfação expressos por funções não lineares, chamadas de funções de pertinência. Para que seja encontrado um ponto que satisfaça todos os objetivos, cada um deles, representados pelas respectivas funções de pertinências, são ponderados conforme suas relevâncias.

A função objetiva a ser maximizada, denominada Conjunto *Fuzzy* de Decisão (CFD), resulta em:

$$\tilde{F}(u, x) = \tilde{W}_1 \mu_1(Y) + \sum_i \tilde{W}_i \mu_i(u_i) + \sum_j \tilde{W}_j \mu_j(x_j) \quad (8)$$

sendo u_i a i -ésima variável de controle, x_j a j -ésima variável de estado, Y a MCT fornecida pela RNA e \tilde{W}_k o peso *fuzzy* da função de pertinência $\mu_k(\cdot)$, que representa o grau de satisfação do respectivo objetivo k .

Nesta formulação, as variáveis de controle são as tensões nos barramentos de geração e os *taps* dos transformadores; as variáveis de estado são as tensões nos barramentos de carga e os despachos de potência reativa dos geradores. A função de pertinência μ_1 da equação (8), que expressa o grau de satisfação da MCT fornecida pela rede RNA, é descrita por:

$$\mu(\Delta Q) = \frac{1}{1 + e^{-k(\Delta Q - C)}} \quad (9)$$

sendo ΔQ a MCT atual, C o coeficiente que determina o deslocamento da curva $\mu(\Delta Q)$ em relação ao eixo vertical e k o coeficiente que determina a inclinação de $\mu(\Delta Q)$ no ponto C .

A cada iteração encontra-se a margem de colapso de tensão, avaliando a segurança do sistema e, conseqüentemente, a ação de controle que proporcional essa resposta.

A função de pertinência que descreve o grau de satisfação da i -ésima variável de controle é descrita por:

$$\mu_i = \begin{cases} 1 & \text{para } u_i^{\min} \leq u_i \leq u_i^{\max} \\ \exp\left(\frac{-(u_i - u_i^{\min})^2}{2\sigma^2}\right) & \text{para } u_i < u_i^{\min} \\ \exp\left(\frac{-(u_i - u_i^{\max})^2}{2\sigma^2}\right) & \text{para } u_i > u_i^{\max} \end{cases} \quad (10)$$

sendo u_i^{\min} e u_i^{\max} os valores mínimos e máximos permissíveis para a variável de controle u_i . O coeficiente σ^2 corresponde a variância da variável u_i .

A função de pertinência que descreve o grau de satisfação da j -ésima variável de estado é descrita por:

$$\mu_j = \begin{cases} 1 & \text{para } x_j = x_j^d \\ \exp\left(\frac{-(x_j - x_j^d)^2}{2\sigma^2}\right) & \text{demais casos} \end{cases} \quad (11)$$

sendo x_j^d o valor desejado.

A determinação dos pesos das funções de pertinências é realizada a cada iteração, seguindo a Tabela 1.

Tabela 1: Pesos lingüísticos para o módulo de tensão

Intervalo	Pesos lingüísticos
$\ x_i - x_i^d\ \leq 0.1$	“peso baixo”
$0.1 < \ x_i - x_i^d\ < 0.2$	“peso médio”
$\ x_i - x_i^d\ \geq 0.2$	“peso alto”

As mesmas regras da Tabela 1 são aplicadas para as variáveis de controle. Os conjuntos *fuzzy* que representam os pesos lingüísticos são do tipo trapezoidal, sendo suas relações entre as vértices dos trapézios apresentadas na Tabela 2.

Tabela 2: Vértices dos conjuntos *fuzzy*

Pesos lingüísticos	Vértices
“peso baixo”	[0.0, 0.03, 0.07, 1.0]
“peso médio”	[0.1, 0.23, 0.37, 0.5]
“peso alto”	[0.5, 0.63, 0.87, 1.0]

Quanto mais alto o grau de satisfação das funções de pertinência, maior o valor do conjunto $\tilde{F}(u, x)$.

4.1. Maximização do Conjunto Fuzzy de Decisão

A busca através do método do gradiente é adotada para maximizar o conjunto $\tilde{F}(u, x)$. O algoritmo, que converge quando todas violações são eliminadas, é implantado no pacote *Matlab versão 4.2c.1*. As funções de pertinência e suas variáveis independentes variam em relação às variáveis de controle. Portanto, o conjunto $\tilde{F}(u, x)$ varia diretamente em função das variáveis de controle, sendo o seu vetor gradiente definido por:

$$\nabla_u \tilde{F}(u, x) = \left[\frac{\partial \tilde{F}}{\partial u_1}, \dots, \frac{\partial \tilde{F}}{\partial u_p}, \frac{\partial \tilde{F}}{\partial u_{p+1}}, \dots, \frac{\partial \tilde{F}}{\partial u_n} \right]^T \quad (12)$$

onde n é o número de variáveis de controle. Como os pesos *fuzzy* não variam diretamente em função dos elementos de controle, mas de acordo com o conjunto de regras estabelecidas na Tabela 1, esses são considerados constantes em relação ao vetor controle. A variação do conjunto $\tilde{F}(u, x)$ em relação ao p -ésimo elemento de controle corresponde a derivada da

equação (8) em relação a u_p . Portanto, o p -ésimo elemento do vetor gradiente pode ser expresso como uma somatória de derivadas parciais definida por:

$$\frac{\partial \tilde{F}(u, x)}{\partial u_p} = \tilde{W}_1 \frac{\partial \mu_1(Y)}{\partial u_p} + \sum_i \tilde{W}_i \frac{\partial \mu_i(u_i)}{\partial u_p} + \sum_j \tilde{W}_j \frac{\partial \mu_j(x_j)}{\partial u_p} \quad (13)$$

Todas as derivadas parciais da equação (13) são demonstradas em [12], utilizando, quando necessário, a regra da cadeia. O principal detalhe que merece destaque é a derivada parcial $\partial \mu_i / \partial u_p$ que pode ser encontrada utilizando a regra da cadeia:

$$\frac{\partial \mu_1(Y)}{\partial u_p} = \frac{\partial \mu_1(Y)}{\partial Y} \frac{\partial Y}{\partial u_p} \quad (14)$$

Para que a equação (14) seja resolvida é necessário incorporar ao problema a estrutura interna da RNA, o que normalmente não é considerado, pois, após o treinamento, a rede é tratada como uma caixa preta que fornece resultados de uma generalização. A derivada $\partial \mu_i / \partial Y$ é direta, pois a função de pertinência μ_i varia em função de Y . No entanto, para encontrar a derivada $\partial Y / \partial u_p$ considera-se a modelagem matemática da estrutura de redes neurais *feedforward* apresentada no Item 2. Nesse caso, a variação instantânea da saída Y da RNA em relação ao elemento de controle u_p não é direta, pois $Y = f(v)$ como em (1). O desenvolvimento dessa derivada, para uma RNA de três camadas, é detalhado em [12], sendo o seu resultado final igual a

$$\frac{\partial Y}{\partial u_p} = f'(v) \sum_{i=1}^m w_i f'(v_i^{(1)}) w_{ip}^{(1)} \quad (15)$$

sendo W_i o i -ésimo peso sináptico do neurônio da saída Y e W_{ip} o p -ésimo peso sináptico do neurônio da saída $Y_i^{(1)} = f(v_i^{(1)})$. O sobrescrito 1 representa a camada em que está localizado o neurônio. Nesse exemplo, a camada 1 corresponde a primeira, e única, camada oculta.

O vetor gradiente da equação (12) é um conjunto *fuzzy*. A construção do vetor gradiente numérico utiliza o método *Centro de Área* [13] para *defuzzificação* de cada elemento i (para $i = 1, \dots, n$) do vetor gradiente de $\tilde{F}(u, x)$.

4.2. Ajuste das Variáveis de Controle

Os ajustes das variáveis de controle são calculados em função do coeficiente η que determina o comprimento do passo sobre a superfície de $\tilde{F}(u, x)$. Na iteração $t+1$, o ajuste da p -ésima variável de controle resulta em:

$$u_p^{(t+1)} = \begin{cases} u_p^{\min} & \text{para } u_p^{(t)} + \eta \nabla F_p^{(t)} < u_p^{\min} \\ u_p^{\max} & \text{para } u_p^{(t)} + \eta \nabla F_p^{(t)} > u_p^{\max} \\ u_p^{(t)} + \eta \nabla F_p^{(t)} & \text{para demais casos} \end{cases} \quad (16)$$

Neste trabalho, o coeficiente η é definido como a distância *Euclidiana* ou norma *Euclidiana* entre as tensões atuais e as fixadas. O coeficiente η , para a iteração t , é definido por:

$$\eta^{(t)} = \alpha \cdot \left\| X^{(t)} - X^d \right\| \quad (17)$$

sendo α ($0 \leq \alpha \leq 1$) a parcela da distância *Euclidiana* que é utilizada. O processo iterativo segue a linha de busca do gradiente, ajustando os elementos do vetor controle, até que todas as violações de tensão sejam eliminadas.

5. Estudo de Caso

O método que melhora a segurança de sistemas de potência, em relação a margem de colapso de tensão, é implantado pelos programas *E&PSG versão 4.01* [14] para execução do fluxo de carga, *MATLAB versão 4.2.c.1* (pacote matemático) e *RNA versão 2.2* [12] (implantação de redes neurais artificiais) em um processador AMD 586 133 MHz. Para análise dos resultados, o sistema IEEE-30 barras é utilizando sem nenhuma modificação em sua configuração básica. Os limites das tensões nos barramentos de carga são [0,975, 1.1] pu, com tolerância de 0.05 pu para o limite inferior nas contingências. O conjunto $\tilde{F}(u, x)$ possui, além da função de pertinência da RNA, 30 funções de pertinência das variáveis de estado e 10 funções de pertinência das variáveis de controle. A *Base de Dados* da RNA é construída considerando apenas o caso base, variando aleatoriamente as tensões dos barramentos de geração, ocasionando mudanças da MCT do barramento 30 – definido como o mais crítico pelo método apresentado em [10]. A RNA projetada para ser utilizada no conjunto $\tilde{F}(u, x)$ tem 6 entradas, 9 neurônios na camada oculta e 1 neurônio na saída. O algoritmo de treinamento convergiu com 277 épocas em 3 minutos, 12 segundos e 35 centésimos. O erro médio global foi de $4,54 \times 10^{-4}$ na fase de treinamento e $18,89 \times 10^{-4}$ na fase de teste.

Para que a RNA esteja pronta para ser adicionada ao conjunto $\tilde{F}(u, x)$ os seguintes passos são seguidos:

1. Execução do fluxo de carga (três por padrão);
2. Cálculo da margem de colapso de tensão;
3. Treinamento da RNA;
4. Teste da RNA.

A Tabela 3 apresenta o tempo gasto para cada passo.

Tabela 3: Tempo de preparação da RNA

PASSO	TEMPO (s)
1	117.30
2	0.35
3	192.35
4	< 0.01
Total	310.00

A *Base de Treinamento* utilizada possui um número relativamente baixo de padrões. Se for utilizada uma *Base de Treinamento* composta por 184 padrões (8 vezes maior) com a mesma topologia de rede, o tempo gasto para os passos 1, 2 e 4 aumentam linearmente. Para o passo 3, o tempo é imprevisível, pois a busca do erro médio global mínimo é estocástica. Estimando que o tempo gasto para o passo 3 seja 15 vezes maior, o tempo total para a preparação da RNA será de 3.826,45 segundos (aproximadamente 64 minutos). Esse tempo previsto é aceitável para a fase *off-line*.

Para uma análise sistemática, foram realizadas três simulações que partiram do mesmo ponto inicial e tiveram pesos diferentes para a função de pertinência da RNA. Com a variação desse parâmetro, pode-se verificar o efeito que a melhoria da MCT causa ao sistema, monitorando a perda total de potência ativa. A Tabela 4 apresenta os resultados do fluxo de carga para o estado inicial e depois do processo de otimização.

Tabela 4: Otimização do sistema IEEE-30

CONTROLE	INICIA L	W _{baixo}	W _{médio}	W _{alto}
Perdas (MW)	3.66	3.83	4.49	4.88
Perdas (MVA _r)	3.17	2.88	5.49	6.96
Reserva Mínima (MVA _r)	34.73 (43 %)	50.60 (39 %)	44.30 (34 %)	20.74 (16 %)
V _{min} (pu)*	0.7857	0.818 6	0.807 8	0.813 7
Violações*	19	5	5	5
Iterações	—	2	3	4

* Considerando todas as contingências simples

Considerando os indicadores da Tabela 4, a primeira simulação (W_{baixo}) obteve melhores resultados, por receber pouca influência da MCT que a rede fornece. A MCT do barramento 30 aumenta quando as ações de controle a ponderam com maior relevância; no entanto, outros indicadores, como a perda total de potência ativa e as reservas de potência reativa, tendem a piorar. Essa evidência é plenamente justificada pela necessidade de aumentar as gerações de potência reativa afim de aumentar a MCT.

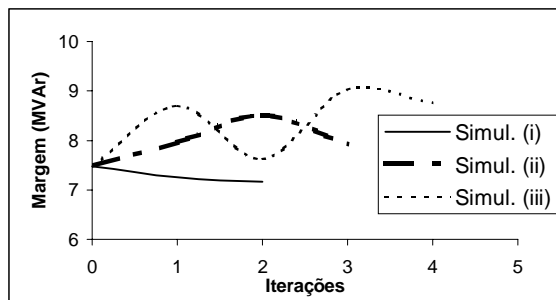


Figura 2: Margem de colapso de tensão por iterações

A Figura 2 apresenta a MCT do barramento 30 a cada iteração para os três casos. Na primeira simulação (W_{baixo}), a MCT diminui a cada iteração e chega a um resultado final inferior aos demais casos por receber baixa relevância no processo. Nas demais simulações ($W_{médio}$ e W_{alto}) a MCT oscila no decorrer do processo ao receber acentuada relevância, resultando em um conflito entre objetivos. Especificamente na terceira simulação, pode-se observar uma tendência em elevar a MCT devido ao acentuado peso lingüístico da RNA.

Com a ponderação por termos lingüísticos, os valores exatos da grandeza *peso* (W) são distribuídos entre três conjuntos *fuzzy*. Assim, diminui-se o espaço de estados e pode-se encontrar os melhores resultados em poucas simulações.

6. Conclusão

Este trabalho apresentou a tecnologia das redes neurais artificiais auxiliando a análise da segurança e a PMF na melhoria da MCT em sistemas de potência. A metodologia adotada tratou objetivos distintos por graus de pertinência, podendo, ao mesmo tempo, maximizá-los e minimizá-los. O pacote computacional implantado foi testado no sistema IEEE-30 barras, onde pôde ser encontrado, pela RNA, a MCT do barramento 30 a cada iteração do processo de otimização. Essa análise avalia o caminho seguido pelo vetor controle no sentido de melhorar a segurança do sistema (aumento da MCT). Foram realizadas três simulações que utilizavam diferentes pesos lingüísticos da RNA, criando-se um espaço de estados que pôde ser utilizado para avaliar quais foram as melhores ponderações e os efeitos causados pelo aumento da MCT no sistema. Os resultados são satisfatórios, pois essa metodologia permite determinar, em tempo real, um indicador de segurança do sistema, podendo ser estendida à análise de contingências.

Referências

[1] A. Tiranuchit and R. J. Thomas. A Posturing Strategy Against Voltage Instabilities in Electric Power Systems. IEEE Transactions on Power Systems, 3(1): 87-93, 1988.

[2] C. Ejebe, G. D. Irisarri, S. Mokhtari, O. Obadina, P. Ristanovic, J. Tong. Methods for Contingency Screening and Ranking for Voltage Stability Analysis of Power Systems. IEEE Transactions on Power Systems, 11(1), 1996.

[3] R. Atteri, M. A. El-Sharkawi. Static Security Assessment of Power System Using Kohonem Neural Network. IEEE Technology Update Series: Neural Networks Applications, pages 297-301, New York, 1996.

[4] H. Kanoh, K. Kanemaru, M. Kaneta and M. Nishiura. A Study on Pratical Fault Location for Power Transmission Lines Using Neural Networks. IEEE Technology Update Series: Neural Networks Applications, pages 245-249, New York, 1996.

[5] B. S. Kermanshahi, C. H. Poskar, G. Swift, W. Buhr and A. Silk. Artificial Neural Network for Forecasting Daily Loads of a Canadian Electric Utility. IEEE Technology Update Series: Neural Networks Applications, pages 268-273, New York City, 1996.

[6] L. A. Zadeh. Fuzzy Logic. IEEE Computer, pages 83-92, 1988.

[7] O. Alsac and B. Stott. Optimal Load Flow with Steady-State Security. IEEE Transactions on Power Apparatus and Systems, PAS-93: 745-751, 1974.

[8] F. Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65: 386-408, 1958.

[9] D. Rumelhart, G. Hinton and R. Williams. Learning Internal Representations by Error Propagation In D. E. Rumelhart and J. L. McClelland (Editors), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge (MA), 1986.

[10] P.-A. Löf, G. Anderson and D. J. Hill. Voltage Stability Indices for Stressed Power Systems. IEEE Transactions on Power Systems, 8(1): 326-335, 1993.

[11] S. U. Mohandas, T. A. Phelps and K. M. Ragsdell. Structural Optimization Using a *Fuzzy* Goal Programming Approach. Computer and Structures, 37(1): 1-8, 1990.

[12] M. M. Gouvêa Jr. Sistema Híbrido Neuro-Fuzzy Aplicado ao Problema do Fluxo Ótimo de Potência Reativa. Dissertação de Mestrado, Pontifícia Universidade Católica de Minas Gerais, Brasil, 1998.

[13] F. A. C. Gomide e R. R. Gudwin. Modelagem, Controle, Sistemas e Lógica Fuzzy. Controle & Automação, 4(3): 97-115, 1994.

[14] L. D. B. Terra and M. J. Short. Security-Constrained Reactive Power Dispatch. IEEE Transactions on Power Systems, 6(1): 109-117, 1991.