



**Universidade Federal de Campina Grande**  
**Centro de Engenharia Elétrica e Informática**  
**Departamento de Engenharia Elétrica**



**Trabalho de Conclusão de Curso**

**Estudo das características de curvas de resistência dinâmica de disjuntores utilizando  
redes neurais artificiais**

**Aluna: Taísa de Almeida Felix**

**Orientador: Edson Guedes da Costa**

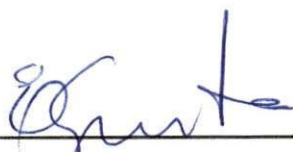
**Dezembro de 2009**

Aluna: TAÍSA DE ALMEIDA FELIX

## Trabalho de Conclusão de Curso

**BANCA EXAMINADORA:**

**Orientador:**



\_\_\_\_\_  
Professor Doutor Edson Guedes da Costa

**Professor Convidado:**



\_\_\_\_\_  
Professor Doutor Benedito Antonio Luciano

CAMPINA GRANDE - PARAÍBA

## **AGRADECIMENTOS**

A minha família, pelos ensinamentos que contribuíram para a formação do meu caráter.

Ao meu orientador, Edson Guedes da Costa pela orientação e incentivo que tornaram possível a conclusão desse trabalho.

A Vladimir Cesarino, pela participação valiosa na minha formação profissional.

A Tarso Vilela e André Germano, pelo apoio no amadurecimento dos meus conhecimentos e conceitos que me levaram a execução e conclusão deste trabalho.

Aos todos meus amigos e colegas, em especial Ana Maria, Cabral e Diego Márcio, pelo incentivo e apoio constantes.

Ao meu namorado e amigo João André, pelo apoio e compreensão da ausência em muitos momentos.

## SUMÁRIO

1.	Introdução.....	5
2.	Motivação .....	5
3.	Objetivos .....	6
4.	Redes Neurais Artificiais .....	7
4.1.	Um Breve Histórico .....	7
4.2.	Os Neurônios Biológico e Artificial .....	7
4.3.	Arranjo da RNA .....	11
4.4.	Algoritmo de Retropropagação do Erro .....	11
4.5.	Definição da Base de Dados.....	16
5.	Disjuntores .....	17
5.1.	Manutenção dos Disjuntores.....	17
5.2.	A Medição da Resistência Dinâmica dos Disjuntores .....	18
5.3.	Considerações sobre a MRD .....	19
5.4.	Estudo de métodos de interpolação para Construção de Curvas de Resistência Dinâmica .....	22
5.4.1.	Spline Cúbica.....	24
5.5.	Simulação de Curvas de Resistência Dinâmica .....	24
6.	Análise das curvas.....	26
7.	Conclusão.....	29
8.	Bibliografia .....	30
9.	Apêndice A - Construção das Curvas Sintéticas.....	31
10.	Apêndice B – Rede Neural Artificial.....	34

## **1. Introdução**

Disjuntores são equipamentos elétricos destinados à interrupção de correntes de falta tão rapidamente quanto possível, de forma a limitar ao máximo os danos causados aos equipamentos pelos curto-circuitos.

Eles estão entre os equipamentos de maior complexidade dentre aqueles em uso em sistemas de potência. Esta complexidade está presente tanto na concepção do disjuntor, seja na sua câmara de interrupção ou no seu sistema de acionamento (em suas múltiplas formas e princípios), quanto na sua aplicação na rede elétrica.

Os disjuntores devem ser mecanicamente capazes de atuar de forma eficaz mesmo após terem permanecido na posição fechada por vários meses. Essa exigência impõe cuidados especiais no projeto do equipamento, no sentido de reduzir ao máximo as massas das partes móveis e garantir a mobilidade das válvulas e o bom funcionamento das ligações mecânicas.

## **2. Motivação**

Muitos disjuntores são submetidos a manobras mais estressantes, como os disjuntores de bancos de capacitores e reatores, utilizados diariamente para regulação da tensão do sistema elétrico. Essas manobras provocam desgastes prematuros dos contatos fixos e móveis das câmaras de interrupção, que nem sempre são detectados através de ensaios convencionais, resultando em dados insuficientes para um diagnóstico preciso do estado real dos contatos das câmaras dos disjuntores, na dinâmica da extinção do arco elétrico.

Atualmente, um diagnóstico preciso só é possível por meio da desmontagem e inspeção interna das câmaras de interrupção, que é um processo caro e demorado. Se um componente interno do disjuntor puder ser inspecionado sem a necessidade de desmontagem, será possível acompanhar o disjuntor de maneira barata e eficiente, fazendo que o período de manutenção seja otimizado.

### **3. Objetivos**

O principal objetivo deste trabalho foi estudar curvas de resistência dinâmica de disjuntores de alta tensão e tratá-las por meio de algoritmos baseados em redes neurais artificiais (RNA) para diagnosticar o estado dos contatos no interior das suas câmaras.

## 4. Redes Neurais Artificiais

### 4.1. Um Breve Histórico

Diante da capacidade de organização natural de um cérebro, os cientistas do século passado tentaram reproduzir uma versão artificial dessa capacidade e para isso, desenvolveram versões artificiais diferentes para o neurônio biológico.

Por volta de 1943, McCulloch e Pitts publicaram artigos que sugeriam a construção de uma máquina baseada no cérebro humano. Surgiram, então, muitos livros e artigos tratando sobre esse assunto. Destaca-se o livro *The Organization of Behavior* de Donald Hebb, publicado em 1949.

Em 1951, Marvin Minsky construiu o primeiro neuro-computador chamado *Snark*. Em 1956, no *Darhmouth College*, surgem dois paradigmas da Inteligência Artificial (IA):

- Paradigma simbólico: buscava imitar o comportamento do cérebro humano, desconsiderando os mecanismos que o fazem;
- Paradigma conexionista: acreditava que se existisse um sistema artificial que simulasse a estrutura do cérebro, esse sistema possuiria inteligência, ou seja, seria capaz de aprender, assimilar, errar e aprender com seus erros.

Em 1957 e 1958, o neuro-computador *Mark I Perceptron*, criado por Frank Rosenblatt, Charles Wightman e outros, fez enorme sucesso. Após a criação do Perceptron, Rosenblatt, Bernard Widrow e alguns estudantes criaram um novo tipo de RNA, a qual foi denominada de *Adaline*. Essa RNA tem um processo eficiente de aprendizado e até hoje é usada.

### 4.2. Os Neurônios Biológico e Artificial

O cérebro humano funciona da seguinte forma esquemática:

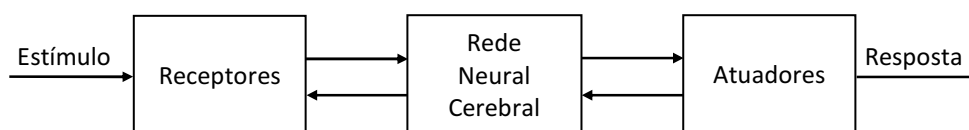


Figura 1 – Representação do funcionamento do cérebro humano

Os receptores convertem o estímulo em pulsos elétricos a serem interpretados pela rede neural cerebral. As setas para a direita e esquerda indicam transmissão de dados (informações). As informações são analisadas e, em seguida, é dada a resposta do estímulo por meio de atuadores que convertem os impulsos elétricos gerados pela rede neural em respostas discerníveis como saídas.

A partir do corpo celular do neurônio biológico, partem o axônio, que se comporta como uma linha de transmissão de impulsos entre neurônios, e os filamentos (ou dendritos), que são as zonas de recepção de estímulos. O axônio é único em uma célula nervosa podendo possuir ramificações. Na figura 2 apresenta-se o formato de um neurônio biológico.

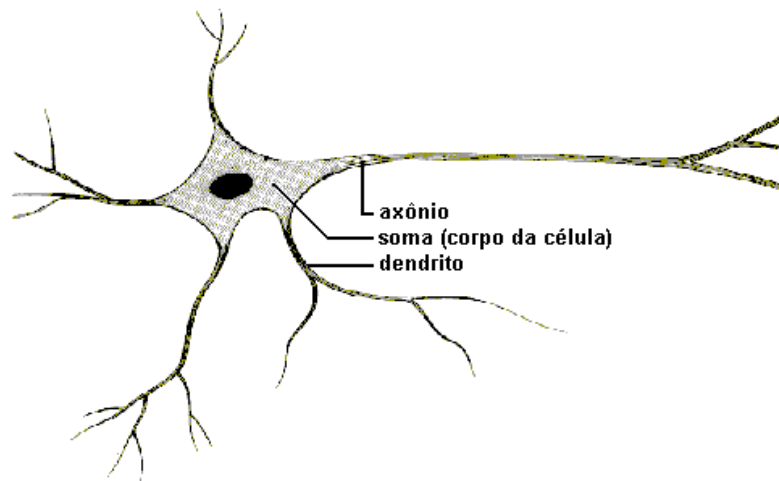


Figura 2 – Aspecto de um neurônio biológico

Os dendritos recebem o estímulo do meio ou das conexões com os outros axônios das outras células nervosas. Esse estímulo é realizado, no caso de ligações entre axônios e dendritos, sob a forma de uma série de pulsos rápidos de tensão, conhecidos como potenciais de ativação e originados no corpo celular do neurônio. O pulso elétrico que sai do neurônio transmite o que se chama de *saída* ou *resposta* da célula.

Um conjunto de neurônios interconectados é chamado de rede neural. O exemplo mais trivial é a *rede neural biológica*. Uma das características do nosso cérebro é a capacidade de se adaptar ao meio, fazendo modificações nas ligações sinápticas.



Imitar as funções e propriedades inerentes do cérebro humano foi o que impulsionou o surgimento das *redes neurais artificiais*, que se assemelharam a rede neural biológica nos seguintes aspectos:

- A utilização da mesma arquitetura lógica;
- O conhecimento é adquirido em um processo de aprendizado;
- As conexões inter-neurônios, conhecidas como pesos sinápticos, são utilizadas para guardar o conhecimento da rede.

Em um neurônio artificial:

- Os dendritos são representados pelas entradas de cada neurônio com o seu respectivo peso sináptico;
- O corpo celular, é um combinador linear, para somar os sinais de entrada e logo depois aplicar a função de ativação, que tem sua saída normalizada geralmente em  $[1,1]$  ou  $[-1,1]$ ;
- O axônio é representado pela saída ou resposta da rede.

O modelo do neurônio também inclui um termo denominado de *polarização* ou *bias*, cujo efeito é elevar ou reduzir a entrada da função de ativação, por meio da aplicação de um sinal de entrada com valor fixo.

Na figura 3 apresenta-se, de forma esquemática, um neurônio artificial [Haykin, 2001].

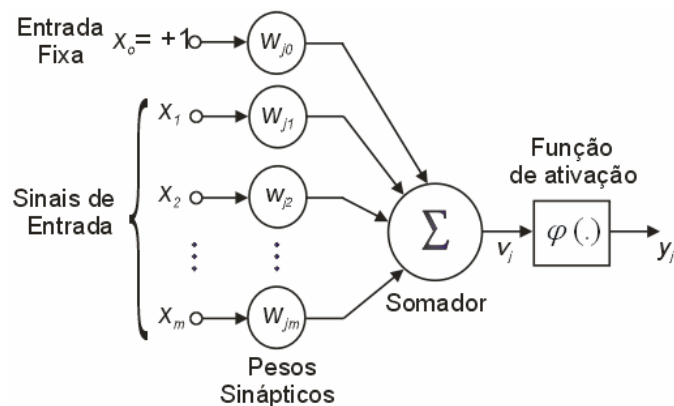


Figura 3 – Aspecto de um neurônio artificial [Haykin, 2001]

Do ponto de vista matemático, tem-se:

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k$$

$$y_k = \phi(v_k)$$

Sendo:

$x_0, x_1, \dots, x_m$  : sinais de entrada;

$w_{k0}, w_{k1}, \dots, w_{km}$  : pesos sinápticos do neurônio  $k$ ;

$v_k$  : potencial de ativação do neurônio  $k$ ;

$\phi(v_k)$  : função de ativação do neurônio  $k$ ;

$y_k$  : sinal de saída do neurônio  $k$ ;

$b_k$  : bias ou polarização.

Para normalizar a saída de cada neurônio, aplica-se uma função de ativação em cada saída  $v_k$  do neurônio.

A função de ativação utilizada neste trabalho foi a função sigmóide:

$$\phi(v_k) = \frac{1}{1 + e^{-av_k}}$$

O parâmetro  $a$  corresponde à suavidade da função sigmóide. Seu valor é sempre maior que zero (vide figura 4).

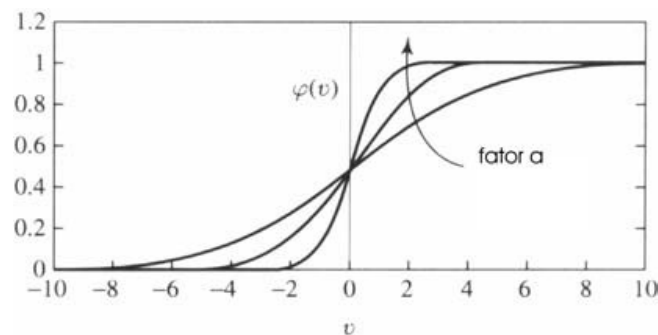
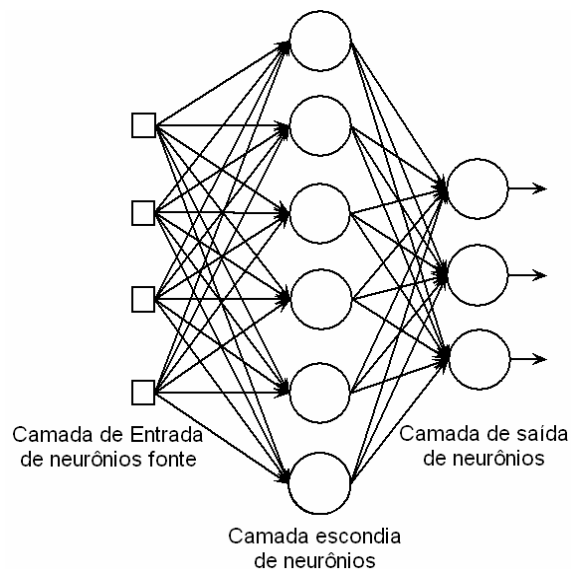


Figura 4 - Função sigmóide

### 4.3. Arranjo da RNA

Para uma aplicação mais complexa, vários neurônios artificiais devem ser associados, formando uma RNA. O arranjo desses neurônios define os tipos de topologias de RNAs encontradas na literatura, entre as quais a topologia utilizada neste projeto é apresentada a seguir.

*Rede de múltiplas camadas:* apresenta camadas internas (ocultas), além da camada sensorial e da camada de saída. Os sinais de entrada se propagam na rede até chegarem à camada de saída. Esta RNA também é chamada de rede perceptron multicamada (*multilayer perceptron*) ou rede MLP.



**Figura 5 - Rede Perceptron multicamada**

Para esse tipo de rede há diversos tipos de métodos de treinamento. Neste trabalho foi estudado e implementado o algoritmo de retropropagação do erro (também conhecido como *backpropagation*).

### 4.4. Algoritmo de Retropropagação do Erro

Nesse método de treinamento, um padrão de entrada apresentado à rede na iteração  $n$  é propagado camada a camada. O valor da saída de um neurônio  $j$  da camada  $k$  é calculado como:

$$y_j^k(n) = \varphi_j^k(v_j^k(n)), \quad (1)$$

sendo  $\varphi_j^k(\cdot)$  a função de ativação e  $v_j^k(n)$  o potencial de ativação do neurônio  $j$  da camada  $k$  na iteração  $n$ .

O potencial de ativação do neurônio  $v_j^k(n)$ , na iteração  $n$ , é calculado como o somatório dos sinais de entrada do neurônio ponderados pelos pesos das respectivas ligações sinápticas, ou seja:

$$v_j^k(n) = b_j^k(n) + \sum_{i=1}^C w_{ji}^k(n) \cdot y_i^{k-1}(n), \quad (2)$$

$C$  é o número de neurônios da camada  $k-1$ ;

$w_{ji}^k(n)$  o peso sináptico que liga o neurônio  $i$  da camada  $k-1$  ao neurônio  $j$  da camada  $k$ ;

$y_i^{k-1}(n)$  a saída do neurônio  $i$  da camada  $k-1$ ;

$b_j^k(n)$  a polarização do neurônio  $j$  da camada  $k$ , cujo sinal de entrada é fixo em  $+1$ .

Após a fase de propagação, um sinal de erro é obtido a partir da comparação entre a saída obtida pela rede e a saída desejada. O sinal de erro  $e_j^z(n)$  do neurônio  $j$  da camada de saída, denominada de camada  $z$ , na iteração  $n$  é definido como:

$$e_j^z(n) = d_j^z(n) - y_j^z(n). \quad (3)$$

$d_j^z(n)$  e  $y_j^z(n)$  são, respectivamente, o valor de saída desejado e o valor de saída obtido do para o neurônio  $j$  da camada de saída  $z$  na iteração  $n$ .

Para cada padrão de treinamento, calcula-se a *função energia de erro* como sendo:

$$\xi(n) = \frac{1}{2} \sum_{j \in Z} [e_j^z(n)]^2, \quad (4)$$

com  $Z$  sendo o conjunto de todos os neurônios da camada de saída da RNA.

Para cada padrão de entrada apresentado à rede é necessário se determinar a correção  $\Delta w_{ji}^k(n)$  aplicada ao peso  $w_{ji}^k(n)$ , que de acordo com o método do gradiente descendente é definida por:

$$\Delta w_{ji}^k(n) = -\eta \frac{\partial \xi(n)}{\partial w_{ji}^k(n)}, \quad (5)$$

sendo  $\eta$  o coeficiente de aprendizagem da RNA, podendo assumir valores no intervalo fechado  $[0,1]$ .

A derivada parcial  $\frac{\partial \xi(n)}{\partial w_{ji}^k(n)}$  representa um fator de sensibilidade, determinando a escolha da direção que a RNA deve tomar na superfície de erro gerada no espaço de todos os pesos  $w_{ji}^k$  da rede, com  $k$  indexando as camadas da rede.

Aplicando a regra da cadeia ao fator de sensibilidade, obtêm-se:

$$\frac{\partial \xi(n)}{\partial w_{ji}^k(n)} = \frac{\partial \xi(n)}{\partial e_j^k(n)} \frac{\partial e_j^k(n)}{\partial y_j^k(n)} \frac{\partial y_j^k(n)}{\partial v_j^k(n)} \frac{\partial v_j^k(n)}{\partial w_{ji}^k(n)}. \quad (6)$$

A partir das equações (2) e (7), pode-se escrever a correção do peso  $w_{ji}^k(n)$  na iteração  $n$  como:

$$\Delta w_{ji}^k(n) = \eta \delta_j^k(n) y_i^{k-1}(n), \quad (7)$$

onde o termo  $\delta_j^k(n)$  é denominado de gradiente local e é expresso como:

$$\delta_j^k(n) = -\frac{\partial \xi(n)}{\partial v_j^k(n)} \quad (8)$$

Analisando-se as equações (7) e (9), observa-se que elas não podem ser calculadas diretamente para um neurônio  $j$  pertencente a uma camada oculta  $k$ , já que não há informação direta do sinal de erro  $e_j^k(n)$ . Isto só é possível para os neurônios da camada de saída  $z$ , já que há informação dos sinais de saída desejados a cada iteração  $n$ .

Dessa forma, torna-se evidente a consideração de dois casos para o cálculo do gradiente local de um neurônio.

No primeiro caso, o neurônio está na camada de saída. Nessa situação é possível saber exatamente a saída do neurônio. A partir das equações (1), (3) e (4), pode-se escrever a equação (9) como:

$$\begin{aligned}
 \delta_j^z(n) &= -\frac{\partial \xi(n)}{\partial v_j^z(n)} \\
 &= -\frac{\partial \xi(n)}{\partial e_j^z(n)} \frac{\partial e_j^z(n)}{\partial y_j^z(n)} \frac{\partial y_j^z(n)}{\partial v_j^z(n)} \quad (9) \\
 &= -[e_j^z(n)] \cdot (-1) \cdot [\varphi_j^{z'}(v_j^z(n))] \\
 &= \varphi_j^{z'}(v_j^z(n)) \cdot e_j^z(n)
 \end{aligned}$$

O segundo caso é o neurônio em uma camada interna ou oculta. Nesse caso, não é possível conhecer o sinal de erro  $e_j^k(n)$ . Na verdade, esses erros são propagados desde os neurônios da camada de saída até os demais localizados nas camadas ocultas. Dessa forma, para o cálculo do sinal de erro de um neurônio oculto em uma camada  $k$ , leva-se em consideração todos os sinais de erro dos neurônios da camada  $k+1$ , que, por sua vez, leva em consideração todos os sinais de erro da camada  $k+2$ , assim sucessivamente até a camada de saída  $z$ , onde se dispõe efetivamente dos sinais de erro  $e_j^z(n)$ . Por simplicidade, considera-se o índice da camada  $k+1=z$  como sendo a camada  $z$ , ou seja, a camada oculta  $k$  é imediatamente anterior a camada de saída. Com isso, a partir das equações (2), (3) e (4), o gradiente local  $\delta_j^k(n)$  pode ser determinado por:

$$\begin{aligned}
\delta_j^k(n) &= -\frac{\partial \xi(n)}{\partial v_j^k(n)} \\
&= -\frac{\partial \xi(n)}{\partial y_j^k(n)} \frac{\partial y_j^k(n)}{\partial v_j^k(n)} \\
&= -\left[ \sum_{l=1}^Z e_l^z(n) \frac{\partial e_l^z(n)}{\partial y_j^k(n)} \right] \cdot \varphi_j^{k'}(v_j^k(n)) \\
&= -\left[ \sum_{l=1}^Z e_l^z(n) \frac{\partial e_l^z(n)}{\partial v_l^z(n)} \frac{\partial v_l^z(n)}{\partial y_j^k(n)} \right] \cdot \varphi_j^{k'}(v_j^k(n)) \\
&= -\left\{ \sum_{l=1}^Z e_l^z(n) \cdot [-\varphi_l^{z'}(v_l^z(n))] \cdot [w_{lj}^z(n)] \right\} \cdot \varphi_j^{k'}(v_j^k(n)) \\
&= \varphi_j^{k'}(v_j^k(n)) \sum_{l=1}^Z \delta_l^z(n) w_{lj}^z(n)
\end{aligned} \tag{10}$$

De forma geral, o cálculo do gradiente local de um neurônio  $j$  localizado em uma camada oculta  $k$  é dado por:

$$\delta_j^k(n) = \varphi_j^{k'}(v_j^k(n)) \cdot \sum_{l=1}^E \delta_l^{k+1}(n) w_{lj}^{k+1}(n) \tag{11}$$

onde  $E$  representa o número total de neurônios da camada  $k+1$ .

A partir das equações (10) e (11), o gradiente local  $\delta_j^k(n)$  de um neurônio é calculado como:

$$\delta_j^k(n) = \begin{cases} \varphi_j'(v_j^k(n)) \cdot e_j^k, & \text{para } k = z \text{ (camada de saída)} \\ \varphi_j'(v_j^k(n)) \cdot \sum_{l \in D} \delta_l^{k+1}(n) w_{lj}^{k+1}(n), & \text{se } k \text{ é a uma camada escondida.} \end{cases} \tag{12}$$

Observa-se da equação (12) que o cálculo do gradiente local  $\delta_j^k(n)$  para cada neurônio de um perceptron de múltiplas camadas requer o conhecimento da derivada da função de ativação  $\varphi_j^k$  associada ao neurônio. Para essa derivada existir, é necessário que a função de ativação seja contínua.

O algoritmo de retropropagação proporciona uma aproximação da trajetória a ser tomada pela RNA na superfície de erro gerada no espaço dos pesos. Quanto menor

for o valor do coeficiente de aprendizagem  $\eta$ , menores serão os incrementos nos pesos das ligações da RNA de uma interação para outra, e mais suave será a trajetória. Isso provoca uma convergência lenta no treinamento da RNA.

O valor inicial dos pesos pode dificultar ainda mais a convergência no treinamento, pois os valores iniciais podem corresponder a um *vale*, que é uma região de difícil acesso na superfície de erro, dificultando o caminho para o mínimo global da função.

#### 4.5. Definição da Base de Dados

A Base de Dados é o conjunto de padrões que vai ser utilizada para o treinamento da RNA. Em geral, uma Base de Dados pode ser estruturada conforme apresentado na figura 6. A base de dados deste trabalho consistiu em 70% dos dados para avaliação, 15% para validação do algoritmo desenvolvido e 15% para os testes.

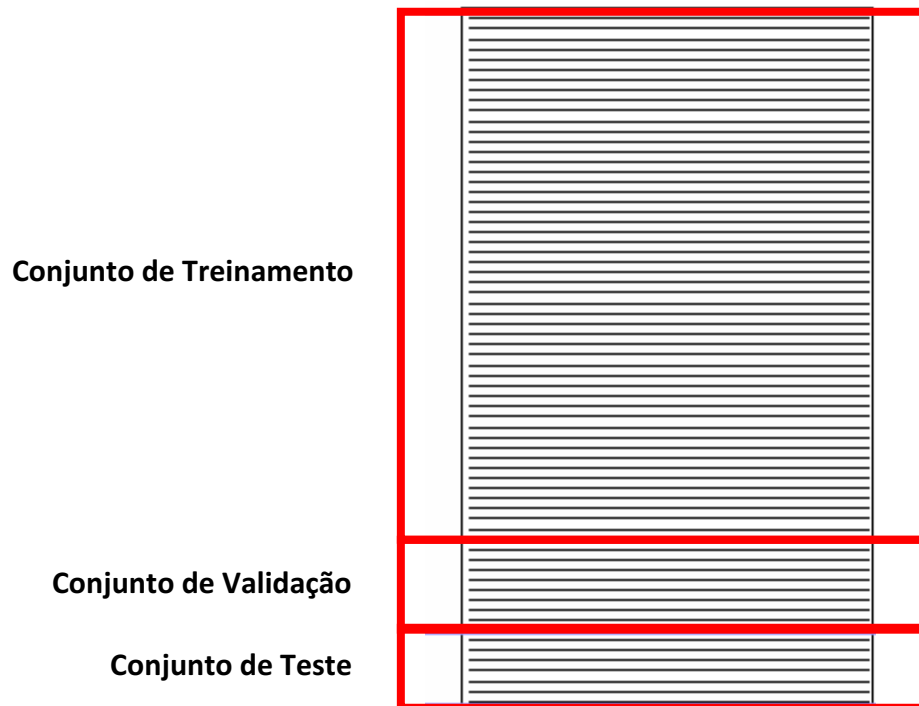


Figura 6 – Base de dados da RNA



## 5. Disjuntores

As funções mais frequentemente desempenhadas pelos disjuntores são, em primeiro lugar, a condução de correntes de carga na posição fechada, seguindo-se o isolamento entre duas partes de um sistema elétrico. Os disjuntores são, em geral, chamados a mudar de uma condição para outra ocasionalmente, e a desempenhar a função de abrir faltas ou fechar circuitos sob falta apenas muito raramente.

Os disjuntores devem ser mecanicamente capazes de abrir em tempos tão curtos quanto dois ciclos, após terem permanecido na posição fechada por vários meses. Essa exigência impõe cuidados especiais no projeto do equipamento, no sentido de reduzir as massas das partes móveis e de garantir a mobilidade das válvulas e ligações mecânicas.

### 5.1. Manutenção dos Disjuntores

Manutenção é um conjunto de ações realizadas direta ou indiretamente em um sistema ou equipamento, com a finalidade de verificar, manter ou restabelecer as condições de cumprir com segurança e eficiência suas funções [Furnas, 1995].

A função da manutenção nos equipamentos de manobra, como em qualquer outro equipamento, é assegurar-lhes um elevado grau de disponibilidade quando eles estão desempenhando suas funções nos sistemas elétricos, para permitir o retorno dos investimentos realizados com estes equipamentos. Outras vantagens esperadas da manutenção são o aumento do tempo de vida, a redução do número de falhas ou defeitos e a possibilidade de se detectar, previamente anormalidades nos equipamentos.

Os fabricantes tendem a ser cautelosos na especificação da manutenção e suas recomendações são caracterizadas por uma alta frequência de desmontagem e substituição de componentes. A periodicidade e os procedimentos de manutenção apresentados pelos fabricantes de disjuntores levam em consideração:

- a soma das interrupções de correntes de curto-circuito ou em carga nominal;
- o número total de ciclos fechamento-abertura;

- o tempo de serviço.

As recomendações dos fabricantes apresentam algumas desvantagens do ponto de vista do usuário, uma vez que a alta frequência de intervenções e o controle implicam:

- maior indisponibilidade do disjuntor para o sistema, causada por suas retiradas de serviço;
- introdução de defeitos em consequência do excesso de desmontagens e remontagens;
- necessidade de grandes quantidades de peças sobressalentes.

O controle do estado das conexões do disjuntor de uma subestação é feito por meio da medição da resistência dos contatos. Nas conexões onde se configure uma má conexão, haverá um aumento gradual da resistência de contato, e conseqüentemente, um sobreaquecimento. Geralmente uma conexão elétrica defeituosa não produz uma falha imediata, porém permite um deterioração gradativa pela ação do ambiente e do próprio sobreaquecimento. Por sua vez, o aquecimento se torna progressivo, fazendo da conexão uma fonte de calor.

A medição da resistência para conhecer o estado das conexões é feita enquanto o disjuntor está fechado, e não fornece nenhuma indicação da condição dos contatos de arco. Para avaliar a condição dos contatos de arco do disjuntor, é necessário fazer uma inspeção interna, o que constitui um processo caro e demorado. É nesse contexto que a idéia da medição da resistência dinâmica do disjuntor enquadra-se. A partir de uma rápida medição torna-se possível detectar possíveis falhas no disjuntor e, dessa forma otimizar o período da manutenção, tornando-a mais barata e eficiente.

## **5.2. A Medição da Resistência Dinâmica dos Disjuntores**

As primeiras descrições e comentários sobre Medição da Resistência Dinâmica de disjuntores, MDR, foram feitas no fim da década de oitenta. O princípio de medição é muito simples. Basicamente, trata-se de um teste de resistência de contato, mas em

vez de tomar um único valor, quando os contatos do disjuntor estão fechados (valor estático), a resistência é medida durante uma operação do disjuntor.

A medição pode ser descrita nos seguintes passos:

- Injeção de uma alta corrente (pelo menos 100 A), com o objetivo de minimizar o nível do ruído;
- Medir a queda de tensão e a corrente nos contatos do disjuntor durante a operação de abertura;
- Calcular a resistência em função da variável de interesse (usualmente o tempo ou a distância entre os contatos do disjuntor).

Uma descrição básica de um conjunto de medição é dada na figura 7.

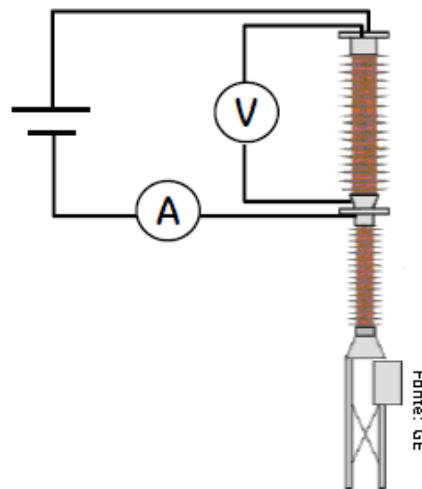


Figura 7 - Esquema para a medição da resistência dinâmica

### 5.3. Considerações sobre a MRD

#### – Medições durante operação de abertura

Não é recomendável que em uma MRD seja feita em operações de fechamento do disjuntor, por dois motivos principais:

- A transição brusca entre a resistência que deixa de ser infinita (contatos abertos) para se tornar a resistência de contato fixo é difícil de ser medida;

- O valor da corrente contínua existente no momento em que os contatos de arco entram em contato gera um ruído indesejável, que compromete a medição.

#### – Medições em baixa velocidade

A literatura recomenda que tais medições sejam realizadas em operações de abertura e em baixa velocidade (de 0,002 a 0,2 m/s), pois, dessa forma a medição possui um comportamento mais estável.

Segundo o que foi constatado pelas medições de Laudry et al, 2006, na velocidade normal de abertura, a curva de resistência não obedece a padrões. Isso é facilmente visualizado comparando-se o gráfico da figura 8 com o gráfico da figura 9, que exibe as curvas de resistência dinâmica para medições realizadas em baixa velocidade.

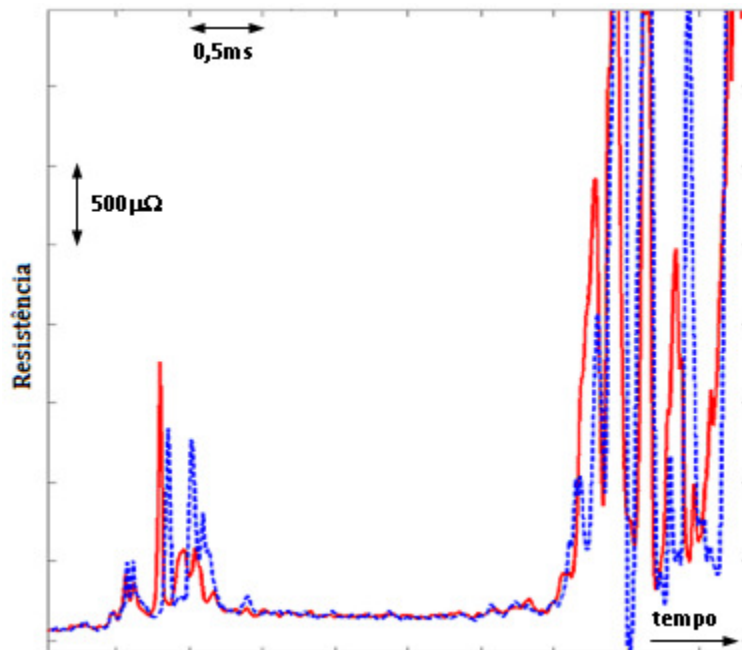


Figura 8 – MRD em velocidade normal

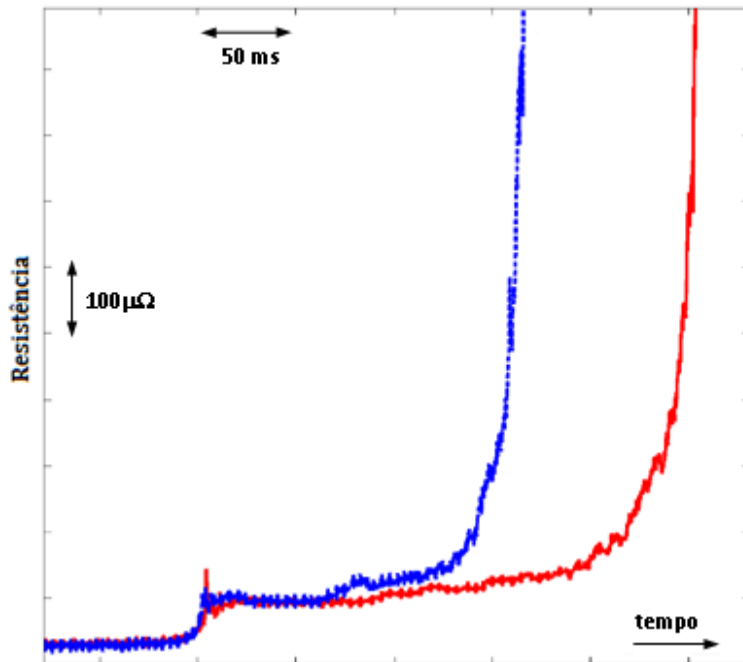


Figura 9 – MRD em baixa velocidade

– **Gráfico da curva de resistência dinâmica em função da distância entre os contatos.**

Para compensar o fato de a curva de resistência dinâmica ser medida em baixa velocidade (que não é necessariamente constante para todos os testes), é recomendável que o gráfico seja feito em função da distância entre os contatos. Outra vantagem da utilização da distância entre os contatos como um parâmetro da curva é que, dessa forma, são obtidas as informações sobre a situação do alinhamento entre os contatos.

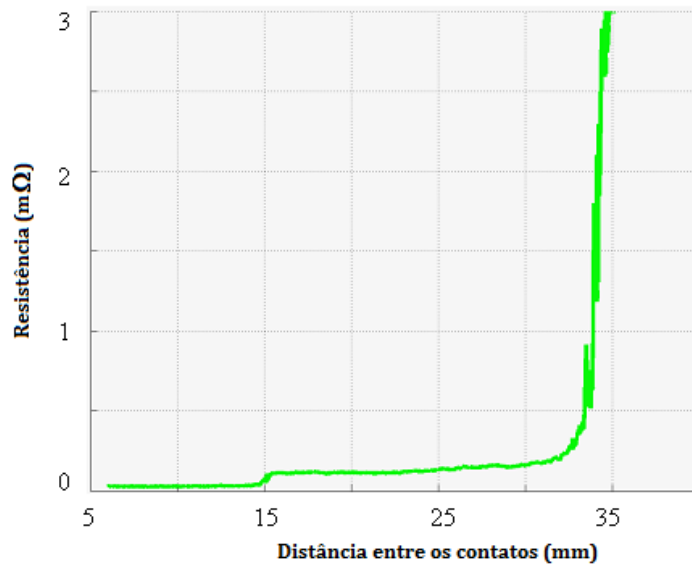


Figura 10 – MRD em função da distância entre os contatos do disjuntor

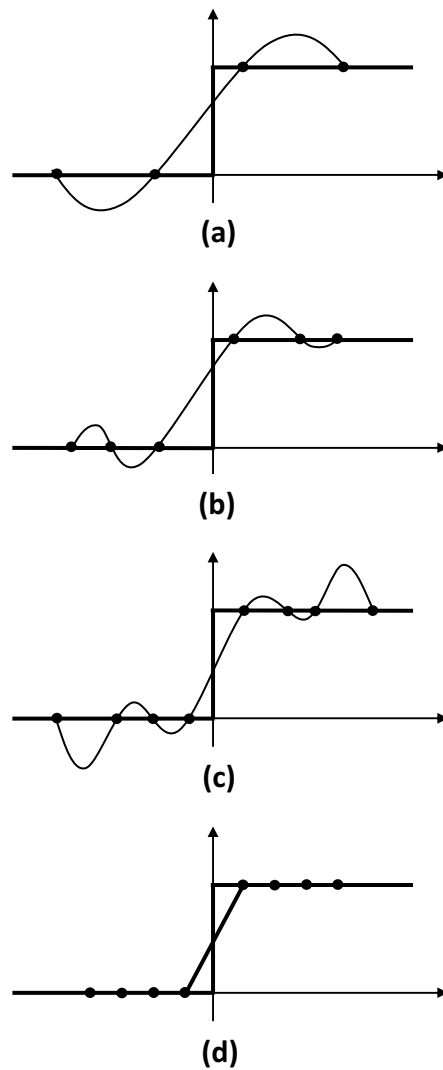
#### 5.4. Estudo de métodos de interpolação para Construção de Curvas de Resistência Dinâmica

A falta de dados experimentais que ilustrassem o comportamento da resistência dinâmica dos disjuntores tornou necessário o desenvolvimento de um algoritmo que representasse o comportamento típico da curva.

Baseado nas curvas de resistência dinâmica obtidas por Laudry et al, 2006, foram coletados dados que deram suporte aos estudos desenvolvidos. Os dados foram interpolados para que fossem construídas curvas representativas de disjuntores considerados bons e de disjuntores considerados defeituosos.

A maioria dos métodos de interpolação conecta os pontos conhecidos e estabelece um polinômio de ordem  $n$  que interpola os pontos e estima os valores intermediários das curvas.

É sabido que a resistência de um disjuntor varia rapidamente de um valor próximo a zero (disjuntor fechado) para um valor muito alto (disjuntor aberto). Esse tipo de variação brusca faz com que uma curva definida por um único polinômio oscile muito, como pode ser visualizado na figura 11a, 11b e 11c.



**Figura 11 - Construção de uma função degrau utilizando diferentes métodos de interpolação**

Na figura 11d, o método de interpolação utilizado é denominado Spline cúbica. Como pode ser observado, ele se adequa bem em caso de mudanças abruptas. Essa característica fez que esse método fosse escolhido para simular as curvas de resistência dinâmica utilizadas neste trabalho.

#### **5.4.1. Spline Cúbica**

Uma spline cúbica é uma curva suave, construída por meio da conexão de determinado conjunto de pontos. A idéia principal desse método é, em vez de aproximar uma dada função  $f(x)$  em um intervalo  $[x_0, x_n]$  por um único polinômio, dividir  $[x_0, x_n]$  em "n-1" subintervalos  $(x_0, x_1)$ ,  $(x_1, x_2)$ ,  $(x_{n-1}, x_n)$  e aproximar  $f(x)$  por um polinômio diferente em cada subintervalo.

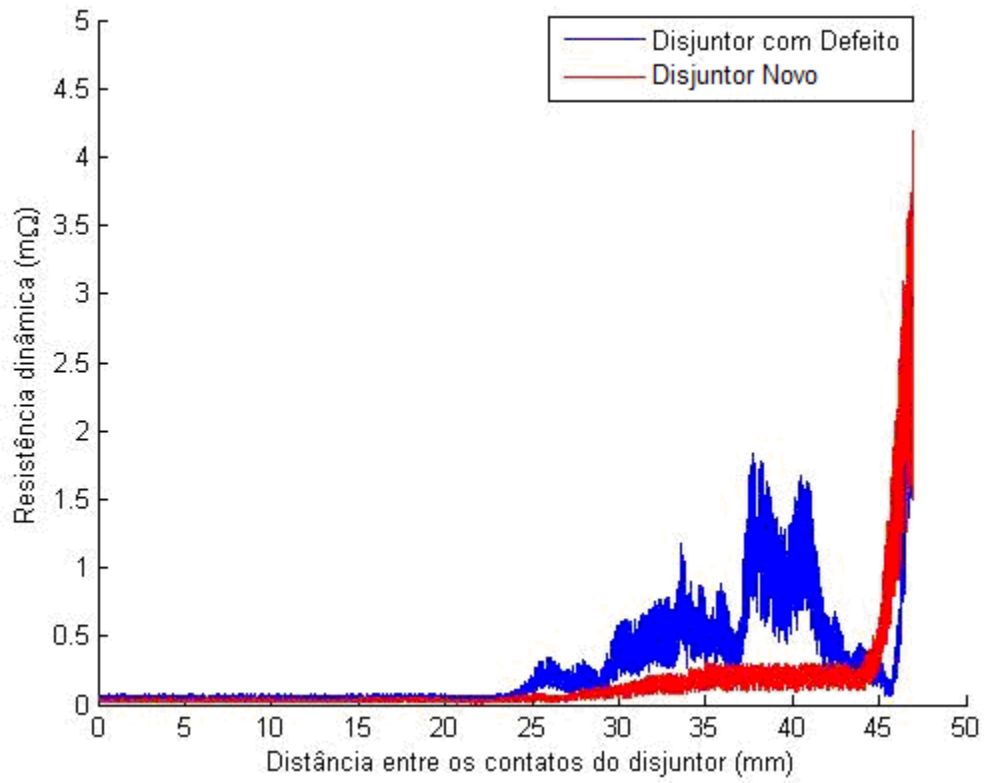
As splines são constituídas por polinômios de baixo grau, por isso são mais úteis para o preenchimento de curvas, em razão da redução do tempo de processamento dos cálculos, que outros métodos que utilizam polinômios de alto grau, os quais geram instabilidade numérica.

#### **5.5. Simulação de Curvas de Resistência Dinâmica**

O algoritmo de simulação de curvas de resistência dinâmica de disjuntores foi implementado no Matlab®, tomando como base o método de interpolação spline cúbica. No gráfico da figura 12, observa-se exibida uma superposição de resultados de 100 curvas de resistência dinâmica (50 de disjuntores bons e 50 de disjuntores com defeito).

O ruído inserido nas curvas da figura 12 foi um valor variável entre 0 e 50% do valor absoluto da resistência em cada ponto do gráfico. Ainda assim é possível observar que há um padrão de comportamento que depende do estado dos contatos do disjuntor.





**Figura 12 – Curvas de resistência dinâmica superpostas**

## 6. Análise das curvas

Após a construção de curvas que simulam comportamentos de disjuntores bons e disjuntores com defeito, foi possível implementar a RNA que analisa o estado dos contatos do disjuntor, baseado na curva de resistência dinâmica.

A entrada da RNA desenvolvida é um vetor contendo 05 valores de resistência de cada curva amostrada. O código foi implementado de forma que metade das amostras enviadas para a RNA correspondessem a disjuntores bons, e a outra metade, a disjuntores defeituosos. O conjunto de amostras foi dividido da seguinte forma: 70% para treinamento, 15% para validação, 15% para testes da RNA.

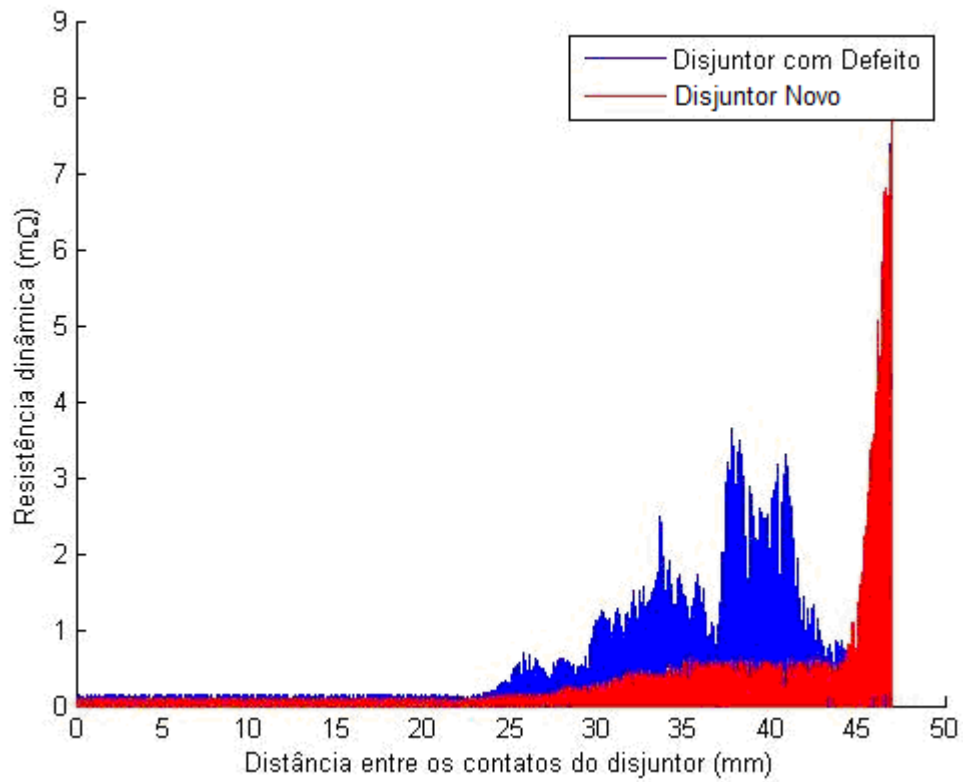
O algoritmo foi testado com diferentes níveis de ruídos. Após a realização de diversos testes, a RNA utilizada para o estudo da resistência dinâmica de disjuntores apresentou as seguintes características:

- 15 neurônios na camada de entrada;
- 2 neurônios na camada de saída;
- Função de ativação sigmóide;
- Algoritmo de treinamento Retropropagação (*Backpropagation*);
- 500 épocas;
- 500 amostras de curvas (250 de disjuntores bons e 250 de disjuntores ruins).

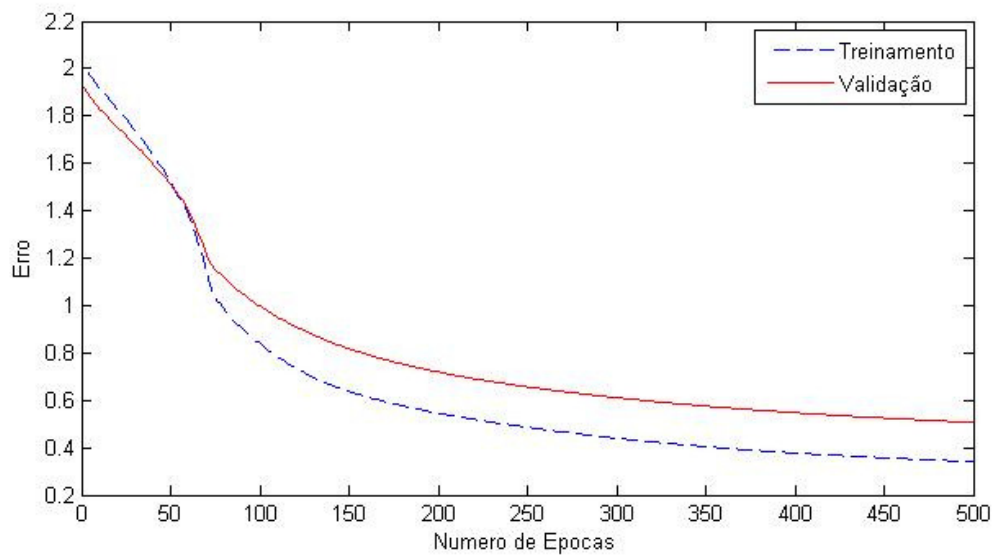
O objetivo do código é treinar a rede, de forma que ela fique apta a reconhecer os padrões de disjuntores bons e os padrões de disjuntores defeituosos.

Após a execução do código, apresenta-se o acerto percentual da RNA e a curva de convergência do método que mostra a quantidade de erros em função do número de épocas durante as etapas de treinamento e validação da RNA.

Por exemplo, para um ruído de 200%, o padrão de curvas inserido na RNA é o da figura 13, a taxa de acerto é de 80% e o gráfico com a convergência do método é o da figura 14.

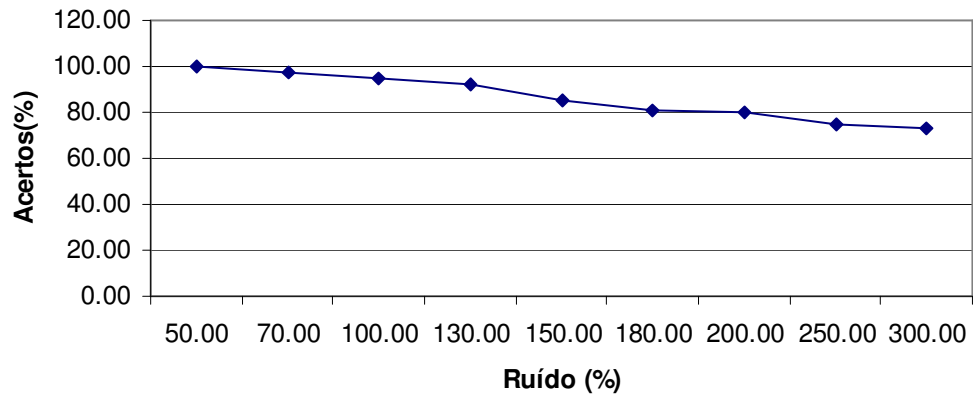


**Figura 13 – Curvas de Resistência dinâmica (ruído de 200%)**



**Figura 14 – Percentual de erros nas etapas de validação e testes da RNA**

Após a realização de simulações com diferentes níveis de ruído, foi possível construir um gráfico contendo a porcentagem de acertos da RNA em função do ruído inserido nas amostras (vide figura 15).



**Figura 15 - Desempenho da RNA para diversos níveis de ruído**

## 7. Conclusão

Ao final deste trabalho, pode-se concluir que as RNAs constituem ferramenta poderosa para problemas de identificação de padrões de curvas. Até mesmo os testes realizados com curvas com alto ruído (150% de variação dos valores de resistência) obtiveram resultados dentro de uma faixa de acertos maior que 80%.

Salienta-se que para elaborar a RNA e testar o algoritmo desenvolvido, realizaram-se simulações (em computador). Os resultados foram satisfatórios e corroboraram o estudo proposto. Entretanto, é válido ressaltar a importância da obtenção de dados experimentais reais, para que haja a validação do trabalho e sejam conhecidas as verdadeiras limitações do código.

A utilização das RNAs para análise de medições de resistência dinâmica dos disjuntores pode proporcionar um ganho real aos serviços de manutenção de disjuntores. Seguem algumas sugestões para trabalhos futuros:

- Aperfeiçoamento do código implementado;
- Avaliação de outros algoritmos de treinamento;
- Comparação do método utilizado com outras técnicas.

## 8. Bibliografia

A. C. C. Carvalho, A. P. Puente, A. Fuchs, C. M. Portela, E. J. Gueratto, D. D. Figueiredo, F. M. Salgado Carvalho, G. Garcia Junior, I. M. de Souza, J. Amon Filho, J. B. Almeida, J. S. Teixeira, L. P. S. Silva, M. Asano, M. A. G. Drummond, M. Lacorte, M. A. Vorpe, O. Kastup Filho, R. Colombo, S. V. Fernandes Júnior, S. A. Morais, S. O. Frontin, W. J. França. *Disjuntores e Chaves – Aplicações em sistemas de potência*. Furnas – Editora da Universidade Federal Fluminense, 1995.

Chapra, S. C., Canale, R. P., *Numerical Methods for Engineers – With Programming and Software Applications*, 3rd Ed., New York: WCB-McGraw-Hill, 1998.

Colombo, Roberto, *Disjuntores de alta tensão*, 1ª Edição, São Paulo: Nobel – Siemens S.A., 1986 – 1987.

Haykin, S., *Redes Neurais, Princípios e Prática*. 2a. ed., Porto Alegre: Bookman, 2001.

Landry, M., Mercier, Ouellet, G., Rajotte, C., Caron, J., Roy, M. and Brikci, F., *A New Measurement Method of the Dynamic Contact Resistance of HV Circuit Breakers*, 2006 IEEE IPES Transmission and Distribution Conference and Exposition Latin America, Venezuela.

Landry, M., Turcotte, O. and Brikci, F., *A Complete Strategy for Conducting Dynamic Contact Resistance Measurements on HV Circuit Breakers*, IEEE Transactions on Power Delivery, Vol. 23, No. 2, April, 2008.

## 9. Apêndice A - Construção das Curvas Sintéticas

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INTERPOLAÇÃO SPLINE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Taisa Felix

clear all
close all
clc

%RESISTENCIA DINAMICA DE CONTATO EM FUNCAO DA DISTANCIA ENTRE OS CONTATOS
%(ABERTURA LENTA)
%A new measurement Method of the Dynamic Contact Resistance
%of HV Circuit Breakers, p. 5 Disjuntores B e C)

%DISTANCIA ENTRE OS CONTADOS (mm)
x = [00 05 10 15 20 23 25 26 27 28 29 30 31 32 ...
     33 33.5 33.7 34 34.1 34.3 34.7 35 36 37 37.5 38 39 ...
     40 41 41.5 42 42.5 43 44 44.5 45 46 47];

%RESISTÊNCIA - DISJUNTOR ANTES DA REVISÃO (*10^-3)
yb = [0.05 0.05 0.05 0.05 0.05 0.05 0.15 0.24 0.17 0.22 0.18 0.4 0.42 0.5 ...
     0.52 0.7 0.85 0.5 0.65 0.52 0.6 0.5 0.6 0.47 1.1 1.25 0.95 0.95 ...
     1.1 0.7 0.45 0.45 0.3 0.3 0.25 0.25 0.35 3 ];

%RESISTÊNCIA - DISJUNTOR DEPOIS DA REVISÃO (*10^-3)
yc = [0.03 0.03 0.03 0.03 0.03 0.03 0.05 0.05 0.05 0.08 0.08 0.1 0.12 0.15 ...
     0.15 0.15 0.15 0.15 0.18 0.18 0.18 0.2 0.2 0.2 0.2 0.2 0.2 ...
     0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.3 0.5 1.5 3];

for i=1:2
    if i==1, y=yb;
        elseif i==2, y=yc;
    end
n = length(x)-1; %NUMERO TOTAL DE INTERVALOS

for k=1:length(x)-1
    h(k+1) = x(k+1)-x(k);
end

%CALCULO DA MATRIZ A (A*g = B)
A = zeros(n-1,n+1);
c = 1;

for L=1:(n-1)
    A(L,c) = h(L+1);
    A(L,c+1)= 2*(h(L+1)+h(L+2));
    A(L,c+2) = h(L+2);

    c=c+1;
end

%CALCULO DA MATRIZ B (A*g = B)
for L=1:(n-1)
    B(L,1) = 6*[(y(L+2)-y(L+1))/h(L+2) - (y(L+1)-y(L))/h(L+1)];
end

%CALCULO DA MATRIZ g (A*g = B)
g = A\B;

```

```

g(1)=0;      %CONDICOES DO SPLINE CUBICO NATURAL
g(n+1)=0;

%COEFICIENTES DAS EQUACOES CUBICAS DOS N INTERVALOS
for k=1:n
    a(k) = (g(k+1)-g(k))/(6*h(k+1));
    b(k) = g(k+1)/2;
    c(k) = (y(k+1)-y(k))/h(k+1)+(2*h(k+1)*g(k+1) + g(k)*h(k+1))/6;
    d(k) = y(k+1);
end

%EQUACOES DOS N INTERVALOS (SPLINE)
j=1;
t=x(1);
P = 0.01;    %VARIAVEL QUE DETERMINA A PRECISAO DA BASE DE DADOS R

for k =1:n
    X = x(k):0.0001:x(k+1);
    Y = a(k)*(X-x(k+1)).^3 + b(k)*(X-x(k+1)).^2+ c(k)*(X-x(k+1))+d(k);

    %GERANDO O VETOR R (BASE DE DADOS DO DISJUNTOR)
    while t<=x(k+1)
        t(j+1)=t(j)+P;
        R(j)= a(k)*(t(j+1)-x(k+1)).^3 + b(k)*(t(j+1)-x(k+1)).^2+ c(k)*(t(j+1)-
x(k+1))+d(k);
        j=j+1;
    end
end

%RESISTENCIA DINAMICA DE UM DISJUNTOR SAUDELAVEL
z = length(t)-2;
XB = 0:P:z/(P^-1);

figure(i)

    if i==1; title ('Curva da resistencia dinãmica dos contatos (Disjuntor
com Defeito)');
    elseif i==2; title ('Curva da resistencia dinãmica dos contatos
(Disjuntor Novo)');
    end

ylabel('Resistªncia dinãmica (m\Omega)')
xlabel('Distancia entre os contatos do disjuntor (mm)');
axis([0 50 0 7]);
hold on;

d = [rand(length(R),1)];    %RUIDO
p = 3;    %PORCENTAGEM DO RUIDO (DE ACORDO COM A RESISTENCIA)

j = 500;    %NUMERO DE CURVAS QUE SERAO SUPERPOSTAS

%ADICIONANDO RUIDO AO VETOR R

rand('state',sum(100*clock));

for k=1:length(R)
    r(k) = -p*R(k) + 2*(p*R(k))*d(k);    %RUIDO
    Rr(k) = R(k) + r(k);
end

```



```
%CURVAS
for k=1:j %SUPERPOSICAO DAS CURVAS
    if i==1 plot(XB,Rr);
        elseif i==2 plot(XB,Rr);
    end
    hold on;
end

%CURVA ENCONTRADA
plot(XB,R,'b')
hold on;

%PONTOS "MEDIDOS"
plot(x,y,'r.');
```

end

## 10. Apêndice B – Rede Neural Artificial

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RNA - RETROPROPAGAÇÃO %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Taisa Felix

clc
clear all
close all

% Quantidade de neurônios na rede
qnr = [15 2]; %15 neuronios na entrada e 2 na saida

% Valor maximo de epocas
maxepocas = 1e4;

%Coeficiente de Aprendizagem
eta = 0.01;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Critérios de Parada %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% PARADA POR PLATEAU

% Porcentagem do Número de Épocas Instantâneo a partir da qual o critério
% de parada deve ser aplicado

cpP = 2;

% Tolerancia da variação do erro de treinamento instantaneo (Valor
% Absoluto)
tolP = 1e-6;
%Inicializando o contador do Critério de Parada
ContP = 0;

% PARADA POR VICIO DE APRENDIZAGEM

% Porcentagem do Número de Épocas Instantâneo a partir da qual o critério
% de parada deve ser aplicado
cpV = 0.5;
% Tolerancia da variação da diferença entre o Erro de Treinamento e o Erro
% de Validação (Valor Absoluto)
tolV = 1e-6;
%Inicializando o contador do Critério de Parada e a variável para
% armazenamento do somatório das diferenças
ContV = 0;

% Constantes para a tangente hiperbólica
ca=1;
cb=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONSTRUINDO A BASE DE DADOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

numpad = 500; %numero de amostras
numpad1 = numpad/2;

%PADRÃO DE RESISTENCIAS (DISJUNTOR RUIM)
x1b = 0.15;
x2b = 1;
```

```

x3b = 0.5;
x4b = 1.5;
x5b = 3;

%PADRÃO DE RESISTENCIAS (DISJUNTOR BOM)
x1c = 0.05;
x2c = 0.1;
x3c = 0.15;
x4c = 0.15;
x5c = 3;

rand('state',sum(100*clock));

r = rand(numpad,5);
p = 10; %VARIACAO PERCENTUAL NA RESISTENCIA

for i=1:numpad
    if i<=numpad1
        xr(i,1) = x1b - p*x1b + 2*p*r(i,1)*x1b; %DISJUNTOR COM DEFEITO
        xr(i,2) = x2b - p*x2b + 2*p*r(i,2)*x2b;
        xr(i,3) = x3b - p*x3b + 2*p*r(i,3)*x3b;
        xr(i,4) = x4b - p*x4b + 2*p*r(i,4)*x4b;
        xr(i,5) = x5b - p*x5b + 2*p*r(i,5)*x5b;
    elseif i>numpad1
        xr(i,1) = x1c - p*x1c + 2*p*r(i,1)*x1c; %DISJUNTOR BOM
        xr(i,2) = x2c - p*x2c + 2*p*r(i,2)*x2c;
        xr(i,3) = x3c - p*x3c + 2*p*r(i,3)*x3c;
        xr(i,4) = x4c - p*x4c + 2*p*r(i,4)*x4c;
        xr(i,5) = x5c - p*x5c + 2*p*r(i,5)*x5c;
    end
    i = i+1;
end

xr = [xr(:,1) xr(:,2) xr(:,3) xr(:,4) xr(:,5) ones(numpad,1)];
x = xr;

indices = randperm(length(xr));

%PERMUTANDO OS ELEMENTOS DA MATRIZ
for k = 1:length(x)
    if (k <=numpad1)
        x(indices(k),:) = xr(k,:); %vetor com a mistura de dados corretos
e incorretos
        d(:,indices(k)) = [-1 -1];
    elseif k>numpad1
        x(indices(k),:) = xr(k,:); %vetor com a mistura de dados corretos
e incorretos
        d(:,indices(k)) = [1 1];
    end
end

%Divisão dos conjuntos de Treinamento, Validação e Teste
Xtr = x(1:round(0.7*numpad),:);
Xval = x(round(0.7*numpad+1):round(0.85*numpad),:);
Xtes = x(round(0.85*numpad+1):numpad,:);

%Divisão dos conjuntos de Treinamento, Validação e Teste
Dtr = d(:,1:round(0.7*numpad));
Dval = d(:,round(0.7*numpad+1):round(0.85*numpad));
Dtes = d(:,round(0.85*numpad+1):numpad);

%Valores dos pesos inicialmente

```

```

w1 = -1 + 2*rand(size(x,2),qnr(1));
w2 = -1 + 2*rand(qnr(1)+1,qnr(2));

for epoca = 1:maxepocas

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TREINAMENTO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Propagação na rede (Camada 1)
    v1 = Xtr*w1;

    % Calculo das saidas (1) a partir do potencial de ativação
    Ytr1 = ca*tanh(cb*v1);

    % Adicionando o bias as saidas da camada 1
    Xtr2 = [Ytr1 ones(size(Ytr1,1),1)];

    % Propagação na rede (Camada 2)
    v2 = Xtr2*w2;

    % Calculo das saidas (2) a partir do potencial de ativação
    Ytr2 = ca*tanh(cb*v2);

    % Calculo do erro de treinamento instantaneo
    Etr = Dtr - Ytr2';

    % Calculo dos deltas da camada de saida, que permitira a atualização
    % dos pesos da camada de saida.
    % 'delta' tem as mesmas dimensoes de 'Etr' e de 'Ytr':
    % numero de neuronios da camada de saida Xtr numero de padroes
    % delta = derivada da função de ativação * erro
    delta2 = ((cb/ca) * (ca*ones(size(Ytr2))-
Ytr2).*(ca*ones(size(Ytr2))+Ytr2)) .* Etr';

    % Para calcular o ajuste de pesos da camada de saida, vamos precisar
    % dos padroes de entrada utilizados na camada de saida. Estes valores
    % sao Xtr.
    %dw = eta * delta * entradas;;
    dw2 = ((eta * delta2)' * Xtr2)/length(Xtr);

    %Armazenando o valor do Erro (NÚMERO)
    acumulaMedEtr(epoca) = sum((sum(Etr.^2)/2)/length(Xtr));

    %Correção dos pesos da camada de saída
    w2 = w2 + dw2';

    % Calculo dos deltas da camada oculta, que permitira a atualização
    % dos pesos da camada oculta.
    delta1 = ((cb/ca).*(ca*ones(size(Ytr1))-
Ytr1).*(ca*ones(size(Ytr1))+Ytr1)) .* (w2(1:size(w2,1)-1,:) * delta2)');

    % Para calcular o ajuste de pesos da camada oculta, vamos precisar
    % dos padroes de entrada utilizados na camada oculta. Estes valores
    % sao x1, y1 acrescidos dos bias.
    dw1 = (eta .* delta1' * Xtr)/length(Xtr);

    %Correção dos pesos da camada de saída
    w1 = w1 + dw1';

    clear v1 N

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VALIDAÇÃO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Propagação na rede (Camada 1)
v1 = Xval*w1;

% Calculo das saidas (1) a partir do potencial de ativação
Yval1 = ca*tanh(cb*v1);

% Adicionando o bias as saidas da camada 1
Xval2 = [Yval1 ones(size(Yval1,1),1)];

% Propagação na rede (Camada 2)
v2 = Xval2*w2;

% Calculo das saidas (2) a partir do potencial de ativação
Yval2 = ca*tanh(cb*v2);

% Calculo do erro de treinamento instantaneo
Eval = Dval - Yval2';

%Armazenando o valor do Erro (NÚMERO)
acumulaMedEval(epoca) = sum((sum(Eval.^2)/2)/length(Xval));

clear v1 N

% Parada pelo Critério de Plateau
if epoca >= 2
    if abs(acumulaMedEtr(epoca)-acumulaMedEtr(epoca-1)) < tolP
        ContP=ContP+1;
        if ContP >= epoca*cpP/100
            break
        end
    else
        ContP = 0;
    end
end

% Parada pelo Critério de Vício de Aprendizagem
if acumulaMedEval(epoca)-acumulaMedEtr(epoca) > tolV
    ContV=ContV+1;
    if ContV >= epoca*cpV/100
        %break
    end
else
    ContV = 0;
end

end

figure
plot([1:1:epoca],acumulaMedEtr,'b-',[1:1:epoca],acumulaMedEval,'r-')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TESTE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Propagação na rede (Camada 1)
v1 = Xtes*w1;

% Calculo das saidas (1) a partir do potencial de ativação
Ytes1 = ca*tanh(cb*v1);

% Adicionando o bias as saidas da camada 1
Xtes2 = [Ytes1 ones(size(Ytes1,1),1)];

```

```

% Propagação na rede (Camada 2)
v2 = Xtes2*w2;

% Calculo das saidas (2) a partir do potencial de ativação
Ytes2 = ca*tanh(cb*v2);

% Arredondando o vetor de saída
YtesBin = zeros(size(Ytes2));
Ytesmaiorzero = find(Ytes2>=0);
Ytesmenorzero = find(Ytes2<0);
YtesBin(Ytesmaiorzero) = 1;
YtesBin(Ytesmenorzero) = -1;

erros = 0;

Dtes = Dtes';

for k = 1:length(YtesBin)
    if YtesBin(k,:) - Dtes(k,:) == [0 0]

        else
            erros = erros + 1;
        end
    end
end

% Taxa de erros
disp('Acertos de teste (%):')
disp(100-erros/length(Dtes)*100)

```