Ml LDH MARIA DA SILVA LIRA

SISTEMAS HÍBRIDOS APLICADOS À FILTRAGEM DE SINAIS DE ALARMES DE PROTEÇÃO DE UMA SUBESTAÇÃO TELECOMANDADA

Dissertação apresentada ao Curso de Mestrado em Engenharia Elétrica da Universidade Federal de Pernambuco, em cumprimento às exigências para obtenção do grau de mestre.

Orientador: Prof. Dr. Manoel Afonso de Carvalho Junior

RECIFE-PE
1999

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE DISSERTAÇÃO DE MESTRADO DE

MILDE MARIA DA SILVA URA

TÍTULO

"SISTEMAS HÍBRIDOS APLICADOS À FILTRAGEM DE SINAIS DE ALARMES DE PROTEÇÃO DE UMA SUBESTAÇÃO TELECOMANDADA"

A comissão examinadora composta pelos professores: MANOEL AFONSO DE CARVALHO JÚNIOR, DEESP/UFPE, WELLINGTON SANTOS MOTA, DEE/UFPB e ANTONIO CARLOS GAY THOMÉ, UFRJ, sob a presidência do primeiro, consideram a candidata MILDE MARIA DA SILVA LIRA APROVADA.

Recife, 25 de novembro de 1999.

MANOEL ÀFÓNSITDEXARVALHO JUNIOR

/ ^ < ^ " ^ C "h-—

SISTEMAS HÍBRIDOS APLICADOS À FILTRAGEM DE SINAIS DE ALARMES DE PROTEÇÃO DE UMA SUBESTAÇÃO TELECOMANDADA

Com muito amor, ao meu marido Carlos Brayner e às minhas filhas Maria do Carmo e Mariana, DEDICO.

AGRADECIMENTOS

Agradeço a Deus por ter-me concedido condições intelectuais e emocionais para o desenvolvimento deste trabalho.

Ao Prof. Manoel Afonso de Carvalho Júnior pela calorosa acolhida a este Mestrado, pela orientação segura e por todo o apoio e compreensão constante, fundamentais para efetivação desta dissertação.

Ao meu marido, Brayner, por sua dedicação durante a organização deste trabalho, principalmente no que se refere a esta redação.

A Coordenação e Secretaria do mestrado, em especial a Andréa Tenório pelo carinho e atenção com que sempre me atendeu.

Aos Eng. Marcello de Martino (CEPEL), João Carlos e José Ernandes (CHESF-DOPR) e ao Prof. Antônio Carlos Thomé (UFRJ-NCE), pela gentileza e empenho ao me fornecerem as informações, permitindo a realização deste trabalho.

A CAPES pela bolsa de estudos, sem a qual este trabalho não teria sido possível.

As minhas filhas, Maria do Carmo e Mariana, por tolerarem a minha falta de atenção durante minhas horas de estudo e pesquisas.

Agradeço ainda, a todas as pessoas que de alguma forma contribuíram para a conclusão desta dissertação.

RESUMO

Neste trabalho, um Sistema Híbrido de Inteligência Artificial é desenvolvido e aplicado ao processo de filtragem de um conjunto de sinais de alarmes, gerados em uma subestação. Estes sinais após sofrerem o processo de filtragem devem chegar em um número reduzido ao CROL (Centro Regional de Operação Leste), local de onde a subestação será telecomandada. O Sistema Híbrido desenvolvido consiste de uma Rede Neural Artificial do tipo MLP- backpropagation e um Sistema de Inferência Fuzzy (SIF) do tipo Mamdani. As Redes Neurais implementadas neste trabalho, quando atuando sozinhas, foram capazes de fornecer bons resultados na classificação dos eventos, apresentando um elevado percentual de acerto, 94 a 100%, no conjunto de teste. Com a finalidade de tornar o sistema ainda mais confiável, os resultados de cada Rede Neural foram submetidos aos seus respectivos SIF, mostrando excelentes resultados ao corrigir todos os casos mal classificados pelas Redes Neurais e demonstrando assim a viabilidade do Sistema Híbrido para esta função de filtragem.

Palavras Chaves: Alarmes de Proteção, Subestação Telecomandada, Redes Neurais, Lógica Fuzzy, Sistemas Híbridos.

ABSTRACT

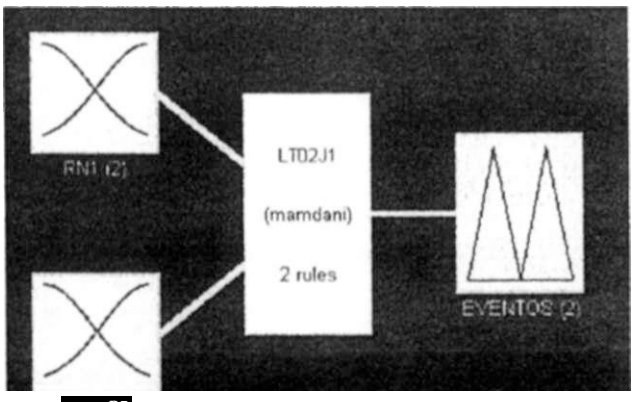
In this work, an Artificial Hybrid System is developed and applied to the filtering process of an alarm's signal set that is generated at a power substation. After the filtering process, a reduced number of signals is sent to the power system control center, from where the substation is remote controlled. The developed Hybrid System consists of an Artificial Neural Network (ANN) of the MLP- backpropagation type and a Fuzzy Inference System (FIS) of the Mamdany type. The implemented ANNs, acting alone, have provided good results in the classification of events, providing the right answer to 94-100% of the test cases. To improve the reliability of the system further, the results of each ANN were submitted to their respective FIS, which in turn showed great performance as they were able to correct all wrongly classified events by the neural network. The Hybrid System has demonstrated to be a viable tool for the signal filtering process.

Keywords: Protection Alarm, Telecommanded Substation, Neural Network, Fuzzy Logic, Hybrid System.

SUMÁRIO

A mesma metodologia foi empregada na montagem do sistema fuzzy dos demais módulos, de forma que todos os módulos utilizam o mesmo tipo de função de entrada (gaussiana) e saída (triangular), com os parâmetros da função gaussiana para os termos linguisticos nulo (0.45 0 0 0) c certeza (0.45 1 0 0), c os parâmetros da função triangular para os termos linguisticos evento 1 (0 0.5 1 0), evcnto2 (1 1.5 2 0), evento3 (2 2.5 3 0), etc; e a alteração dos pesos das regras em função do estado do conjunto de alarmes referentes a cada módulo. lista alteração dos pesos será discutida na próxima seção. A figura 5.9 mostra o Sistema de Inferência Fuzzy do módulo LT-02J, composto por duas entradas, cujos temos são **nulo** e **certeza** c uma saída, cujos termos são **eventol** c **evento2**

Como os módulos TR-04T1 e I.T-04F3 são compostos por um número de eventos maior que 9, que é o número máximo de termos linguisticos de cada variável aceitável pelo "Fuzzy Logic Toolbox", necessitou-se montar um sistema de inferência complementar para cada um desses módulos. Estes foram montados da mesma forma que o anterior, alterando-se apenas o número de termos linguisticos complementar da variável de saída (EVENTOS) e consequentemente as regras referentes a esses eventos. Estes dois sistemas, ou seja, o sistema de inferência e o seu complementar são conectados dentro do programa de alteração dos pesos das regras.



Pn.' (2J

".•/st-n. :.T;*i2 !! *2* inputr, I oiit&uts, *2* ru!*s

Fig. 5.9 Sistema de Inferência Fuzzy do módulo LT-02J1

Além da escolha dos tipos de funções membros, fixação dos seus parâmetros e mudança dos pesos das regras segundo a configuração dos alarmes, os seguintes processos foram empregados para a montagem do Sistema de Inferência tipo Mamdani de cada módulo: da lógica E = min; da lógica OU = max; de implicação = min; de agregação = max e finalmente de defuzificação = centróide.

5.4.1 - Programa de alteração dos pesos das regras

Com a finalidade de alterar os pesos das regras montou-se dentro da plataforma do MatLab um programa para cada um dos módulos [Anexo III], que lê o **arquivo de entrada** da rede neural, processa esses dados e fornece a atualização dos pesos ao Sistema de Inferência Fuzzy. E importante salientar que as regras da Lógica Fuzzy são as mesmas para cada um dos módulos, qualquer que seja a configuração do conjunto dos alarmes. O que se altera, são os pesos das regras.

Este programa realiza ainda a leitura do **arquivo de saída** da rede neural fornecido pelo simulador de redes neurais - SiReNe, e disponibiliza os valores da saída de todos os exemplos do conjunto de teste para a utilização no sistema de inferência fuzzy. O exemplo específico a ser analisado é selecionado pelo usuário na plataforma de trabalho do MatLab. Além disso, para melhor compreensão dos resultados da defuzificação do sistema de inferência foi implementada uma saída de forma lingüística (tabela 5.5). Nesta tabela, para os intervalos de defuzificação de cada um dos módulos, tem-se o evento ocorrido em formato lingüístico, que o torna inteligível.

Tab. **5.5** Forma linguistica da dcfuzificaçâo

n. Módulo Intervalos. dc >v dcfuzificaçâo \	TR-04T1	LT-04F3	LT-04C4	LT-02J1
0 - 1	ANORM PROTEÇÃO	ANORM_ PROTEÇÃO	ANORM_ PROTEÇÃO	PROT_SOBREC FASE
1 - 2	DESLIG_ BLOQUEIO	ATUAÇAO_ PROTEÇÃO	ATUAÇAO_ PROTEÇÃO	PROT_SOBREC NEUTRO
2 - 3	PROT_SOBREC_ FASE	DISJ_ANORM_ ChLoc Manut	ANORM ALIDADE_ DISJUNTOR	-
3 - 4	PROT_INTRINS_ 1°GRAU	ANORMALIDADE_ VÀO_LT_04F3	ANORM ALIDADE_ SECCIONADORA	-
4 - 5	ANORMALIDADE_ DISJJ4T1	DISJ_ABERTURA_ BAIXA_PRESSÃO	DISCORDÂNCIA, DE_POLOS	-
5 - 6	ANORMALIDADE_ PCP_()4TI	DISJUNTOR_APTO_ REL1GAR	PROTEÇAO_FALHA_ DISJ_DESL1G	-
6 - 7	ANORMALIDADE, SEC_34Tl_4/6	DISJUNTOR_BLOQUEIO _FUNÇÃO	BLOQUEIO_ FALH AFUZÍ VEL	-
7 - 8	BLOQUEIO_FUNÇAO_ DISJUNTOR	DISJUNTOR_NÃO_ DESLIGOU	BLOQUEIO_FUNÇAO _DISJUNTOR	-
8 - 9	OLTC_DISC_TAP_ ENTRETRAFO	PROT_FALUA_DISJ_ DESLIG	-	-
9 - 10	SOBTEMP_ PREDISP_TR1P	SECCIONADORA_ FALHAOPERAÇÃO	-	-
10-11	TEMPANORM A L I G	TELEPTRANSM_ PERMIS_TRIP	-	-
11-12	DISCORD_POLOS_ DISJUNTOR	-	-	-
12 - 13	FALHA_ DISJUNTORJ4T1	-	-	-
13-14	NAODESLIG DISJ_14T1	-	-	-
14 - 15	FALHA_SISTEMA_ REFRIGERAÇÃO	-	-	-

CAPÍTULO 6

RESULTADOS E DISCUSSÕES

Primeiramente, serão analisados os resultados fornecidos pela Rede Neural e em seguida os resultados fornecidos pelo Sistema Híbrido.

A tabela 6.**1** apresenta os resultados obtidos para os quatro módulos, utilizando apenas o sistema de redes neurais. Esta tabela contem os erros RMS para os conjuntos de treinamento e de teste, os números dc exemplos de teste, os números de acertos, de erros e as percentagens de acertos. O número de acertos é tomado como o número de vezes que a rede classificou corretamente o evento ocorrido. O critério adotado para se obter uma saída binária **0** ou **1** foi estabelecido da seguinte forma:

$$RN > \mathbf{0,5} \Rightarrow RN = 1$$
$$RN < \mathbf{0,5} \Rightarrow / \mathbf{W} = \mathbf{0}$$

onde *RN* é a saída da rede neural, cujo valor varia de **0-1.** De forma similar, definiu-se os números de erros e as percentagens de acertos.

O maior valor do erro RMS no conjunto de treinamento foi de 1,1 Oxl O"[2], enquanto no conjunto de teste o maior valor encontrado foi de 5,86x10"[2], ambos para o módulo LT-04F3. O erro RMS é calculado segundo a equação 5.1 c é adimensional, pois os valores que o originam são adimensionais.

Observa-se que o erro no conjunto de treinamento foi sempre menor que o erro no conjunto de teste, embora o contrário possa acontecer [12]. O resultado obtido neste trabalho é o mais provável de ocorrer, visto que a rede calcula as saídas para entradas que não lhes foram apresentadas no conjunto de treinamento. Não existe nenhuma forma de prever como a rede irá se comportar diante de exemplos de teste apresentados após o treinamento. Por esta razão ao se treinar a rede é aconselhável monitorar o seu comportamento em um conjunto de dados de teste. Apesar dos erros RMS no conjunto de teste apresentarem valores maiores que os valores do conjunto de treinamento verificamos que a percentagem de acerto, mostrada na última coluna da tabela 6.1, é bastante elevada, significando que o treinamento preparou as redes com capacidade para generalizar.

A rede LT-02J1 não apresenta conjunto de teste, pelo fato de serem esses seis exemplos todos os casos possíveis de ocorrerem. Logo, para o cálculo do percentual de acerto, tomou-se os exemplos de treinamento, que mostra um percentual de 100% de acerto. Este resultado já é esperado devido a pequena dimensão da rede e principalmente ao fato de todos os casos possíveis estarem sendo apresentado no treinamento, os quais são responsáveis pela fixação dos pesos das conexões entre os neurônios.

Tab. 6.1 Resultado das redes neurais após treinamento

Rede Neural	RMS		Número de exemplos de Teste	Número de Acertos	Número de Erros	Percentagem de acerto (%)
	Conjunto de treinamento	Conjunto de teste				
LT-04F3	**1,10x1o*³**	5,86x10'²	**120**	**113**	7	94
LT-04C4	2,70x10"³	2.61xl0"²	66	65	**1**	98
TR-04T1	5,43x10"³	1,14x10"'	33	33	0	100
LT-02J1	8,65x10"³	-	6	6	0	100

Os resultados obtidos utilizando apenas a rede neural, na tabela 6.1, mostram que a rede apresentou um bom desempenho na tarefa de classificação, apresentando um percentual de acerto bastante elevado (94% a 100%) no conjunto de teste, porém a necessidade de sistemas que apresentem maior confiabilidade é imprescindível em Sistema de Potência, e por isto a preocupação em melhorar esses resultados deu origem ao Sistema Híbrido utilizado neste trabalho.

Utilizando o Sistema Híbrido, os resultados apresentados confirmam o objetivo do trabalho, cuja principal finalidade é classificar corretamente o evento, dado o estado do conjunto dos alarmes.

A seguir serão analisados em maiores detalhes os resultados apresentados, pela Rede Neural e em seguida pelo Sistema de Inferência Fuzzy complementando o Sistema Híbrido, para seis exemplos devidamente selecionados. Estes seis exemplos escolhidos são do módulo LT-04F3 por este ser um dos módulos que apresentou um maior percentual de erro.

Tab.6.2 Saída desejada e obtida pela Rede Neural do módulo LT-04F3.

	Exemplo 1	Exemplo 2	Exemplo 3	Exemplo 4	Exemplo 5	Exemplo 6
Saída desejada 1	1	0	0	1	0	1
Saída obtida 1	0,9900	0,0021	0,0262	0,9891	0,0000	0,9983
Saída desejada 2	0	1	0	0	1	' 1 \.
Saída obtida 2	0,0162	0,9994	0,0562	0,0151	0,9995	0,8080
Saída desejada 3	0	1	1	1	0	0
Saída obtida 3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Saída desejada 4	1	0	1	0	0	0
Saída obtida 4	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Evento desejado (código)	1 00 1	0 1 10	00 11	10 10	0 100	1 100
Evento classificado (código)	1000	0 100	0000	1000	0 100	1100

A tabela 6.2 mostra os 6 exemplos, onde os exemplos 1, 2, 3 c 4 foram mal classificados pela Rede Neural representativa do módulo LT-04F3. Os resultados preenchidos, das saídas desejadas e obtidas, foram extraídos do arquivo de saída do SiReNc. As linhas em cinza referem-se aos resultados desejados e as linhas em branco, aos obtidos. Nesta tabela observa-se que a Rede Neural, além de classificar os exemplos 1, 2 e 4 em classes erradas, ela classificou o exemplo 3 em uma classe (0000) que não existe neste módulo, como pode se verificar na tab.6.3.

Tab. 6.3 Código e descrição dos eventos do módulo LT-04F3

Código dos eventos	Descrição dos eventos
1000	ANORMAL PROTEÇÃO
0 100	ATUAÇÃO PROTEÇÃO
00 10	DISJ ANORM ChLocManut
1100	ANORMALIDADE VAO LT-04F3
100 1	DISJ ABERTURA BAIXA PRESSÃO
0 110	DISJUNTOR APTO RELIGAR
00 11	DISJUNTOR BLOQUEIO_FUNÇAO
10 10	DISJUNTOR NÃO DESLIGOU
0 10 1	PROT FALHA DISJ DESLIG
1110	SECCIONADORA FALHA OPERAÇÃO
110 1	TELEP TRANSM PERMIS TRIP

As saídas da Rede Neural de cada um dos exemplos, através do programa de alteração dos pesos das regras, foram submetidas ao Sistema de Inferência Fuzzy, resultando, do processo da defuzificação, os valores mostrados na coluna destacada da tabela 6.4.

Tab. 6.4 Resultado da defuzificação

Exemplo	Saída da Rede Neural				Lógica Fuzzy (defuzificação)
	RN 1	RN2	RN3	RN4	
1	0,9900	0,0162	0,0000	0,0000	4,5000
2	0,0021	0,9994	0,0000	0,0000	5,4923
3	0,0262	0,0562	0,0000	0,0000	6,4922
4	0,9891	0,0151	0,0000	0,0000	7,4927
5	0,0000	0,9995	0,0000	0,0000	1,5008
6	0,9983	0,8080	0,0000	0,0000	3,5002

Estas saídas numéricas correspondem a forma lingüística mostrada na coluna 3 da tabela 6.5, conforme os termos lingüísticos adotados para cada intervalo unitário da defuzificação associados ao módulo LT-04F3 (tab.5.5).

Tab. 6.5 Evento classificado pelo Sistema Híbrido

Exemplo	Defuzificação	Descrição do evento	Evento classificado (Código)	Evento desejado (Código)
1	2	3	**4**	5
1	4,5000	**DISJ_ABERTURA_ BAIXA PRESSÃO**	1001	1001
2	5,4923	**DISJUNTOR APTO RELIGAR**	0110	0110
3	6,4922	**DISJUNTOR BLOQUEIO FUNÇÃO**	0011	0011
4	7,4927	**DISJUNTOR NAO DESLIGOU**	1010 .	1010
5	1,5008	**ATUAÇÃO PROTEÇÃO**	0100	0100
6	3,5002	**ANORMALIDADE VÃO LT 04F3**	jfioo	1 100

Na tabela 6.5, a coluna destacada, foi preenchida de acordo com o código associado a descrição do evento (tab.6.3); e a coluna 5, de acordo com o código do evento desejado associado ao número do exemplo (tab.6.2).

Comparando-se as colunas 4 e 5, dessa tabela, conclui-se que o Sistema Híbrido (Rede Neural - Sistema de Inferência Fuzzy), para os exemplos 1, 2, 3 e 4, conseguiu classificar corretamente os eventos classificados erroneamente pela Rede Neural atuando sozinha; e para os exemplos 5 e 6, classificados corretamente, manteve a mesma classificação.

CAPÍTULO 7

CONCLUSÕES GERAIS E SUGESTÕES

7.1 - CONCLUSÕES GERAIS

Neste trabalho foi estudado o desempenho de um sistema híbrido - Rede Neural / Lógica Fuzzy - na função de filtragem de sinais de alarmes de proteção de uma subestação telecomandada.

Como resultado dos estudos desenvolvidos para realização deste trabalho foi possível extrair as seguintes principais conclusões:

=> No estudo com as Redes Neurais Artificiais (RNAs) verificou-sc que elas apresentaram um bom desempenho na tarefa de classificação.

=> As redes que apresentaram maior número de alarmes conectados pela mesma porta lógica OU forneceram maior erro percentual, em razão destas redes necessitarem de

LISTA DE TABELAS

um grande número de padrões de treinamento que transporte as características da classe do problema.

=> Os resultados após o processamento das saídas das RNAs pelo Sistema de Inferência Fuzzy (SIF) mostraram-se excelentes. Todos os eventos mal classificados pelas RNAs foram corrigidos pelo SIF, tornando o Sistema Híbrido mais confiável.

=> Alguns eventos classificados pelas RNAs no conjunto de teste, nem sequer existiam, como mostra o exemplo 3 apresentado pela Rede Neural do módulo LT-04F3, onde o código do evento classificado foi 0000, porém esta classificação não é relevante para o resultado final apresentado pelo Sistema Híbrido, visto que o evento em questão também foi corrigido pelo SIF.

7.2 - SUGESTÕES PARA TRABALHOS FUTUROS

=> Treinar as Redes Neurais para multiplicidade de eventos e conectá-las ao Sistema de Inferência Fuzzy adequado para essa nova configuração das Redes.

=> Neste trabalho as classificações são exclusivas, ou seja, apenas um evento é ativado, e esta saída ativada é interpretada como indicadora da categoria à qual o padrão de entrada pertence. E além disso, esses eventos são indicados por um código binário de 2 ou 4

dígitos, onde cada dígito é representado por um neurônio de saída. No caso de treinamento da rede para eventos múltiplos ativados simultaneamente, é preferível utilizar, para cada evento, um neurônio representativo de seu estado, ativado ou desativado.

REFERÊNCIAS BIBLIOGRÁFICAS

111 CHAN, Edward H. P. Using Neural Network To Interpret Multiple Alarms. IEEE Computer Applications in Power, Apr. 1990. p.33-37.

[2| SILVA, Victor Navarro A.. L. da., ZEBULUM, Ricardo Salem. Sistema Híbrido para diagnose em Sistemas de Potência utilizando Redes Neurais e Lógica Nebulosa. In: II Congresso Brasileiro de Redes Neurais, 1995, Curitiba.

|3| KEZUNOVIC, Mladen, RIKALO, Igor. Detect and Classify Faults Using Neural Nets. IEEE Computer Applications in Power, Oct. 1996. p.42-47.

|4| BIONDI NETO, Luiz, PACHECO, Marco Aurélio C, VELLASCO, Marley Maria B R et al Sistema Híbrido de apoio à decisão para detecção e diagnóstico de falhas em redes elétricas. In: III Simpósio Brasileiro de Redes Neurais, Nov. 1996, Recife, p. 197-204.

[5| SATO, Fujio. Análise de Faltas. Curso de Engenharia de Análise e Planejamento de Operação de Sistemas Elétricos - CEAPO, Out. 1986. UNICAMP.

|6| JARDINI, José Antônio. Sistemas Digitais para Automação da Geração, Transmissão e Distribuição de Energia Elétrica. São Paulo: FCA, 1996. 320p cap 4: Automação de Subestações, p.79-115.

|7| HAYKIN, Simon. Neural Networks: A Comprehensive Foundation. Prentice Hall, 1999. 842p cap 3: Single Layer Perceptrons, p. 118-143; cap4: Multilayer Perceptrons, p. 156-175.

|8| VIEIRA, Adriana Moura. Simulador de Redes Neurais. Rio: 1998. Trabalho de Conclusão de Curso. DEL-UFRJ.

|9| SILVA, Victor Navarro A. L. da, CARVALHO, L. A V. de. Artificial Neural Network for power Systems Diagnosis. IEEE Internacional Conference on Neural Networks, June. 1994, p.3738-3743.

|10] KOSKO, Bart. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice Hall, 1991. 425p cap 8: Fuzzy Associative Memories, p.299-335

MAMDANI, H. E., ASSILIAN, S. An experiment in linguistic syntheses with a fuzzy logic controller. International Journal of Man-Machine Studies, 1975 Vol. 7. N°. 1, p. 1-13.

[12] ZADEH, L. A. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetics, Jan. 1973. Vol.3, N°. 1, p.28-44.

[13] HSU, Yuang-Yih, SU, Chung-Ching. A Rule Based Expert System For Steady-State Stability Analysis. IEEE Transaction on Power Systems, May 1991. Vol. 6, N ° 2.

|14| ASSAF, Liliane Videira. NeuroLab - Ambiente de Desenvolvimento e Aplicação de Redes Neurais. Rio: 1999. Projeto Final de Curso. DCC/IM - UFRJ.

|15] KIM, Kwang-Ho, PARK, Jong-Kcun. Application of hierarchical neural network to fault diagnosis of power systems. Electrical Power & Energy Systems, Apr. 1993. Vol. 15, N° 2, p.65-70.

[16] WESLEY, C. Lefebvre. NeuroSolutions v2.0 for windows, 1995.

|17| JANG, J.S. Roger, GULLEY, Ned. Fuzzy Logic Toolbox for use with MATLAB, 1996.

[18] BEALE, R., JACSON, T. Neural Computing: An Introduction. Adam Hilger,1991. 240p cap 4: The Multilayer Pcrceptron, p.63-104.

[19] POLAK, E. Computational Methods in Optimization : A Unified Approach. New York : Academic Press, 1971. 329p.

[20] MASTERS, Timothy. Praticai Neural Network Recipes C++. Book&Dish, 1993. 493p.

APÊNDICE A

Problema do pêndulo invertido

A figura A. 1 mostra um banco de regras FAM necessárias para controlar um pêndulo invertido através da Lógica Fuzzy. Este é um problema de controle clássico que pode ser visto cm [10].

0, AO e v definem as variáveis fuzzy. As variáveis fuzzy 0 e AO definem as variáveis de estado do sistema. A variável angular fuzzy 0 mede o ângulo que o pêndulo faz com a vertical e varia de -90° a 90°. A variável da velocidade angular fuzzy AO mede a taxa de variação instantânea do ângulo. Na prática ela mede a diferença entre valores dos ângulos sucessivos . A variável de saída fuzzy v mede a corrente do mecanismo de controle de um motor que ajusta o pêndulo.

Cada variável pode assumir cinco valores do conjunto fuzzy: Negativo Médio (NM), Negativo Pequeno (NS), Zero (ZE), Positivo Pequeno (PS), e Positivo Médio (PM).

8

A0 \ 9	NM	NS	ZE	PS	PM
NM			PM		
NS			PS		
ZE	PM	PS	ZE	NS	NM
PS			NS		
PM			NM		

Fig. A. 1 Banco de Regras FAM para controlar um pêndulo invertido. Cada entrada na matriz FAM define uma associação fuzzy entre os conjuntos de saída fuzzy e os conjuntos dos pares de entrada fuzzy.

A entrada no centro da matriz FAM define o estado estável da Regra - FAM:

"SE 9 = **ZE** E A0 = **ZE**, ENTÃO v = **ZE**".

APÊNDICE B

B.1 - Função Membro

A função Membro é a curva que define o quanto cada ponto no espaço de entrada é mapeado em um valor membro (ou grau de pertinência) entre 0 e 1. O espaço de entrada é algumas vezes referido como *universo de discurso.*

Um dos mais comuns exemplos usados em conjuntos fuzzy é o conjunto da altura de pessoas. Neste caso, o universo de discurso é todo o potencial de alturas, de l,00m a 2,50m, e a palavra 'alto' corresponde a curva que define o grau com que cada pessoa é alta. Se o conjunto de altura é dado em uma fronteira de valores bem definidos de um conjunto clássico, pode-se dizer que todo indivíduo maior que l,70m é considerado oficialmente alto. Mas, tal distinção é claramente absurda. Pode fazer sentido considerar o conjunto de todos os números reais maiores que 1,70 porque os números pertencem a plano abstrato, mas quando se quer falar sobre pessoas reais, é irracional chamar uma pessoa de baixa c uma outra de alta quando elas diferem em altura de apenas a espessura de um fio de cabelo. Qual é então a maneira certa de definir o conjunto de pessoas altas? A figura B.l mostra uma curva variando suavemente que passa de não-alto para alto.

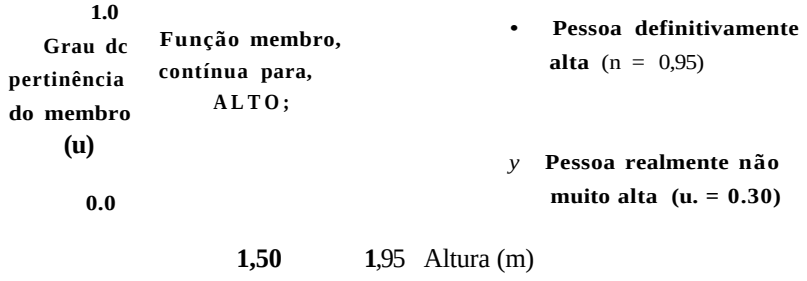
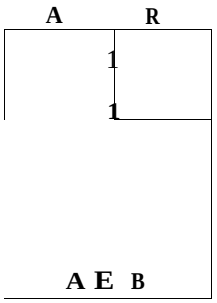


Fig. B. 1 Curva representativa da função membro das pessoas altas

O eixo de saída é um número conhecido como valor membro entre 0 e 1. A curva é conhecida como uma função membro e geralmente é designada por \i. Esta curva define a transição de não-alto para alto. Todas as duas pessoas são altas em um determinado grau, mas uma é significantemente menos alta que a outra.

B.2 - Operadores fuzzy

Na lógica fuzzy o operador E é representado pelo min(A,B); OU é representado pelo max(A,B), onde A e B estão limitado entre 0 e 1. Exemplos gráficos.



E
min(AB)

l i :

A OU B

OU
max(AB)

APÊNDICE C

Prova do Teorema SAM. O teorema se desenvolve através da expansão do centróide de *B(x)* e requer a suposição do SAM, ou seja, a suposição de que **& ^ j&j ~ S J i)^j** para organizar os termos:

$$F(x) = (\backslash > nimide\{B) = \wedge \text{—:} \frac{\text{í } yk.(y)dy}{} \quad (1.1)$$

$$l,y1L^{w}J^{b}'j(y)t!y \quad (1.2)$$

7=1

$$\frac{\geq \text{=}\langle}{m} \quad (1.3)$$

$$LH^{w}j^{w}A^{x})^{h}j(y)^{d}y$$

7=1

$$\frac{m}{T,^{w}J^{a}_{j}\langle^{x}\wedge i.py^{h}jM^{d}y}{./-'} \quad (1.4)$$

m

$$7\text{->}\frac{}{}*7 \quad (1.5)$$

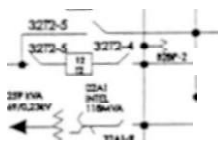
7=1

(1.6)

O modelo aditivo padrão envolve pouco cálculo. Calcula-se os volumes V_s e os centroides ***Cj*** antecipadamente. Para cada entrada ***x*** calcular-se os ***m*** valores ajustados ***a/x)*** e atualiza-se a taxa em :

/<(*) =——————————————————————————————(17)

As partes **Então** dos conjuntos Bj podem possuir formas triangulares ou trapezoidais ou curvas em forma de sino que possuem áreas e centroides simples.



MODIFICAÇÕES:

INDICAÇÃO DA POTENCIA DE TENSÃO DO TRAFO DE SERVIÇO ASSOQADO AO02A1

DOMO 09/07/99
DO-97.3.0075

COMPANHIA HIDRO ELÉTRICA DO SAO FRANCISCO
SE MIRUEIRA (MRR)

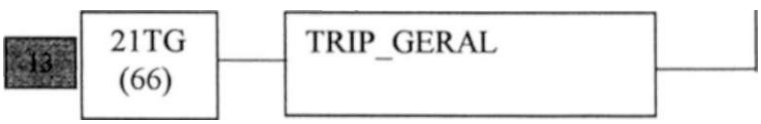
05,99

LISTA DE FIGURAS

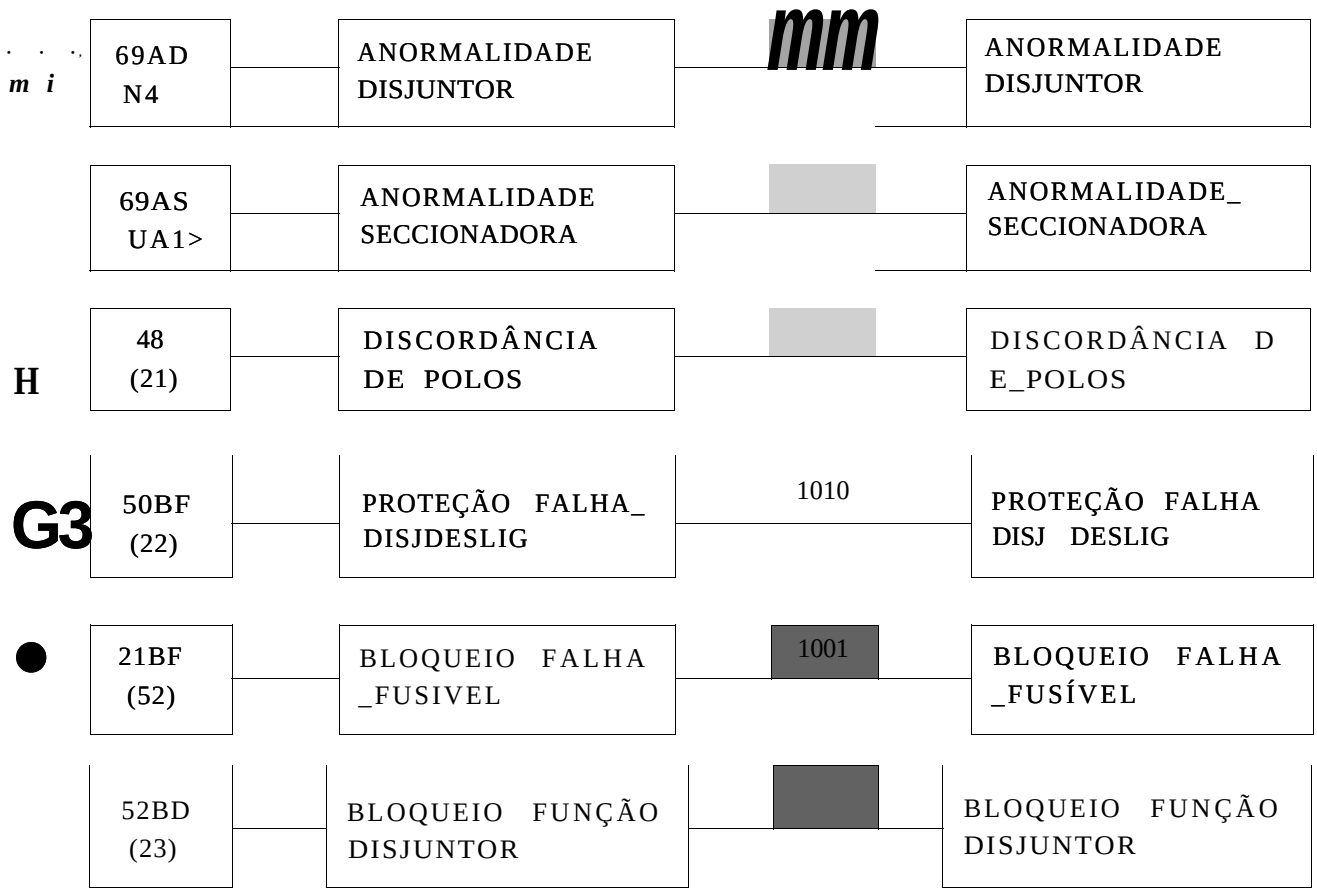
ANEXO II - Diagramas dc Blocos

LT-04C4

27AA Al	FALHA ALIM 125Vcc PROTEÇÃO		
27FC 21	FALHA CONV 125Vcc PDS2000		**Código do evento**
85FC (60)	FALHA_TELEP_ CARRIER TRANSM	>= 1 [OU:2A]	74AP ANORM PROTEÇÃO(ANPR)
85FR (29)	FALHATELEP RECEPÇÃO		
S41 FP-FP	OSCILOGRAFO FIM PAPEL		
21_1 (49)	ATUAÇAO_TEMPO_ 1ª ZONA		
21_2 (50)	ATUAÇAO_TEMPO_ 2ª ZONA		
21_3 (51)	ATUAÇÃO_TEMPO_ 3ª ZONA	>= 1 [OU:IA]	•0100 94AP ATUAÇÃO PROTEÇÃO
21_4 (61)	ATUAÇAOTEMPO 4ª ZONA		
21DF (56)	DESLIGAMENTO FASE		
21DT (55)	DESLIGAMENTO TRIFÁSICO		
67N (59)	SOBRECOR_DIRECI NEUTRO		



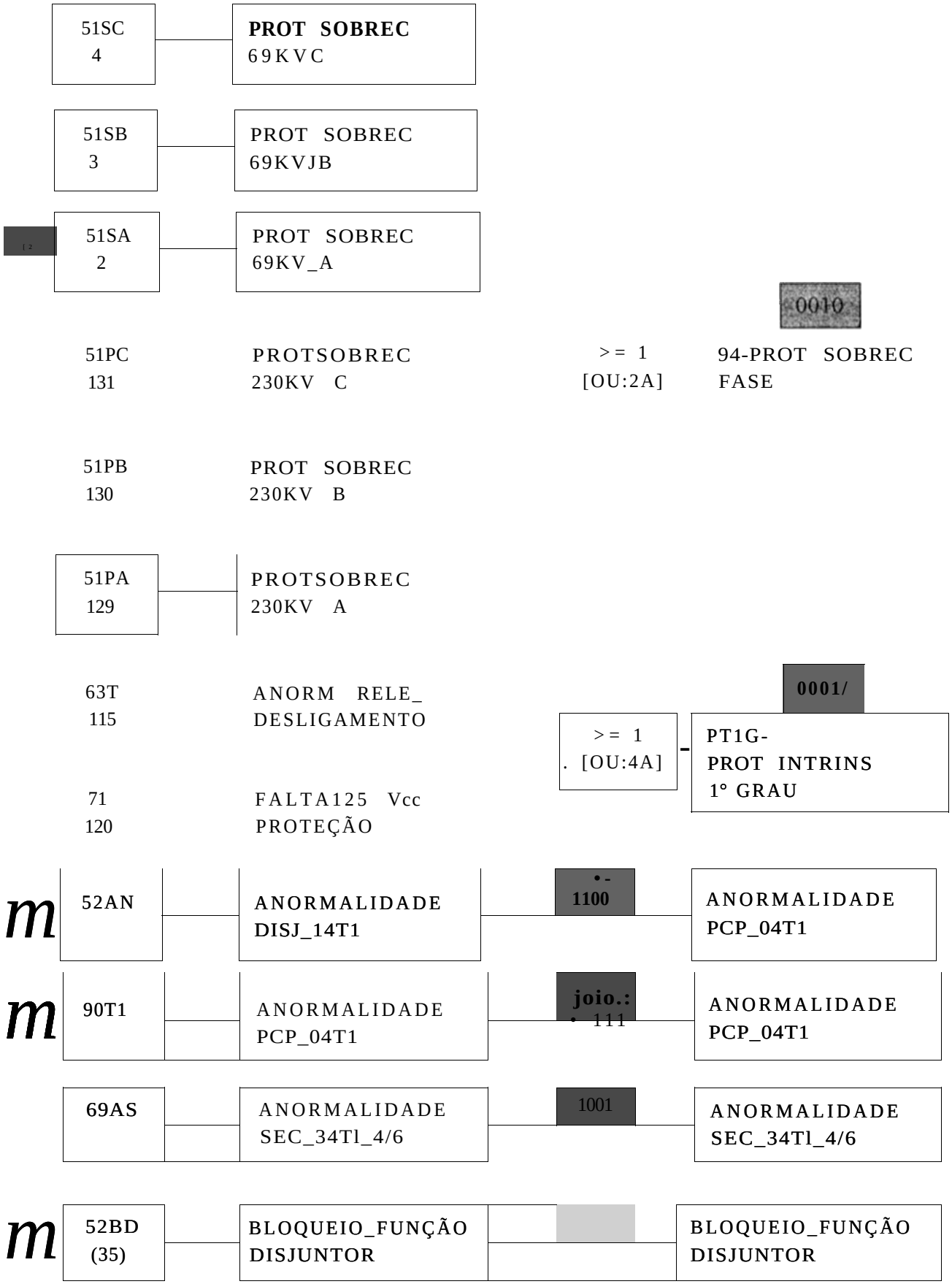
LT-04C4 (Continuação)



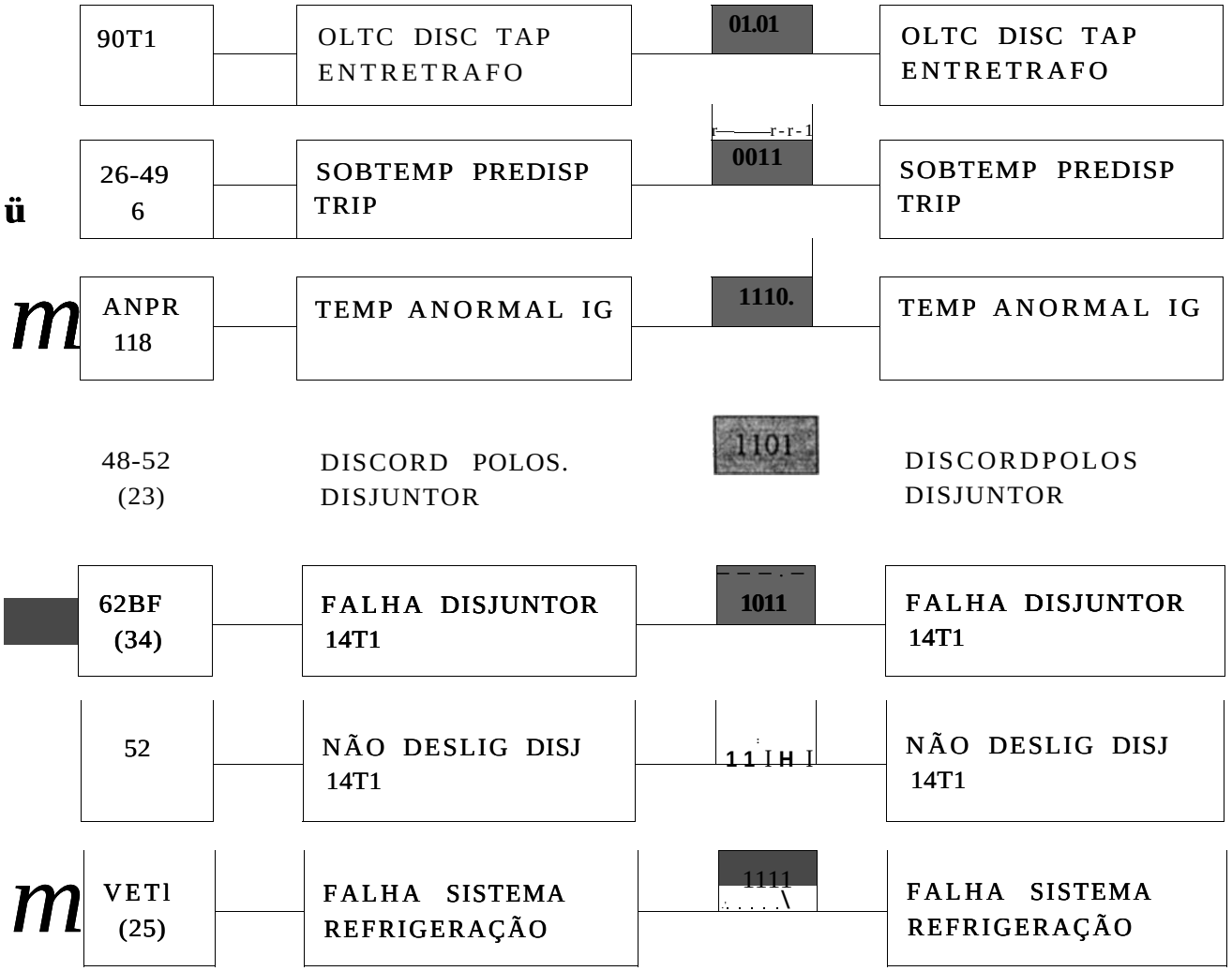
TR-04T1

				Código do evento
	86-94	ANORM RELE DESLIGAMENTO		1000
			[OU:3A]	72AP ANORM_ PROTEÇÃO (ANPR)
	27CC	FALTA 125 Vcc PROTEÇÃO		
3	51SN1	PROT SOBREC 69KV		
4	51PN 132	PROT SOBREC 230KV N		
	87A 126	PROT DIFERENCIAL FASE A		
				0100
	87B 127	PROTDIFERENCIAL FASE B	>= 1 [OU:IA]	86-94 DESLIG BLOQUEIO
	87C 128	PROT DIFERENCIAL FASE C		
4	63C 117	PROT_GAS_COMUT 2G		
4	63T 116	PROTGASTRAFO 2G/VS		

TK-04T1 (continuação)



TK-04T1 (continuação)



LT-04F3

59SN		CHTESTEPROT 591T TESTE
21CT		CHTESTEPROT 7SL32 TESTE
79DF		DEFEITO FALHA 79

Código do evento

1000

LDDF	DEFEITOFALHA LOCALIZ DEF	>= 1 [OU:2A]	74 AP-ANORMAL PROTEÇÃO

59DF		DEFEITO_FALHA_ PROT59IT
21DF		DEFEITO FALHA PROT 7SL32
52FA		DISJ FAL 125Vcc CONT PROT

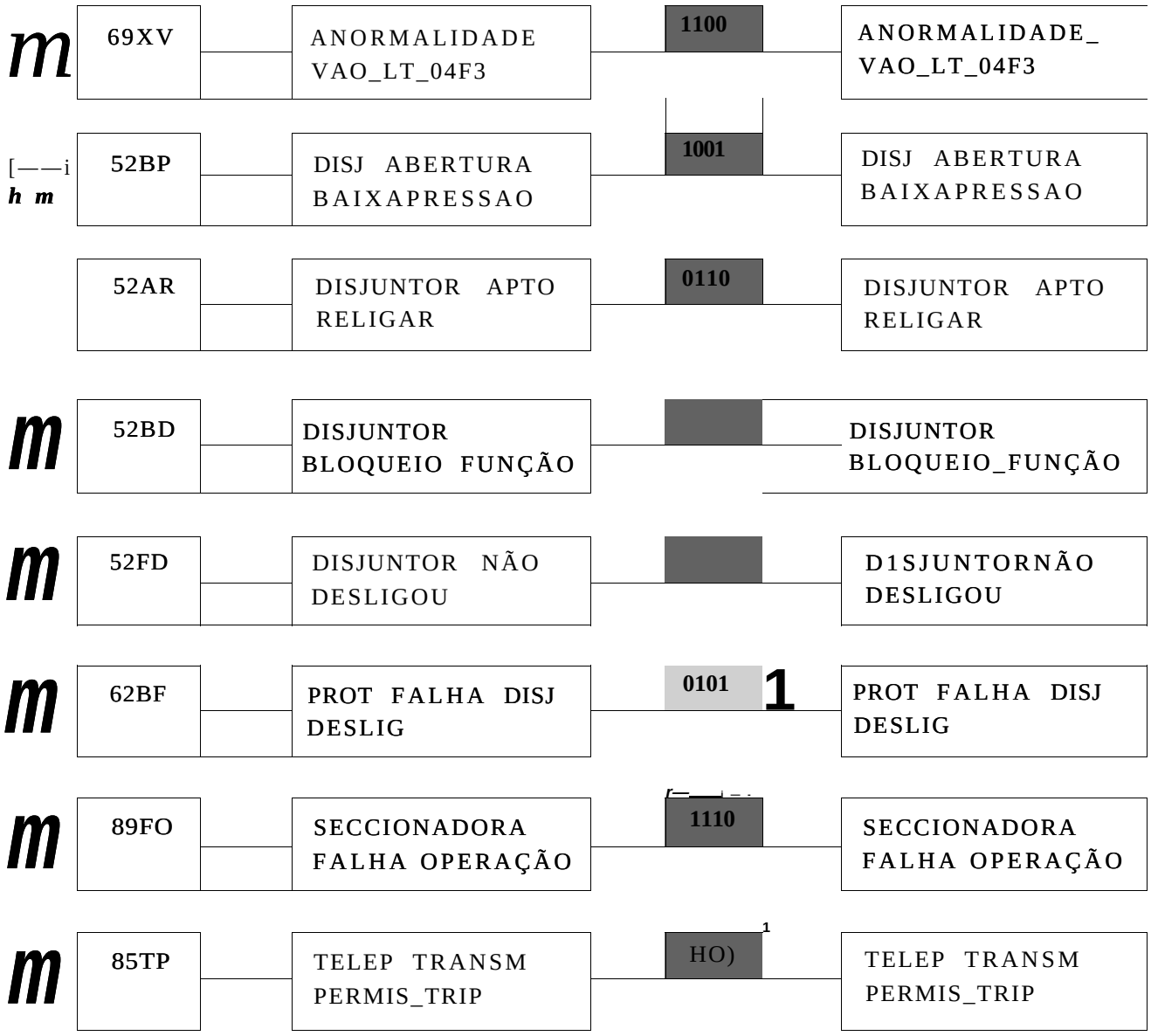
AL85		DISPARA MICRO DISJ_F2_85TT
85FR		FALHA TELEP RECEPÇÃO
85FT		FALHA TELEP TRANSMISSÃO
27XP		FALTA ALIM 125Vcc PROT
TPD1		FALTA ALIMTP DELTAABERTO
S41 FP-FP		OSCILÓGRAFO FIM PAPEL

LT-04F3 (Continuação)

21-2		ATUAÇÃO TEMPO 2ª ZONA
21-3		ATUAÇÃO TEMPO 3ª ZONA
21-4		ATUAÇAOTEMPO 4ª ZONA

68D	DESLIG_OSClLAÇAO POTÊNCIA	>= 1 [OU:IA]	94APATUAÇÃO PROTEÇÃO
21DL	DESLIGAMENTO DEAD LINE		
21DF	DESLIGAMENTO FASE		
21DR	DESLIGAMENTO, ZONA REVERSA		
67N	SOBCOR_DIRECI. NEUTRO		
591T	SOBTENS_INST_ TEMPOR DESLIG		
52CS	DISJ_ChSelec_LOC MANUTENÇÃO	>= 1 [OU:3A]	0010 52BFDISJANORM ChLocManut
52AC	DISJ_ANORM_SF6 MOTOR		

LT-04F3 (Continuação)



LT-02J1

				Código do evento
51SF		PROTEÇÃO SOBREC FASE TEMP		10
			[OU:3]	51SF-PROT SOBREC FASE
50SI		PROTEÇÃO SOBREC FASE INST		
5INT		PROT SOBREC_ NEUTRO TEMP		,0,1
			> = 1 [OU:4]	50SN-PROT SOBREC NEUTRO.
J 50NI		PROT SOBREC NEUTRO INST		

ANEXO III - Programas de Atualização dos Pesos

I. Módulo LT-02.II

detf

```
%
m dlmreadCe:\milde\al.tst7 ',0,0,1 • 2 4 10));
s dlmrcad(c:\mildc\h02jl.txt7 ',0,0,(12410]);
y-sizc(s);
y y(i)-i;
z-sprinlíVNúnicro máximo dc exemplos: %3.()í',y);
disp(z);
alann s(2:yH,l:4);
alarm alarm';
inpul(1'lntrc o número do exemplo a ser avaliado: ');
n ans;
ifn <= y

   fbri 1:2.
    p(i.n)- 0;
   end
%
% preparacílo do vetor peso
%
   for i - 1:4,
    if alarm(i,n)== 1
     ifi <=2

     end
     if i > = 3 & i < = 4
       P(2JI)=1;
     end
    end
   end

  alarmes alarni(:,n:n)
  RN=m(n»-131+1,2:2:5)
  p-p(:ji:n);
%
% Sistema dc inferência fuzzy
%
  a-rcadlisClt()2jr).
  a(36:37,4)-p;
  writcfis(a.H02jr);
pausc(0);
  x-cvalfis(RN,a);
  a-VintlCEVKNTOS" %6.4r.x);
  disp(z);
%
% Saída linguistica
%
  if x<» 1
     dispC                                    [PROT_SOBRF.C FAS1-)")
  end
  if"x>- 1 & x<=2
     dispC                                    IPROT SOHRKC N1HJTRO]')
  end
%
cise
 error('() número do exemplo a ser avaliado excede o número máximo dc exemplos')
end
%
end
```

II. Módulo LT-04C4

```
clear
%
nr dlmread('e:\milde\all tstV ',0,0,1 • 2 8 600));
s  dlmrcad('c:\mildc\lt04c4.txtV  ',0,0,(1 2 19 600]);
y=sizc(s);
y=y(l)-l;
/. sprinlft/Númcro máximo de exemplos: %3.()P,y);
disp(/);
alarm=s(2:yt 1,1:19);
alarm-alarm';
inputCKntrc o número do exemplo a ser avaliado: ');
n-ana;
if n <= y

   fori = 1:8,
    p(i,n) = 0;
   end
%
% preparação do vetor peso
%
   fori - 1:19,
    ifalarm(i,n)    1
     if i <= 5
      pOn)-l;
     end
     if i>=6&i<= 13
      P(2,n)=l;
     end
     ifi — 14
      P(3JI)-1;
     end
     ifi== 15
      P(4,n)= 1;
     aid
     ifi== 16
      P(5ji)=l;
     end
     ifi — 17
      p(6ji)=l;
     end
     ifi — 18
      P(7JI)=1;
     end
     ifi == 19
      P(8JI)= 1;
     end
    end
   end

  alarmes-alarm(:,n:n)
  RN=m(n+l:ni 1,2:2:9)
  p-p(:ji:n);
%
% Sistema de inferência fuzzy
%
  a-readfisClt04c4');
  a(70:77,6)=p;
  writcfis(a,'ll04c4');
pausc(0);
  x-evallis(RN,a);
  z-sprintlíT.VF.NTOS-%6.4f,x);
  disp(z);
%
% Saída linguística
%
  ifx<= 1
    dispC                                   IANORM PROTEÇÃO]')
  aid
  ifx>= 1 & x<=2
    dispC                                   [ATUAÇÃO, PROTEÇÃO]')
  end
```

```
ir x > - 2 & x < - 3
   dispC                                   IANORMAIJDADF. DISJI'NIOU|')
 aid
 ir x > = 3 & x < = 4
   dispC                                   [ANORMALIDADE SECCIONADORA)')
 aid
 irx > - 4 & x < - 5
   disp('                                  [DISCORDÂNCIA DE P()IX)S]')
 end
 if x > = 5 & x < - 6
   disp('                                  [PROTEÇÃO FALHA DISJ DESIJG.f)
 end
 if x > = 6 & x < = 7
   dispC                                   [BLOQUEIO FALHA FUZÍVELJ')
 end
 ifx>=7
   dispC                                   IBLOQUEIO FUNÇÃO DISJUNTORf)
 aid
%
cise
 crror('0 número do exemplo I ser avaliado excede o número máximo de exemplos")
end
%
aid
```

III. Módulo LT-04I3

```
clear
%
m dlmrcad('e:\mildc\linha.lst',' \0,0.|I 2 8 6001);
s dlmrcad('c:\mildc\U04F2.txl,".0,0,|l 2 32 6001);
y sizc(s);
y-y(l)-l;
z-sprintrçKúmcro máximo de exemplos: °/o4.0f,y);
disp(z);
alarm s(2:yH.1:32);
alarm-alarm';
inpirtCFntrc o número do exemplo a ser avaliado ');
n a n s;
ifn <=y

   fori = 1:11.
    P(i.n) - 0;
   end
%
% preparação do vetor peso
%
   fori = 1:32,
    ifalarm(iji) -= 1
     ifi<= 13
      P(l.n)- I;
     end
     if i>~14 & i<=22
      p(2.n)   1;
     end
     ifi >= 23 & i <= 24
      p(3.n)   1;
     end
     ifi    25
      P(4,n)=l;
     end
     ifi — 26
      p(5.n)   1;
     end
     ifi — 27
      p(6.n) = l;
     end
     ifi — 28
      p(7,n)-l;
     end
     ifi = 29
      P(8.n)- 1;
```

```
    end
    ifi    30
     P(9,n)- 1;
    end
    ili    31
     p(10,n) = 1;
    aid
    if i — 32
     P(ll,n)- 1;
    aid
   end
  aid
  fork-1:11,
   ifp(k,n)== 1 & k<=9
   alarnics-alarm(:ji:n)
   RN-m(ntl:n» 1,2:2:9)
   I» p(l:9ji:n);
%
% Sistema de inferência fuzzy
%
   a=readfisOIK)4nV);
   a(73:81,6)-P;
   writcfisia/lKMO');
pausc(0);
   x=evaIfis(RN,a);
   z  sprintfO- VKNTOS- %6.4f,x);
   disp(z);
%
% Saída linguística
%
   ifx<= 1
     dispC                                    (ANORM  PROTEÇÃO)')
   end
   ifx>= 1 & x<=2
     disp('                                   [ATUAÇÃO PROTEÇÃO)')
   end
   irx>=2 & x<=3
     dispC                                    IDISJ  ANORM  Chl.ocManut|')
   end
   if x>=3 & x<=4
     disp('                                   [ANORMAUDADF VÃOJ.T 04F3)')
   end
   if x>=4 & x<=5
     disp('                                   |I)ISJ  ABERTURA  BAIXA  PRESSÃOI')
   end
   if x>=5 & x<=6
     dispC                                    |I)ISJUNTOR  APTO  RFI.IGAR|')
   end
   if x>=6 & x<=7
     dispC                                    IDISJUNTOR  BI.OQUFIO  FUNÇÃO)')
   aid
   if x>=7 & x<=8
     disp('                                   [DISJUNTOR  NÃO  DESLIOOUj")
   end
   if x>=8 & x<=9
     dispC                                    IPROT  FAIJIA  DISJ  DESFIO)')
   end
   clscifp(kji) — 1 &k >= 10
    alarmes  alarm(:.n:n)
    RN-m(n+1:n+l,2:2:9)
    I' p(10:llji:n);
%
% Sistema de infaência fuzzy
%
     a-rcadfis(•|t04Oc•)•,
     a(52:53,6)=P;
     writcfis(a,'lt04Dc');
pausc(0)
     x~cvalfis(RN,a);
     z=sprintfCEVENTOS = %6.4P,x),
     disp(z);
%
%saída linguística
%
```

```
if x - O & x       I
       dispC                                          |SKCCI()NAIK)KA  FAI.HA  OPERAÇÃO|*)
      end
      if x    1
       dispC                                          ITELEP  TRANSM  PERMIS  TRIP|*)
      end
     cise
     end
  end

cise
 crror('() número do exemplo a ser avaliado excede o número máximo de exemplos')
end
%
end
```

IV. Módulo TR-04T1

```
clear
%
m  dlmrcad('c:\mildc\trafo.tslV ',0,0,|1  2  8  6()0|);
s dlmrcad('c:\mildc\lr04ll.txlV  ',0.0.1' 2 28f,00|).
y-sÍ7.e(s);
y-y(i)-i;
z^sprinlííTvJúmcro máximo de exemplos: %3.()P,y);
disp(z);
alarm-s(2:yt  1.1:28).
alarm  alarm';
inpirtCKntre o número do exemplo a ser avaliado  ');
n~ans;
if n <• y


   fori  -  1:15.
    p(i.n)    0.
   end
%
% preparação do vetor peso
%
   for  i  -  1:28.
    if alarm(i.n)  =  1
     if i <— 2
     P(lji)    1;
     end
     if i > = 3 & i < = 9
      P(2.n)=l;
     end
     ifi>=  10 & i<-  15
      POJ)  -  I;
     end
     ifi  >-  16&i<=  17
      p(4.n)    1;
     end
     ifi —  18
      p(5.n)-  1;
     end
     ifi —  19
      p(6,n)=  1;
     end
     ifi —  20
      P(7.n)-1;
     end
     ifi-=21
      P(8.n)=  1;
     end
     ifi   —22
      P(9.n)-  1.
     end
     ifi == 23
      P(10,n)=l;
```

```
    aid
    ifi    24
     p(11.n)   I;
    aid
    ifi=-25
     P(12,n)   1;
    aid
    ifi    26
     P(13.n)   1.
    aid
    ifi=- 27
     p(14.n)   1,
    aid
    ifi = 28
     P(15,n)- 1;
    aid
   aid
  aid
   fork- 1:13,
    ifp(k.n) = 1 & k <= 9
     alarmcs-alarm(:,n:n)
     RN-mín+ln «1,2:2:9)
     P-p(1.9.n:n);
%
% Sistema de infaência RIZA
»/o
    a-rcadrisCtr04tl').
    a(73:81,6)-P;
    writens(a.1r<)4tr);
pausc(0);
     x-cvalfis(RN.a);
     zsprintlíi:Vl-NT()S %6.4P,x);
     disp(z);
%
% Saída linguistica
»/o
    ifx<= 1
      dispC                                    {ANORM   PROTEÇÃO]*)
    end
   ifx>= 1 Ã x <= 2
      dispC                                    [DESLIG BLOQUEIO]*)
    end
    ifx>=2 & x<=3
      disp('                                   [PROT SOBREC FASE]*)
    end
    ifx>=3 & x<=4
      dispC                                    [PROT INTRINS l-XiRAUI')
    aid
    ifx>=4 & x<=5
      dispC                                    [ANORMALIDADE PCP 04T11')
    end
    ifx>=5 & x<=6
      dispC                                    [ANORMALIDADE 1'CP 04T11')
    aid
    ifx>=6 & x<=7
      disp('                                   [ANORMALIDADES EC 34T1 4/6]*)
    aid
    ifx>=7 & x<=8
      dispC                                    [BLOQUEIO FUNÇÃO DISJUNTOR|')
    end
    ifx>=8
      disp('                                   IOI.TC DISC TAP ENTRE TRAFO)')
   end
  ciseifp(kji) == 1 & k >- 10
   alarmes alarm(:ji:n)
   RN=m(n+l:n•1,2:2:9)
   P=p(10:15ji:n);
%
% Sistema de inferência fuzzv
%
   a-read^ls(tr04tlc);
   a(64:69,6)-P;
   writcfis(a,tr()4t1c'),
pause(0);
```

```
    x  eva1fía(RN,a);
    /  sprint ffKVKNTOS   %6.4f,x);
    disp(z);
%
'o Saida linguistica
»b
    if x<- I
      dispC                                          [SOBTEMP PREDISP TRIP]*)
     aid
    if x>=   l & X < - 2
      dispC                                          [TEMP ANORMAL IO]')
     end

     if x > = 2 & x < = 3
       dispC                                         [DISCORD  POLOS  DISJUNTOR)')
     end
     if x > = 3 & x < - 4
       dispC                                         [FALHA  DISJIINTOR  14T1)')
     end
     if x > = 4 & x < = 5
       dispC                                         |NÂO  DLSLKi  DISJ  14TI)')
     end
     if x > = 5 & x      6
       dispC                                         [FALHA  SISTEMA  REFRIOERAÇÂO]*)
     end
    else
     aid
   end
%
else
  CTTor('0 número do exemplo a $CT avaliado excedo o número máximo de exemplos')
aid
%
aid
```

GLOSSÁRIO

Agregação - A combinação do conseqüente de cada regra em um sistema fuzzy de inferência tipo Mamdani na preparação da defuzifícação.

Antecedente - A parte inicial ("se") da regra fuzzy.

Conjunto fuzzy - um conjunto que pode conter elementos com um certo grau de pertinência parcial.

Conseqüente - A parte final ("então") da regra fuzzy.

Defuzifícação - O processo de transformar uma saída fuzzy de um sistema fuzzy de inferência em uma saída numérica.

Função Membro - A função que especifica o grau com que uma dada entrada pertence a um conjunto fuzzy.

Fuzifícação - O processo de gerar valores membros para a variável fuzzy utilizando as funções membros.

Fuzzy - Nebuloso, indistinto, não claro, obscuro.

Grau de pertinência - A saída de uma função membro cujo valor está sempre limitado entre 0 e 1. É também conhecido como valor membro.

Implicação - O processo de modular o conjunto fuzzy no conseqüente com base nos resultados do antecedente em um sistema fuzzy de inferência tipo Mamdani

Inferência tipo Mamdani - Um tipo de inferência fuzzy no qual os conjuntos fuzzy do conseqüente de cada regra são combinados pelo operador agregação e o resultado do conjunto fuzzy é defuzificado para produzir a saída do sistema.

Operadores fuzzy - Operadores E, OU e NÃO. São também conhecidos como conexões lógicas.

CAPÍTULO 1

INTRODUÇÃO

A energia elétrica é considerada um dos insumos básicos mais importantes na evolução da sociedade moderna. De fato, o consumo per capita de energia está diretamente associado ao nível de desenvolvimento econômico e social de uma população. As taxas de crescimento da demanda de energia elétrica têm apresentado valores crescentes ao longo de várias décadas, à medida que a população vai tendo acesso ao conforto oferecido pelas tecnologias atuais. O padrão de instalação de centrais elétricas tem consequentemente crescido, ocasionando- sistemas de grande porte e de alta complexidade.

A finalidade de um sistema elétrico de potência é distribuir energia elétrica para uma multiplicidade de pontos, para diversas aplicações. Tal sistema deve ser projetado e operado para entregar esta energia obedecendo dois requisitos básicos: qualidade e economia.

A garantia de fornecimento da energia pode ser aumentada melhorando o projeto, prevendo uma margem de capacidade de reserva e planejando circuitos alternativos para o

suprimento. A subdivisão do sistema em zonas, cada uma controlada por um conjunto de equipamentos de chaveamento, em associação, proporciona flexibilidade operativa e garante a minimização das interrupções.

A maior ameaça para o fornecimento da energia elétrica é a ocorrência de curtos-circuitos (ou faltas) nos componentes do sistema, que impõem mudanças bruscas e violentas na operação normal. O fluxo de uma elevada potência com uma liberação localizada de uma considerável quantidade de energia pode provocar danos de grande monta nas instalações e equipamentos. Considerando-se um componente isoladamente, o risco da ocorrência de uma falta é pequeno, entretanto, globalmente pode ser bastante elevado, aumentando também a repercussão numa área considerável do sistema.

Um rápido isolamento do componente submetidos a uma falta eliminará a possibilidade de danos ou, na pior das hipóteses, minimizá-lo. Porém, quando ocorre um distúrbio, os eventos isolados apresentam-se em quantidade tal que dificultam ao operador o reconhecimento da causa destes distúrbios (diagnose) e o estabelecimento de ações corretivas a serem executadas.

Para tentar resolver esses problemas, vários trabalhos vêm sendo desenvolvidos. Em [1], uma rede neural é aplicada na montagem de um processador inteligente capaz de interpretar múltiplos alarmes e fornecer informações ao operador do sistema. [2] aplica um Sistema Híbrido, integrando Lógica Nebulosa e Redes Neurais, para diagnóstico de falhas a partir do processo de alarmes originados do sistema elétrico. Em [3], uma rede neural treinada para reconhecer os padrões de falhas em linhas de transmissão é implementada em um PC que analisa os arquivos de dados dos registradores digitais de falhas (Digital Fault Recorder -DFR). Assim que o DFR registra um distúrbio, os dados são automaticamente transferidos para o PC e o "software" analisador de falhas

determina se ocorreu falhas e, em caso positivo, o tipo de falha. Em [4], um Sistema Inteligente Híbrido, Redes Neurais e Sistemas Especialista, é aplicado na detecção e diagnóstico de falhas em redes elétricas. O sistema especialista verifica as causas prováveis, e finalmente, de forma lingüística, sugere as ações corretivas a serem tomadas.

Verifica-se que os trabalhos citados utilizam técnicas recentes como Redes Neurais e ainda, outra parte utiliza técnicas mais tradicionais, como sistemas baseados em Lógicas/Regras ou sistemas que utilizam duas ou mais destas Técnicas de Inteligência Artificial (IA) - Sistemas Híbridos - visando alcançar as vantagens de cada uma delas e reduzir suas desvantagens.

Embora os trabalhos anteriores tenham alcançado relativo êxito no campo do diagnóstico de falhas, não existe uma forma acessível de utilizá-los em aplicações similares quando se deseja abordar este tipo de análise por meio de técnicas de IA. Desta forma, é necessário obter os simuladores adequados, familiarizar-se com seus métodos, conjuntos de entrada e saída, e finalmente, implementar o problema específico nestes simuladores, realizando o seu acoplamento quando necessário.

1.1-OBJETIVO

Neste trabalho, um Sistema Híbrido é desenvolvido e aplicado ao estudo da filtragem de sinais de alarmes de proteção da subestação da Mirueira, pertencente ao sistema da Companhia Hidroelétrica do São Francisco (CHESF). O Sistema de Supervisão Local daquela subestação está sendo automatizado e seus sinais de alarmes serão enviados para o Centro Regional de Operação

Leste (CROL), com a finalidade de ser supervisionada e controlada remotamente.

Com o intuito de reduzir a quantidade de alarmes a ser transmitida ao CROL/CHESF, o qual deverá abrigar não apenas os alarmes daquela subestação, mas futuramente os alarmes de todas suas outras a serem contempladas pelo projeto de teleassistência, será realizada uma filtragem dos alarmes da SE Mirueira. Quando essa filtragem é realizada através de "hardware", o sistema apresenta-se pouco flexível e quaisquer modificações no sistema de proteção resultam em alterações físicas no mesmo. Nesse caso, espera-se que o custo operacional seja elevado.

A utilização do Sistema Híbrido para realizar esta função de filtragem é bem mais vantajosa em virtude da necessidade de alterar apenas o programa controlador.

O objetivo principal deste trabalho é demonstrar a viabilidade e o bom desempenho desses sistemas para executar função de filtragem.

Para atingir este objetivo, foi montada e treinada uma Rede Neural (RN) adequada e testado seu desempenho na tarefa de classificar eventos ocorridos na subestação da Mirueira. Em seguida, foi desenvolvido e implementado um Sistema de Inferência Fuzzy (SIF) acoplado à RN e analisado o desempenho global do Sistema Híbrido (RN-SIF) em relação aos mesmos eventos anteriores.

1.2 - ORGANIZAÇÃO DO TRABALHO

O trabalho é desenvolvido em 6 capítulos:

Capítulo 2: Descreve a finalidade básica da proteção e sua importância dentro do Sistema Elétrico, apresenta os conceitos de automação e teleassistência de subestações, e finalmente, aborda as principais funções do Sistema de Supervisão e Controle do Sistema de Potência segundo os níveis hierárquicos de ação.

Capítulo 3: Faz um breve relato da origem das Redes Neurais, expõe sua técnica e algoritmo. Além disso, destaca suas vantagens e desvantagens no diagnóstico de falhas.

Capítulo 4: Apresenta uma breve revisão teórica da Lógica Fuzzy, mostra a técnica por ela utilizada e descreve detalhadamente as etapas realizadas pelo Processo de Inferência Fuzzy. Para melhor compreensão deste capítulo, em virtude das palavras estrangeiras e dos termos utilizados nas etapas do processo de inferência, recomenda-se a leitura do glossário e do apêndice B.

Capítulo 5: Este capítulo apresenta a finalidade das duas técnicas de inteligência artificial abordadas nos capítulos 3 e 4; mostra o desenvolvimento da Rede Neural através do Simulador de Redes Neurais (SiReNe) e os processos utilizados para o seu treinamento; apresenta detalhadamente a concepção do Sistema de Inferência Fuzzy através do "Fuzzy Logic Toolbox" do MatLab; e finalmente, aborda o processo de conexão dos dois sistemas inteligentes por meio de um de programa de atualização de pesos, desenvolvido na plataforma do MatLab.

Capítulo 6: Relata os principais resultados obtidos e procede às discussões dos tópicos mais relevantes extraídos deste trabalho.

Capítulo 7: Finalmente, este capítulo resume as conclusões do presente trabalho e apresenta sugestões e aperfeiçoamentos para trabalhos futuros.

CAPÍTULO 2

PROTEÇÃO NOS SISTEMAS ELÉTRICOS

2.1 - INTRODUÇÃO

A finalidade do sistema elétrico de potência é suprir continuamente de energia seus usuários para diversas aplicações. O sistema deve ser projetado e operado para entregar esta energia obedecendo dois requisitos principais: qualidade e economia.

A finalidade básica da proteção é minimizar os efeitos dos grandes problemas que surgem quando da operação do sistema, ou seja, minimizar:

a) o custo nos reparos dos danos causado aos equipamentos pelas anormalidades;

b) o número de equipamentos envolvidos;

c) o tempo que o componente fica fora de serviço,

d) os aspectos sociais da interrupção ao fornecimento.

É fácil observar que os sistemas de proteção não impedem o aparecimento de anormalidades no sistema elétrico, embora tenham a importante finalidade de evitar ou minimizar os reflexos destas perturbações.

2.2 - ANÁLISE DE FALTAS

A análise de faltas é imprescindível tanto no planejamento como na operação de um sistema elétrico de potência. Muitas vezes são tomadas decisões, não só técnicas como econômicas, baseadas nos resultados dessa análise.

Faltas em sistema de potência são conseqüências de distúrbios em operação ou aparecimento de falhas em qualquer um de seus componentes.

A forma mais comum de distúrbio em operação é a sobrecarga, isto é, uma corrente em um determinado equipamento que excede o seu valor nominal.

O tipo de falha mais comum e também o mais severo é a falta decorrente do contato elétrico direto ou através de arco entre partes sob potenciais diferentes ou de uma ou mais partes para a terra, num sistema ou equipamento elétrico energizado. A amplitude da corrente de falta depende de vários fatores, tais como: tipo de falta, capacidade do sistema de geração, topologia da rede elétrica, tipo de aterramento do neutro dos equipamentos, conexão dos transformadores de potência, distância elétrica da falta em relação às unidades geradoras, etc.

A corrente de falta excede consideravelmente a corrente nominal do equipamento afetado, podendo provocar danos materiais e perturbações ao sistema de potência.

Na ocorrência da falta é necessário que a parte atingida seja isolada, do restante do sistema, tão rapidamente quanto possível para evitar danos materiais e o envolvimento das partes intactas do sistema de proteção.

O risco de uma ocorrência de falta considerando-se um componente separadamente é muito pequeno, entretanto, considerando-se globalmente o sistema, este risco pode ser bastante elevado, aumentando também a repercussão em todo o sistema. Portanto, caso não se disponha de equipamentos apropriados para detectar e interromper estas faltas, a operação do sistema tornar-se-á inviável. Estas considerações mostram a importância de Sistemas de Proteção.

2.3 - IDÉIA BÁSICA DE SISTEMA DE PROTEÇÃO

Sistema de proteção consiste em um conjunto de relés que fica constantemente comparando os dados do sistema de potência com um valor de referência previamente estabelecido (ajuste). A fig.2.**1** mostra esta idéia básica [5].

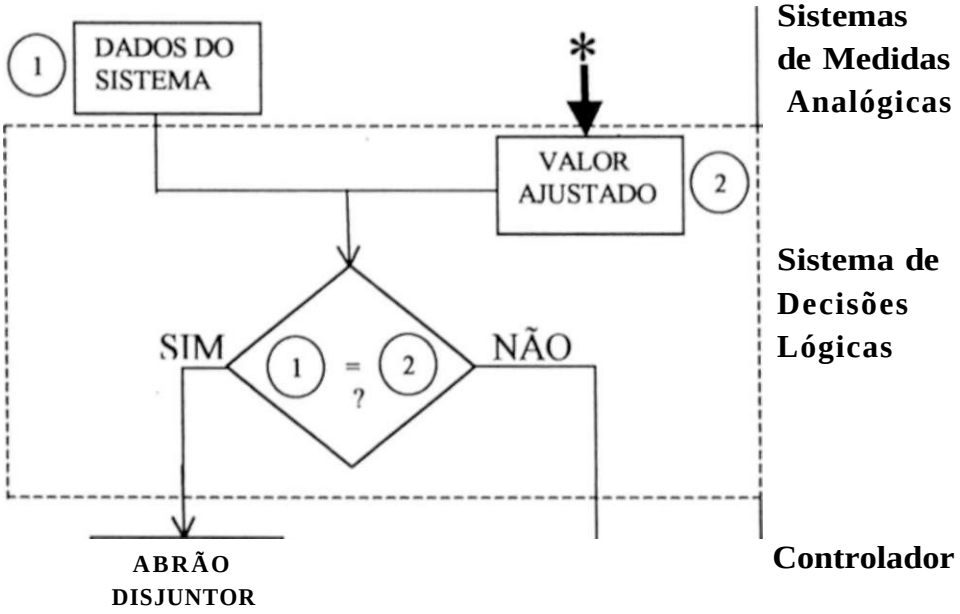


Fig. 2.**1** Diagrama de bloco da idéia básica de Sistema de Proteção.

*

Intervenção externa
Exige o conhecimento prévio de:
Sistema Elétrico (parâmetros, configuração, níveis de curtos-circuitos);
Tipos de proteção (esquemas, relés)

2.4 - CONJUNTO COERENTE DE PROTEÇÃO

A utilização de um conjunto coerente de proteções é um dos requisitos, dentre outros, necessário para atenuar os efeitos das perturbações que afetam as redes elétricas e seus órgãos de controle. O sistema de proteção deve:

a) assegurar, o melhor possível, a continuidade de alimentação dos usuários,

b) salvaguardar o material e as instalações da rede.

No cumprimento dessas missões ele deve. Tanto alertar os operadores em caso de perigo não imediato, como retirar de serviço a instalação se há, por exemplo, um curto-circuito que arriscaria deteriorar um equipamento ou afetar toda a rede.

Verifica-se, assim, que há necessidade de dispositivos de proteção distintos para:

a) as situações anormais de funcionamento do conjunto interconectado ou de elementos isolados da rede (perdas de sincronismo, por exemplo);

b) os curtos-circuitos e os defeitos de isolamento.

2.5 - ANÁLISE GENERALIZADA DA PROTEÇÃO

Basicamente, em um sistema elétrico encontram-se os seguintes tipos de proteção:

a) proteção contra incêndio;

b) proteção pelos relés (releamento) e por fusíveis;

c) proteção contra descargas atmosféricas e surtos de manobras.

E importante destacar que o releamento minimiza:

a) custo de reparação dos estragos;

b) a probabilidade de que o defeito possa propagar-se e envolver outro equipamento;

c) tempo que o equipamento fica inativo, reduzindo a necessidade das reservas,

d) a perda de renda e o agastamento das relações públicas, enquanto o equipamento está fora de serviço.

A proteção por meio de relés tem duas funções:

a) função principal - que é a de promover uma rápida retirada de serviço de um elemento do sistema, quando esse sofre um curto-circuito, ou quando ele começa a operar de modo anormal que possa causar danos ou, de outro modo, interferir com a correta operação do resto do sistema. Nessa função um relê (elemento detector e comparador) é auxiliado pelo disjuntor (interruptor), ou então, um fusível engloba as duas funções;

b) função secundária - promovendo a indicação da localização e do tipo do defeito, visando mais rápida reparação e possibilidade de análise da eficiência; e características de mitigação da proteção adotada.

Dentro dessa idéia geral, os chamados *princípios fundamentais do releamenío* compreendem:

releamento primário ou de primeira linha;

releamento de retaguarda ou de socorro;

releamento auxiliar.

a) O releamento primário é aquele em que uma zona é estabelecida ao redor de cada elemento do sistema, com vistas à seletividade. Para se alcançar a seletividade, disjuntores são colocados na conexão de cada dois elementos, criando assim, uma

superposição das zonas de proteção em torno dos mesmos, a qual possibilita o socorro em caso de falha da proteção principal;

b) releamento de retaguarda, cuja finalidade, é a de substituir o releamento primário durante a manutenção ou na falha deste. Só é usado, por motivos econômicos, para determinados elementos do circuito e contra curto-circuito. No entanto, sua previsão deve-se à probabilidade de ocorrer falhas, seja na corrente ou tensão fornecida ao relê, na fonte de corrente de acionamento do disjuntor; no circuito de disparo, no mecanismo do disjuntor; ou no próprio relê, etc. Nestas condições, é desejável que o releamento de retaguarda seja instalado independentemente das possíveis razões de falha do releamento primário. Uma observação importante é que o releamento de retaguarda não substitui uma boa manutenção ou vice-versa;

c) releamento auxiliar tem função como multiplicador de contatos, sinalização ou temporizador, etc.

2.6 - AUTOMAÇÃO E TELEASSISTÊNCIA DE SUBESTAÇÕES

As subestações convencionais possuem vários tipos de automação, podendo-se destacar algumas delas como o comando de disjuntores e chaves a partir da sala de controle, e os intertravamentos na operação de equipamentos, funções estas providas via relés eletromecânicos e lógica de contatos.

Com o advento dos sistemas digitais, estas funções e outras são realizadas de forma diferente utilizando equipamentos como microprocessadores e lógica estabelecida em "software". Não significa que os sistemas digitais trouxeram novas funções, mas modificaram, principalmente, a forma de fazê-las.

Desta forma pode-se estabelecer as funções dentro de uma subestação, de forma genérica servindo para implementação através do sistema digital, convencional ou misto.

Estas funções são :

a) Sinalização ou monitoração de estado (status) de equipamentos;

b) Medição;

c) Proteções:

- de linha
- de transformador
- de barra

de reator

por perda de sincronismo, etc;

d) monitoração das proteções;

e) religamento automático;

f) estimativas de localização de falta na linha;

g) telecomando;

h) proteção por falha de disjuntor;

i) controle de equipamentos de chaveamento (intertravamento);

j) seqüência automática de chaveamentos;

k) monitoração de sobrecarga em transformadores;

1) controle local de tensão e fluxo de reativo;

m) corte seletivo de cargas (load shedding)

n) sincronização,

o) alarmes;

p) indicação e registro de seqüência de eventos;

q) oscilografia;

r) interface homem-máquina;

s) impressão de relatórios;

t) interface com COR/COS e outros sistemas;

u) autodiagnose.

O sistema digital, para a realização destas funções, pode variar de complexidade dependendo de como se pretende implementá-las, porém, sempre será composto de um sistema central, um sistema de aquisição de dados com ou sem dispositivos de interface com o processo, e por unidades dedicadas com objetivos específicos (proteção, oscilografia, por exemplo).

A complexidade do sistema central é função do tamanho da subestação e da escolha se a subestação será assistida ou desassistida (telecomandada, assistida a distância).

A subestação é dita assistida quando tem operadores locais durante todo o tempo de serviço. E comum numa área haver várias subestações telecomandadas através de uma outra. Neste caso o sistema central, na subestação de comando, deve ser composto de computadores onde residirá a interface homem-máquina (IHM) para os operadores.

Já na subestação desassistida (telecomandada) o sistema central pode ser simples,

com apenas equipamentos para a comunicação e troca de dados com a subestação no comando

2.6.1 - FUNÇÕES

No item anterior foram listadas as funções numa subestação. Algumas das funções, listadas anteriormente, e seus requisitos serão descritos a seguir. Em [6] pode-se encontrar as demais funções não tratadas aqui, por não serem de importância para a finalidade a que se destina esta dissertação.

Proteção

Num sistema de automação de uma subestação moderna pode-se considerar a utilização de proteção digital ou convencional. Algumas empresas preferem usar a proteção convencional (já bastante testada) em algumas funções ou em todas. Em particular, várias empresas tem modernizado as suas subestações existentes, substituindo, dentre outras coisas, o comando e o controle convencional por digital. Entretanto, é comum neste caso manter a proteção convencional existente, porém, fazendo o sistema digital monitorá-la.

Em qualquer dos casos, digital ou convencional, utiliza-se para proteção, módulos dedicados e separados, sendo sua atuação feita diretamente no disjuntor sem passar pelos computadores do sistema digital, ou seja, a proteção atuando, fecha um contato que provoca a operação do disjuntor. Esta atuação da proteção é monitorada paralelamente.

Telecomando / Telecontrole

O comando e controle de abertura e fechamento de chaves e disjuntores, a modificação da referência de um regulador, a mudança de tap em um transformador com comutador sob carga, são

exemplos de ações que podem ser executadas em diferentes locais. Portanto, os comandos podem ser :

(a) locais (junto ao equipamento)

com comando mecânico;

com comando elétrico (utilizando motores, solenóides, etc);

b) a distância ou remotos

a partir da unidade de aquisição de dados (UAC);

- a partir da sala de comando da subestação;

a partir de outros centros (COS, COR, e centro de operação de subestações desassistidas).

Adicionalmente existem controles, como o de tensão pela alteração do tap de transformadores, que podem ser acionados manualmente através de teclas (botoeiras) de acionamento pelo operador, ou automaticamente por equipamentos sensores (relê de tensão, por exemplo).

Os seguintes comandos e controles aparecem numa subestação:

a) Operação de disjuntores;

b) Operação de seccionadoras e chaves;

c) Seleção de controle AUTO / MANUAL;

d) Seleção da localização LOCAL / REMOTO, em suas várias alternativas;

e) Bloqueio / desbloqueio de operação de disjuntores;

f) Bloqueio / desbloqueio de relés, inclusive religamento;

g) Movimentação do comutador sob carga;

h) Seleção de sincronização;

i) Transferência de proteção;

j) Valor de referência de reguladores e controladores locais.

Os comandos e controles podem ser originados simultaneamente de locais diferentes. A ordem a ser obedecida será definida pela posição de chaves seletoras instaladas nos vários locais. Assim, podem ser utilizadas para a priorização de obediência, as chaves de duas posições, LOCAL / DISTANTE, localizadas em diferentes pontos. A chave utilizada junto ao equipamento na posição LOCAL define que o comando (controle) só poderá ser iniciado mecanicamente ou eletricamente do painel do equipamento; na posição DISTANTE o comando passa a ser feito do nível hierárquico imediatamente superior, no caso a UAC. A chave de seleção na UAC na posição LOCAL define que o controle deve ocorrer via tecla instalada nesta UAC; na posição DISTANTE passa para a sala de comando e assim sucessivamente. A tabela abaixo mostra estas várias combinações:

Tab. 2.1 Combinações de comandos

Local da chave seletora	Posição	Local único de comando
Painel do equipamento	LOCAL DISTANTE	no painel na UAC
UAC	LOCAL DISTANTE	na UAC na sala de comando
Sala de comando	LOCAL DISTANTE	na sala de comando COS/COR/outros centros

Esta função vem associada com uma outra função, o intertravamento. Da ação conjunta destas duas funções, o comando não é realizado se certas condições (intertravamento) necessárias à segurança da operação não forem satisfeitas. O intertravamento visa estabelecer condições para manobra de seccionadoras.

Alarmes

Tanto as variáveis analógicas (correntes, tensões, temperaturas) como as digitais (atuação de relés, operação de disjuntores) podem ser usadas na função alarme.

Periodicamente, à medida que os dados analógicos vão sendo recebidos, a função alarme deve executar uma comparação para verificar se o valor medido está dentro de limites inferior e superior especificados. Antes de acionar um alarme, a variável deve ser tratada (filtragem digital, banda morta).

Dados digitais também ativam as funções alarmes como, por exemplo, a atuação de um relé.

Ao ser detectada uma condição de alarme, um evento deve ser sinalizado em memória, e/ou em disco, e/ou em impressora, e ser armazenado numa lista cronológica para indicação no vídeo.

Esta função deve promover mecanismos para apresentação e alteração de limites e criar lógicas de inibição e reconhecimento de alarmes.

Todas as mudanças de estado, quando provocadas pelo operador, podem ser consideradas como alarmes.

A função alarmes pode utilizar atributos de vídeo (cor, intensidade, campo piscante) e dispositivos sonoros para registrar, de forma clara, as transições de estado de um alarme (alarme não reconhecido, alarme reconhecido, fim de alarme sem reconhecimento, etc). Além disso, para evitar uma avalanche de alarmes em condições de emergência, normalmente é prevista a

implementação de alarmes condicionados e individuais.

Interface homem-máquina (MIM)

Esta função implementa a intenção entre o operador e o processo elétrico. Ela representa informações que descrevem o estado da subestação, do complexo informático, e permite que o operador interaja com esses ambientes, através de operações executadas via console de operações.

Tais operações são classificadas como segue:

a) apresentação de dados;

b) entrada de dados,

c) operações via console de operação;

d) funções de diagnóstico e manutenção.

A IHM permite ao operador obter as informações que lhe são úteis, tais como. lista de alarmes, valores de medições, estado de equipamentos, etc.

Interface com o COR/COS e outros sistemas

Quando uma subestação possui um sistema digital, ela acumula as informações em duas bases de dados: a de tempo real e a histórica.

Alguns dados em tempo real são necessários ao sistema de supervisão e controle da rede (COS/COR). Desta forma são previstos nos sistemas digitais meios para a intercomunicação destes dados. O sistema digital estará então substituindo a função das unidades de aquisição de dados nestes sistemas de supervisão.

Por exemplo, por este caminho de comunicação de dados, os sistemas de supervisão podem agir na subestação, ligando/desligando linhas e transformadores.

Esta função tem maior ou menor dificuldade de ser implementada dependendo do padrão de comunicação usado.

Se o padrão entre a automação da subestação (físico e lógico) e aquele dos outros centros são diferentes, é preciso incluir em um dos centros uma unidade tradutora de informações (gateways).

E pois importante que os sistemas digitais utilizem sistemas de comunicação que sejam padronizados.

2.6.2 - SISTEMA DE SUPERVISÃO E CONTROLE DO SISTEMA DE POTÊNCIA

Também denominado Sistema de Supervisão e Controle (SSC), Despacho de Carga ou Sistema de Gerenciamento (EMS : Energy Management System), provê os meios para coordenação da operação e da manutenção do sistema elétrico, isto visto de uma forma global.

O SSC é composto por vários níveis hierárquicos de ação:

UAC- Unidade de Aquisição de Dados e Controle

COR- Centro de Operação Regional

COS- Centro de Operação do Sistema

Nas UAC desenvolvem-se a aquisição de dados do processo e o comando de manobra de equipamentos. Neste nível encontra-se a interface com o processo.

Normalmente, são instaladas uma ou mais UAC para cada subestação. Os dados relativos a elas são comunicados ao COR via canal de telecomunicação (tipicamente a microondas). Apenas os dados mais significativos da subestação dizem respeito às atividades do SSC. Por exemplo: o estado dos disjuntores das linhas, geradores e transformadores, as potências ativas e reativas em cada elemento, e a tensão nos vários trechos da barra. Não interessa ao SSC, o estado dos disjuntores do serviço auxiliar da subestação.

No COR ocorrem a operação e o atendimento das subestações de uma região da área global. Dele partem, por exemplo, os sinais de telecomando dos disjuntores, os sinais para mudança de tap de um transformador, etc. e chegam todos os dados coletados nas UAC. No COR está localizado um sistema computacional com a interface homem-máquina (1HM) adequada ao operador da rede regional. A IHM permite a ele tomar o conhecimento dos alarmes, da seqüência de eventos, das medições, bem como executar telecomando. Em resumo reside no COR a função SCADA (Supervisory Control and Data Acquisition).

No COS, a operação global centralizada do sistema e a coordenação da geração e carga são facilitadas. Nele, onde encontram-se as funções denominadas de "alto nível", está localizado um sistema digital de onde são obtidas as informações necessárias a operação adequada e segura do sistema

No COS são ligadas muitas vezes as UAC correspondentes à malha principal do sistema. Portanto, nele pode estar também incluída a função SCADA.

Tanto no COR como no COS, os relatórios gerenciais e técnicos são gerados com facilidades. Na fig.2.2 é mostrado um SSC onde estão incorporados outros níveis de ação tais como:

COU- Unidade de Operação de conjuntos de Usinas

CAU- Centros de Atendimento de conjunto de Usinas

COS- Centros de Operação de conjuntos de Subestações

CAS- Centros de Atendimento de conjuntos de Subestações

COD- Centro de Operação da Distribuição

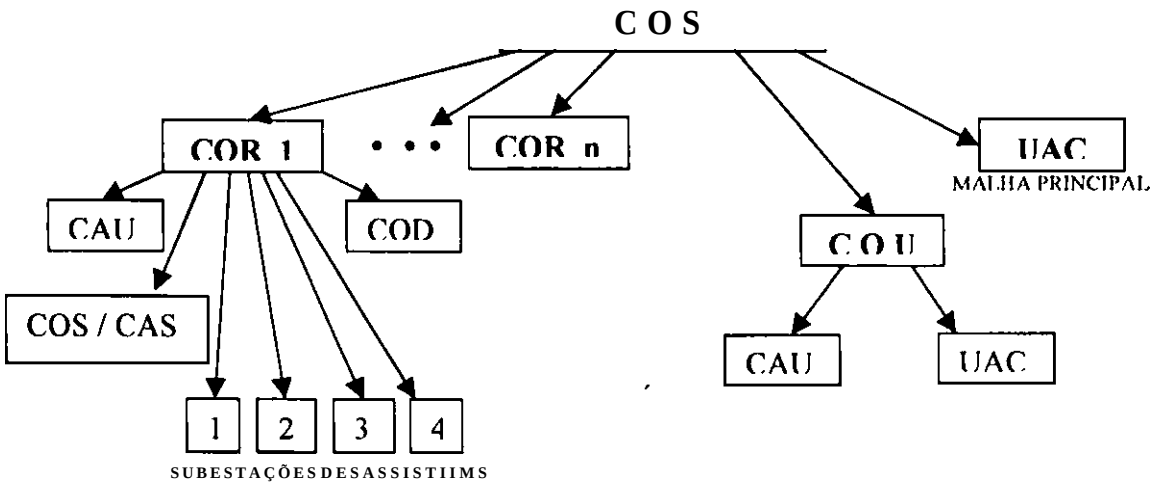


Fig.2.2 SSC com outras funções hierárquicas

2.6.3 - SISTEMA DE AQUISIÇÃO DE DADOS

O sistema de aquisição de dados é composto pelas unidades de aquisição de dados e controle (UAC) e por unidades dedicadas (especializadas) como os relés digitais, os equipamentos de oscilografia, os reguladores (de tensão, de velocidade) e os eventuais equipamentos para intertravamentos e para alarmes locais.

Nos sistemas digitais de automação elétrica, os dados normalmente coletados são:

a) tensões e correntes (e grandezas derivadas);

b) estado de equipamentos (disjuntores, seccionadoras, chaves de comando, atuação de relés etc), inclusive o estado dos módulos de "hardware" do sistema digital;

c) temperaturas;

d) no caso de usinas hidrelétricas: níveis de água, vazões, pressões;

e) no caso de usinas térmica: fluxos, níveis de combustíveis, etc.

2.6.3.**1** - Unidade de Aquisição de Dados e Controle (UAC)

As UAC podem abrigar grande número de pontos (entrada/saída) dependendo da aplicação. As vezes a UAC tem que atender especificações para ler algumas entradas digitais com resolução de lms e outras com resolução mais lenta lOms. Neste caso, pode resultar que o

fabricante venha optar por dividir a UAC em duas ou mais partes com processadores separados para UAC de grande porte. Alguns fabricantes apresentam como solução uma arquitetura distribuída consistindo de uma rede local com vários módulos separados, cada um com sua própria CPU, memória e comunicação (resultando num gabinete com um ou mais módulos para entrada analógica, outros para entrada digital, outros para saída digital e outros para saída analógica e alguns módulos mistos).

No mercado encontram-se dois tipos de equipamentos que podem ser utilizados para UAC: as denominadas UTR - Unidade Terminal Remota e os CLP - Controladores Lógicos Programados. Ambos têm arquitetura semelhante e podem ser utilizados para aquisição de dados, dependendo do requisito desejado.

1

CAPÍTULO 3

REDES NEURAIS ARTIFICIAIS

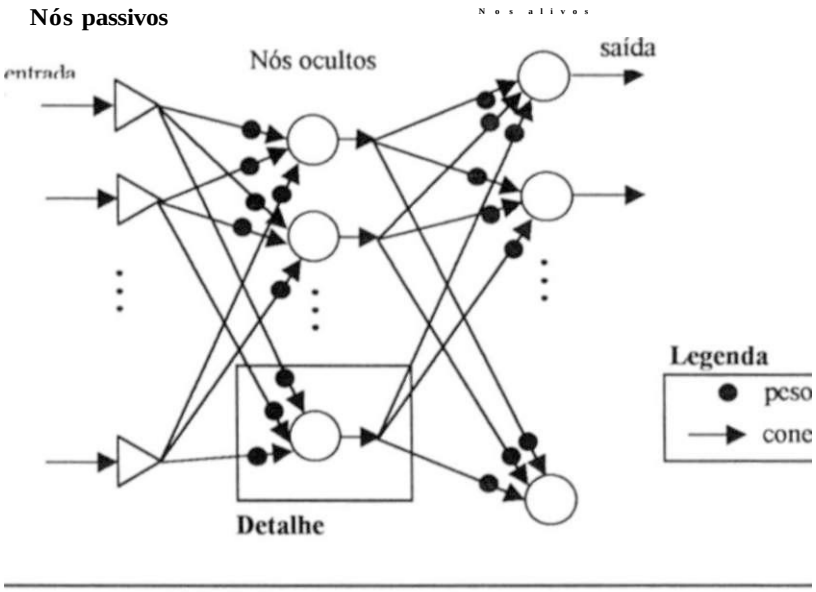
3.1 - INTRODUÇÃO

A Rede Neural é uma técnica de Inteligência Artificial cada vez mais presente em aplicações e publicações científicas. Ela procura imitar o modo de funcionamento do cérebro humano, tentando produzir artificialmente certos comportamentos característicos das redes neuronais biológicas e usá-los como ferramenta de computação.

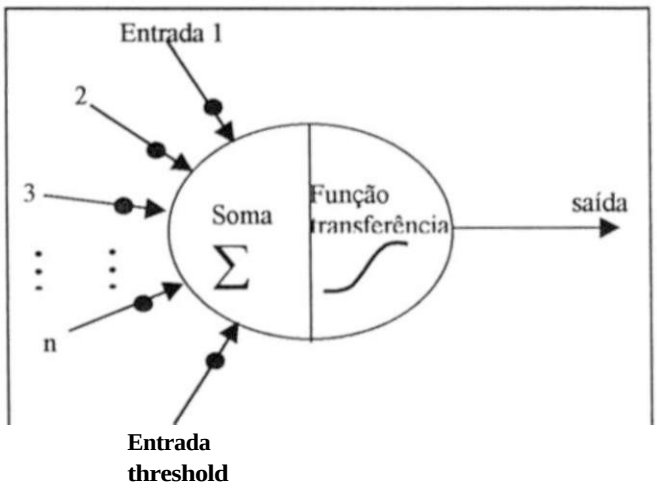
Um grande número de tarefas complexas que um ser humano pode realizar com aparente facilidade, não é realizado tão facilmente por computadores que usam métodos de algoritmo tradicional. Espera-se que estas tarefas sejam melhor executadas por computadores cujas estruturas e operações de processamentos sejam similares àquelas encontradas no cérebro humano.

As Redes Neurais Artificiais (RNAs) são sistemas paralelamente distribuídos.

compostos de simples neurônios como unidade de processamento, dispostos em uma ou mais camadas. Existe um grande número de conexões com pesos associados entre os neurônios. Estes pesos codificam o conhecimento de uma RNA e são usados para definir a influência de cada entrada recebida por um neurônio na sua saída. A saída de um neurônio é o resultado de uma função de ativação aplicada à soma ponderada de suas entradas (fig. 3.1).



(a)



(b)

Fig.3.1 (a) Arquitetura de uma Rede Neural de três camadas, (b) Detalhe.

Uma das principais características dos modelos das RNAs é sua habilidade de aprender através de exemplos, o que significa que elas não necessitam ser programadas para executar uma dada tarefa. Para isto, a rede necessita ser treinada com um conjunto de exemplos representativos da tarefa que ela deve realizar. Por exemplo, na tarefa de classificação, um bom conjunto de exemplos representativos seria aquele que contivesse exemplos de todas as classes do problema. Após treinada, a rede deve ser capaz de generalizar o que aprendeu, de forma que possa ser aplicada a situações similares. Esta habilidade permitirá o correto reconhecimento de padrões similares àqueles encontrados no conjunto de treinamento. Sistemas RNAs têm sido aplicados com muito sucesso em vários problemas práticos, como: reconhecimento de padrões, sistema de controle dinâmico e previsão.

Dependendo da aplicação em vista, também tem sido necessário a colaboração de outras áreas para a boa utilização das redes: Psicologia, Neurologia, Neurolingüística, Algoritmos Genéticos e Lógica Fuzzy.

3.2 - HISTÓRICO DAS REDES NEURAIS

A origem das RNAs se encontra em um estudo do psicólogo Donald Hebb, em seu livro Organization of Behavior (1949), onde descreveu pela primeira vez o funcionamento quantitativo da sinapse e do processo de treinamento humano. A idéia foi desenvolvida por Rosemblatt, do MIT (USA) e publicada em seu Perceptrons (o modelo matemático da sinapse humana) de 1957. Mas, em 1969, Minsky, em outra obra também com o mesmo título, mostrou que

o futuro desse modelo não era muito promissor, devido às grandes dificuldades da matemática envolvida, bem como aos poucos recursos computacionais disponíveis. Essa colocação encerrou a primeira onda das RNAs (1949-1969).

O assunto ficou praticamente esquecido (a segunda onda), até que, em 1986, (início da atual terceira onda) o professor de psicologia da Stanford University, David E. Rumelhart, e seu colega James L. McClelland, professor de psicologia da Carnegie-Mellon University, publicaram o famoso livro, Parallel Distributed Processing: Explorations in the Microstructure of Cognition (vol.l: Foundations, vol.2: Psychological and Biological Models). Neste livro, eles apresentaram o algoritmo backpropagation, um modelo matemático e computacional que, pela primeira vez, propiciou o treinamento supervisionado dos neurônios artificiais. Tornando a rede operacional e prática, deu-se um explosivo desenvolvimento dessa área.

Porém, logo começaram a surgir alguns problemas. O treinamento da rede era muito complexo e demorado. Novos algoritmos de otimização dos pesos das sinapses, como o do Gradiente Conjugado de Polak [19] e do "Simulated Annealing" [20] tornaram o treinamento supervisionado mais rápido.

O treinamento das RNAs requer uma enorme massa de dados históricos, já que a
1

aprendizagem se dá pela repetida (e até enfadonha) apresentação de exemplos passados (patterns) à rede. E o tempo de treinamento cresce muito com o número de exemplos e de camadas ocultas. Para simplificar, lança-se mão de uma nova técnica, a Lógica Fuzzy (lógica nebulosa) criada nos EUA, mas desenvolvida no Japão (os americanos não levaram a sério essa nova teoria, que desmontava, ao mesmo tempo, os fundamentos da Teoria da Probabilidade e da Lógica Aristotélica!). Através da Lógica Fuzzy é possível reduzir o volume dos dados, fuzificando-os (processo de gerar valores

membros para as variáveis fuzzy através das funções membros) e usando-os assim para treinar a rede. Após o aprendizado, os neurônios começam então a ter um comportamento condizente com o treinamento recebido, mas suas respostas são fuzzy. Realiza-se, então, uma defuzificação (processo de transformar uma saída do contexto fuzzy em uma saída do contexto aristotélico), obtendo resultados compreensíveis para os usuários da rede.

O uso das RNAs tem crescido de forma explosiva nos países desenvolvidos, abrindo às empresas uma tecnologia revolucionária na tomada de decisão. Não é difícil prever que, dentro de mais alguns anos, as empresas que ainda não estiverem fazendo uso dessa técnica, fatalmente acabarão por perder competitividade no mercado, com suas naturais conseqüências, numa situação similar ao que ocorre com o uso da tecnologia da informática no mundo atual.

3.3 - REGRA DE APRENDIZAGEM

A regra de aprendizagem para o Perceptron de Múltiplas Camadas (perceptron - modelo mais simples do neurônio biológico proposto em 1943 por McCulloch e Pitts) é chamada de Regra Delta Generalizada ou Regra da retropropagação do erro (erro backpropagation), e foi sugerida em 1986 por Rumelhart, McClelland e Williams.

A operação da rede é similar àquela do Perceptron de uma camada [7,18], onde mostram-se à rede os padrões e calcula-se suas respostas. A comparação com a resposta desejada permite que os pesos sejam alterados de forma que a rede possa produzir uma saída mais exata da

próxima vez em que o mesmo padrão lhe for apresentado. A regra de aprendizagem provê o método para ajuste dos pesos na rede com base na diferença entre os valores desejados e os produzidos pela rede.

3.3.1 - A MATEMÁTICA

A notação utilizada é a seguinte:

E_p - função erro para o padrão ***p***
tpj - saída **desejada** para o padrão ***p*** no nó ***j***
o_{pj} - saída **real** para o padrão ***p*** no nó ***j***
Wu - o peso da conexão do nó ***i*** para o nó ***j***

Definindo a função Erro como sendo proporcional ao quadrado da diferença entre a saída desejada e a real para todos os padrões a serem aprendidos

$$E_p = \frac{1}{2}\sum_{j}(t_{pj} - o_{pj})^2 \qquad (3.1)$$

A fração $\frac{1}{2}$ torna as expressões finais mais simples.

O estado de ativação de cada nó ***j*** para o padrão ***p,*** é dado por:

$$net_{pj} = \sum w_{ij} o_{pi} \qquad (3.2)$$

Ou seja, é obtido pela soma ponderada das entradas no n ó /

A saída de cada nó ***j*** é gerada pela função de ativação ***f*** atuando sobre o resultado da soma ponderada.

$$<>_{pj}=fj("e<_{pj}) \quad \textbf{(3.3)}$$

No perceptron múltiplas camadas, geralmente a função de ativação é a função sigmóide (abordada no item 3.4), apesar de que qualquer função monotônica, contínua c diferenciável possa ser utilizada.

A idéia do gradiente descendente no erro é calcular o erro de cada exemplo de treinamento (eq. 3**.1)** que é apresentado à rede e determinar o gradiente descendente do erro, o qual é função dos pesos, visto que a saída real é função do estado de ativação (eq. 3.3), e esta é função dos pesos (eq. 3.2). Haverá um gradiente ou declividade para cada conjunto de peso das conexões. O objetivo é encontrar os pesos que fornecem o menor erro. Procedendo-se no sentido de minimizar a função erro pode-se escrever:

$$\frac{\partial E_p}{\partial net_{pj}}$$

pela regra da cadeia.

Olhando para o segundo termo na equação (3.4), e substituindo o termo net_{pj} da equação (3.2),

$$\frac{\partial net_{pj}}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}}\left(\sum_k w_{kj} o_{pk}\right) = o_{pi} \quad (3.5)$$

Uma vez que é assumida a independência dos pesos, isto é,

$$dw_{ij} = \eta \frac{\partial E}{\partial w_{ij}}$$

$0^{\wedge} \sim^{u}$ exceto quando $k = i$, quando $^{\wedge} \sim\sim^{1}$

Definindo-se a mudança no erro em função do estado de ativação *net* da unidade como:

$$\hat{o}net_{pj} \quad » \tag{3.6}$$

de modo que a equação (3.4) torna-se

$$\tilde{O}E_{,,}$$

Decrescer o valor de *Ep* então significa fazer a mudança no peso proporcional *&õpj°pi,* isto é,

$$\&p^{w}ij^{=r}l\tilde{o}pj^{0}ni \tag{3.8}$$

Precisa-se saber o que significa *Opi* para cada uma das unidades. Sabendo-se isto, então pode-se decrescer *E.* Usando a equação (3.6) e a regra da cadeia, pode-se escrever

$$S_{"} \quad \frac{\hat{O}Ep}{\tilde{O}net_{pj}} = \frac{\tilde{O}E}{\hat{O}o_{pj}\hat{o}net_{pJ}} > ' \; d0pj \tag{39}$$

Considere o segundo termo, e da equação (3.3)

$$\frac{d0_j}{dn^{**}_{pJ}} = fj(\hat{}pj) \tag{3.10}$$

Considere agora o primeiro termo da equação (3.9). Da equação (3.1), pode-se derivar E_p em relação a o_{pj}, resultando

$$\tilde{O}E_{,,} = -(' p j - ° P j) \tag{3.11}$$

Então

$$8_{ti} = f'(net_{lj})(t_{l>l} - o_{pl}) \tag{3.12}$$

Isto é válido para as unidades de saída, onde tanto a saída desejada quanto a saída real pela rede estão disponíveis, mas não para as unidades ocultas, pois nestas a saída alvo não é conhecida.

Então, se a unidade j não é uma unidade de saída, pode-se escrever, pela regra da cadeia, que

$$7\ \hat{}\ = 2\ ,\ 7\ 7\frac{\hat{}}{} = 2\text{-}\acute{i}\ \tilde{a}\ ,\ \tilde{a}\frac{L}{}\ ,\ ''\ i\ k\ °\ p\ i \tag{3.13}$$
$$0°PJ \quad k \quad dnet_{pk} \quad \tilde{O}o_{pj} \quad {}_k \quad dnet_{p\ k} \quad do_{pj}\ ,$$

$$= 'Tj^{s}pk^{w}Jk \tag{3.14}$$

Usando-se as equações (3.2) e (3.6), e observando que o somatório $'',k^{\cdot)},i$ desaparece, pois a derivada parcial só não é zero para um único valor, exatamente como em

(eq. 3.5). Substituindo (eq. 3.14) em (eq. 3.9), encontra-se finalmente

$$\Delta_p w_{ji} = \eta \delta_{pj} i_{pi} \qquad (315)$$

Esta equação representa a mudança dos pesos das conexões das camadas ocultas. Ela provê um método para mudar a função erro de forma a minimizá-la. Sua expressão mostra que ela é proporcional aos erros δ_{pk} nas unidades subsequentes, logo os erros têm que ser calculados primeiro na camada de saída através da equação (3.12), para em seguida serem passados através da rede no sentido entrada para as unidades das camadas anteriores, de modo a permitir o ajuste dos pesos das conexões da rede . E a passagem destes erros no sentido contrário, que caracteriza a rede como rede backpropagation. As equações (3.12) e (3.15) definem como as redes múltiplas camadas podem ser treinadas.

3.3.2 - ALGORITMO BACKPROPAGATION

Passo 1 Apresenta-se um estímulo p à camada de entrada da rede através de um esquema pré-definido de apresentação, e também a correspondente resposta desejada t_p .

Passo 2 Computam-se a partir da primeira camada intermediária até a camada de saída, as respostas fornecidas (resposta real) pelas unidades de processamento da rede.

Passo 3 Calcula-se na camada de saída, o erro cometido correspondente à diferença entre a resposta desejada e a resposta real fornecida através de uma determinada **função erro** Ep . Se o

valor de Ep em um dado instante é menor que um certo valor minimo *z* estabelecido, o processo de aprendizagem é finalizado e o algoritmo desvia para o **passo** 6. Caso contrário, segue para o **passo 4.** Diferentes tipos de função erro podem ser empregados, sendo entretanto, a função mais comum aquela que utiliza o *erro quadrático* dado pela equação (3.1).

Passo 4 Atualização dos pesos das conexões. O processo ocorre da camada de saída em direção à camada de entrada, passando pelas camadas intermediárias. O ajuste dos pesos é dado pela Equação:

$$\text{A w } U \text{ - T } l \text{ s } p\ j \text{ o } p\ i \qquad (316)$$

onde *TJ* representa a taxa de aprendizagem, On/é a saída gerada pela unidade / que se conecta à unidade***j*** (entrada para a unidade /) e ***o,j*** corresponde ao erro ocorrido entre a resposta desejada e a resposta fornecida pela unidade***j*** para o padrão ***p.*** No caso dos neurônios da camada de saída, o valor de ***Ô pj*** é dado por:

$$Spj = f\ (net\ pj)(/_{,j} - o\ pj) \qquad (3.17)$$

Quando a função de ativação jÇ utilizada é a função ***sigmóide,*** a equação acima se torna, após o cálculo da derivada primeira /,•:

$$\text{ô pj} = \text{ Pj(PJ-°P.Í^'-°Pj)} \qquad (3\ 18)$$

Passo 5 Para as unidades ***j*** nas camadas intermediárias, não há como computar diretamente os ajustes em função do erro $(t_{pj} \sim O_{pj})$ e, portanto, os mesmos são calculados em função dos erros ***Oji*** observados nas unidades das camadas subsequentes, da seguinte forma:

$$^{5}Pj=fj(^{net}Pj)YPp\wedge Jk \qquad (3\ 15)$$

$$k$$

que no caso da função ***sigmóide,*** se torna:

$$àpi\wedge O\wedge -OpifL^{o}*!»» \qquad (3.20)$$

$$k$$

Após os ajustes, retorne ao **passo 1**

Passo 6 Fim do Algoritmo

3.4 -FUNÇÃO SIGMOIDE

Uma das vantagens de se usar a função sigmóide como função de ativação é que ela é bastante similar a função degrau, e então, demonstra um comportamento de natureza similar.

A função sigmóide é definida como:

$$f(net) = \frac{1}{(1 + e^{-k\ net}} \tag{3.21}$$

e assume valores entre Oel.A: é uma constante positiva que controla a inclinação da função - valores elevados de *k* achatam a função e quando *k*—>oo,/f/7t'^->função degrau, como pode-se observar na fig.3.2. O parâmetro *k* também atua como um controle automático de ganho, desde que para sinais de entrada pequeno a inclinação é quase degrau e assim a função muda rapidamente produzindo um ganho elevado. Para entradas de valor elevado a inclinação é pequena e então o ganho é muito pequeno. Isto significa que a rede pode aceitar entradas de valor elevado e ainda permanecer sensível a pequenas mudanças.

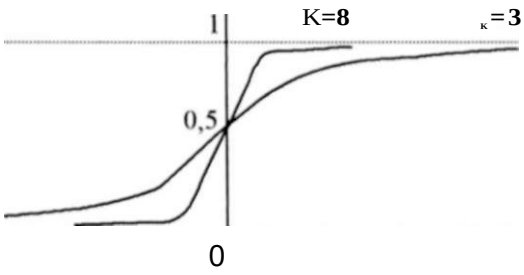


Fig.3.2 Curvas da função sigmóide para diferentes valores dc K

Uma das grandes razões para seu uso é que a função sigmóide apresenta uma derivada simples, e isto torna a implementação do sistema de retropropagação muito mais fácil. Dado que a saída da unidade o_{pj} é dada por:

$$o_{pj} = f(net) = \frac{/}{(1 + e^{-k\ net}} \tag{3.22}$$

a derivada em relação àquela unidade, / *(neí)*, é dada por:

$$f'(net) = \frac{ke^{-knet}}{(1 + e^{-knet})^2} = kf(net)(1 - f(net)) = kO_j(1 - O_j) \quad \textbf{(323)}$$

A derivada é então uma função simples da saída.

3.5 - DIFICULDADES NO APRENDIZADO

Alguns problemas apresentam dificuldades associadas à aprendizagem com o perceptron de múltiplas camadas. Algumas vezes a rede converge para uma solução estável que não fornece a saída correta. Nestes casos, a função energia está em um *mínimo local.* Isto significa que em qualquer direção que a rede se mova na superfície de erro, a energia será maior que a da posição atual. Pode ser que haja a necessidade de transpor uma pequena região em elevação antes de alcançar um mínimo mais profundo, mas o algoritmo não tem meios de saber disso, uma vez que o aprendizado é realizado seguindo a função energia descendo na direção oposta do gradiente que coincide com a direção de decréscimo da energia, até que alcance a base de um poço e neste ponto permanece, uma vez que não há mais nenhuma direção que reduza a energia.

Há meios alternativos que podem minimizar essas armadilhas [18], que são destacados a seguir.

a) Diminuindo o ganho ***tj***

Se a taxa que altera os peso decresce progressivamente, o algoritmo do gradiente descendente é capaz de alcançar uma solução melhor. Se o termo do ganho *"H* é grande no início, passos largos são tomados pelo peso e pelo espaço de energia na direção da solução. A medida que o ganho decresce, os pesos da rede estabelecem-se em uma configuração de mínima energia sem sobrepassar a posição estável quando o gradiente descendente toma curtos passos em declínio. Esta abordagem capacita a princípio a rede a ultrapassar o mínimo local e a instalar-se em algum mínimo mais profundo sem oscilar muito. Entretanto, a redução do ganho significa que a rede levará mais tempo para convergir.

b) Adição de ***nós*** internos

Um mínimo local pode ocorrer quando duas ou mais classes independentes são classificadas como pertencentes a mesma classe. Isto leva a uma representação interna pobre nas unidades ocultas, e então, a adição de mais unidades a estas camadas permitirá um melhor registro das entradas e menor ocorrência de mínimos.

c) Momento *d*

O momento pode modificar os pesos através da introdução de um termo extra na equação de adaptação dos pesos e produzirá uma mudança acentuada no peso se as mudanças atuais forem grandes, e decairá a medida que as mudanças se tornarem menores. Isto significa que a rede tem pouca probabilidade de ficar presa em um mínimo local no início do treinamento, visto que o termo do momento irá produzir variações superiores aos valores locais

da função energia, seguindo a tendência decrescente em toda a superfície. O momento é de grande eficácia na aceleração da convergência sobre gradientes de pouca declividade, permitindo que a rede desenvolva um caminho mais rápido na direção do declive. A superfície de energia pode conter ravinas longas e gradualmente inclinadas que terminam em um mínimo. A convergência ao longo destas ravinas é lenta, desde que a direção a ser seguida tem um gradiente pequeno, e geralmente o algoritmo oscila sobre o vale a medida que caminha para a solução, como mostra a fig.3.3 abaixo. E difícil acelerar sem aumentar a chance de sobrepassar o mínimo, mas a eficácia de atuação do termo do momento é tão elevada que a sua adição é um procedimento amplamente utilizado.

Este termo do momento pode ser escrito como se segue na equação de adaptação do peso:

$$^{w}(t + !) = +1 \bullet d\,W_{(i+1)mal} + a \bullet dW_{(i)}$$

onde ***CC*** é o fator momento, ***0 < CC < 1.***

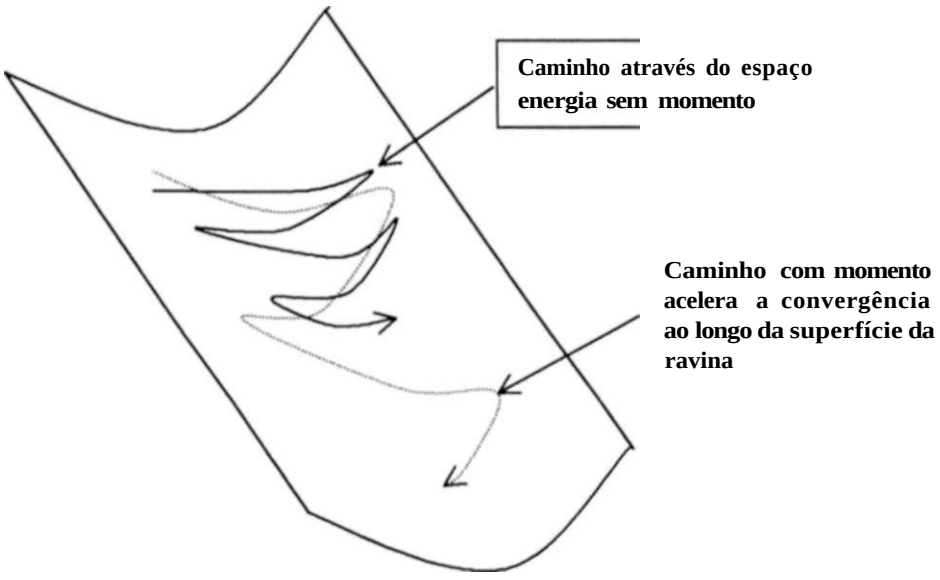


Fig.3.3 Caminho traçado pela rede durante o aprendizado com e sem o momento a

d) Adição de ruído

Se um ruído aleatório for adicionado ao peso, isto afastará o algoritmo do gradiente descendente da linha de maior declividade, podendo dessa forma deslocar o sistema para fora de um mínimo local, caso este ruído seja suficiente. Esta técnica tem a vantagem de consumir pouco tempo de computação, e logo, a adição desse termo não torna o algoritmo apreciavelmente mais lento que o algoritmo do gradiente descendente direto.

3.6 - FORMAS DE TREINAMENTO

Em função do número de exemplos do banco de dados que é utilizado para ajustar os pesos das conexões na rede, pode-se definir a forma de treinamento [8] como:

a) Cada exemplo (incremental)

1. Para cada exemplo do banco de dados

1.1 processa-se a entrada obtendo-se uma saída

1.2 calcula-se o erro quadrático

1.3 calcula e ajusta as variações necessárias nos pesos, isto é:

$$dW_{(t+1)total} = \sum dW_{(t)cada\ exemplo}$$

1.4 Atualiza os pesos a partir das variações calculadas

$$w(t+1) = w(<) + V \bullet dw(l+l)lotal + \alpha \bullet dw(t)$$

onde: $T]$ *é a* taxa de aprendizado ; a é taxa de momento. Ambas as taxas servem para acelerar o aprendizado da rede.

Esse processo, e a conseqüente exteriorização do erro da rede utilizando-se esse tipo de treinamento é mais rápida, já que cada passo significa treinar um único exemplo do banco de dados. Esse tipo de treinamento proporciona um gráfico de erro mais oscilatório, já que a rede está sujeita à variações de pesos mais bruscas do que nas outras formas de treinamento.

b) Todos exemplos

1. Para cada exemplo do banco de dados

1.1 processa-se a entrada e calcula-se o erro quadrático

1.2 calcula e acumula as variações necessárias nos peso, isto é:

$$dW_{(t+1)total} = \sum dW_{(t)cada\ exemplo}$$

2. Atualiza-se os pesos a partir das variações calculadas

$$W_{(1}M, \quad = \quad W_{(1)}+/7\text{-}dW_{\mathbf{(t+l)total}} \qquad d»$$

Ou seja, a atualização dos pesos só é feita após acumular as variações de todos os exemplos no banco de dados.

Esse processo, e conseqüente exteriorização do erro da rede utilizando-se esse tipo de treinamento é mais lenta, porem menos sujeita a oscilações e mais robusta, já que cada passo significa treinar todos os exemplos do banco de dados.

c) Cada N exemplos (subconjunto do banco de dados)

1. Para cada exemplo do subconjunto do banco de dados

   1.1 processa-se o erro e calcula-se o erro quadrático

   1.2 calcula e acumula as variações necessárias nos pesos, isto é:

$$dW_{(t+1)total} = \sum dW_{(t)cada\ exemplo}$$

2. Atualiza-se os pesos a partir das variações calculadas

$$w(t+!) = w(t) + n \bullet dW_{(t+1)total} + a \bullet dW_{(t)}$$

ou seja, a atualização dos pesos é feita após acumular as variações dos N exemplos escolhidos.

Esse processo, e a conseqüente exteriorização do erro da rede utilizando esse tipo de treinamento dependerá da quantidade de exemplos que o usuário escolher. Quanto mais exemplos mais lento o processo. Esse tipo de treinamento proporciona um gráfico de erro bem menos oscilatório que o treinamento standard, já que a rede está sujeita à variações de pesos bem menos bruscas.

3.7 - PROCESSO DE CLASSIFICAÇÃO

Dentre as possíveis aplicações de redes neurais artificiais destaca-se o seu emprego em reconhecimento de padrões, que em geral, é a ciência que compreende a identificação e a classificação de informação em categorias.

O objetivo fundamental para o reconhecimento de padrão é a classificação: dado uma entrada, pode-se analisar esta entrada para prover uma classificação significativa dos seus conteúdos?

Pode-se considerar um sistema de reconhecimento de padrão como um dispositivo de dois estágios. O primeiro estágio é a extração das características e o segundo é a classificação.

Define-se a característica como a medida tirada no padrão de entrada a ser classificado. Está-se tipicamente procurando por características que ofereçam mais poder de discriminação.

O classificador é suprido com a lista das características medidas. Sua tarefa é mapear estas características das entradas em um estado de classificação, isto é, dado as características da entrada, o classificador deve decidir a que tipo de classe elas se aproximam mais.

3.7.1 - MLP - CLASSIFICADOR

Um classificador neural dos mais comuns é o Perceptron Múltiplas Camadas (MLP). Este tipo de rede neural pode empregar um grande número de regras de aprendizagem, entre as quais a mais importante é a regra backpropagation, baseada no método do gradiente descendente de otimização. Como pode-se observar na figura 3.4, a arquitetura de uma rede neural MLP consiste de camadas de blocos computacionais chamados neurônios. A camada de entrada tem um número de neurônios sensores igual a dimensão do dado de entrada. A única função desses neurônios sensores é ler os dados de entrada e passá-los para a camada seguinte via uma conexão ponderada. Os neurônios na camada seguinte somam todos os sinais de entrada e então aplicam uma função não linear ao resultado. Alguma das funções comuns mais usadas com redes MLP são: a degrau, a sigmóidal e a hiperbólica. As saídas das funções não lineares em uma camada são transferidas para a próxima camada de neurônios via conexões ponderadas. Este procedimento é repetido até que o sinal alcance a camada de saída. Os sinais que saem da camada de saída são comparados com as classes alvo para um padrão particular de entrada, e a diferença entre a saída e o sinal desejado são retornados à rede no sentido contrário, ou seja, no sentido da entrada. Usando esses sinais de erros e os gradientes da função não linear, os pesos das conexões entre as camadas são ajustados. Este

procedimento é repetido para todos os dados de treinamento por várias iterações, até que os ajustes dos pesos na rede produza uma saída adequada para o problema em questão, ou seja, tão próxima o quanto possível da saída desejada. Neste instante o treinamento é finalizado e a rede é considerada treinada.

Deve-se notar que o número de unidades das camadas de entrada e saída dependem da aplicação específica do sistema a ser analisado e que, a determinação do número de unidades das camadas ocultas é empírico, obtido através da análise do desempenho da rede neural.

O TESTE da rede é realizado através da apresentação de um conjunto de teste para a rede, e nesse estágio é permitido a passagem dos dados apenas no sentido entrada - saída. Nenhuma passagem no sentido saída - entrada ou ajuste de pesos é realizado durante a fase de teste da rede.

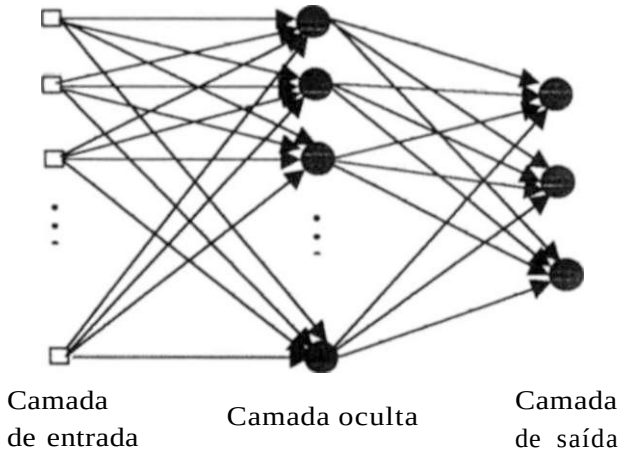


Fig. 3.4 Perceptron de múltiplas camadas com uma camada oculta.

3.8 - VANTAGENS E DESVANTAGENS DAS REDES NEURAIS

Observou-se em **[9]** as seguintes vantagens e desvantagens das Redes Neurais.

Vantagens:

- A capacidade de generalização, isto é, saber responder de forma correta à casos novos, desde que similares aos casos aprendidos. Esta capacidade surge do fato de o conhecimento das redes estar distribuído pelos diversos neurônios na forma de pesos.
- A capacidade de mapear um conjunto de entrada a um conjunto de saída (que estão relacionados por uma função complexa). Esta propriedade torna esta técnica adequada para se fazer a associação entre o conjunto de alarmes acionados e as falhas ocorridas.
- A característica de processamento paralelo intrínseco das Redes Neurais permite alta velocidade de obtenção do diagnóstico. Esta característica é muito importante em um sistema de potência onde a necessidade de um diagnóstico em tempo real é crucial.
- Capacidade de responder adequadamente a uma entrada incompleta. Esta característica é adequada no processamento de alarmes, visto que muitas vezes é possível durante um distúrbio no Sistema de Potência que um alarme não atue devido ao mau funcionamento de algum equipamento tornando o conjunto de alarmes que caracteriza a falta incompleto. A RNA é capaz de diagnosticar falhas mesmo quando o conjunto de alarmes que a caracteriza estiver incompleto.

Dentre as desvantagens pode-se citar:

- Não é capaz de explicar porque obteve uma determinada resposta, pois a representação do conhecimento está distribuída na forma de pesos das conexões entre os neurônios.
- Treinamento adicional da rede para acrescentar um novo dado pode acarretar na perda de alguma informação anteriormente aprendida Desta forma, é seguro reiniciar o treinamento incluindo este novo dado entre os dados previamente usados no conjunto de treinamento.

3.9 - VANTAGENS DA REDE NEURAL EM RELAÇÃO A UM SISTEMA ESPECIALISTA NO DIAGNÓSTICO DE FALHAS

Diagnóstico de falhas é uma técnica de grande importância para a operação estável e confiável de um sistema de potência. Nas técnicas convencionais os Sistemas Especialistas em diagnóstico de falhas são construídos com base em regras que incorporam a experiência e o conhecimento de seres humanos especialistas. Estes sistemas diagnosticam as falhas, combinando seqüencialmente os estados dos disjuntores e relés com as hipóteses da base em regras. Entretanto, os Sistemas Especialistas são muito lentos para serem aplicados em tempo real e requerem um processo exaustivo e demorado na construção de suas base em regras. Como o Sistema Especialista com base em regras procura seqüencialmente pela solução através da combinação do padrão com as hipóteses, a velocidade do diagnóstico depende do tamanho das bases em regras e da estratégia da pesquisa. Particularmente, o diagnóstico de falhas múltiplas é um processo complexo e difícil.

As Redes Neurais são modelos matemáticos da atividade do cérebro; são também conhecidas como arquitetura "conexionista" c processo paralelamente distribuído. A alta velocidade de execução, representação distribuída do conhecimento e tolerância ao erro são vantagens consideráveis das Redes Neurais no diagnóstico de falhas em Sistemas de Potência. Uma outra vantagem da abordagem de Rede Neural sobre os Sistemas Especialistas é que aquela tem a capacidade de interpolação para produzir resultados apropriados para entradas com ruído e entradas não utilizadas no treinamento.

CAPÍTULO 4

LÓGICA FUZZY

4.1 - INTRODUÇÃO

O desenvolvimento de técnicas de inteligência artificial (IA) nos últimos anos, ocupa cada vez mais uma posição de destaque em pesquisas na área de controle de processos industriais e, aos poucos, começam a ser implantadas em plantas industriais com enorme sucesso. Dentre as técnicas mais comumente utilizadas, além das Redes Neurais aplicadas a sistemas de controle, pode-se destacar, os Sistemas Fuzzy de controle que estão atualmente em tamanha evidência que os japoneses as consideram como duas das mais promissoras técnicas para o século XXI.

A lógica fuzzy tem sido aplicada a problemas de sistemas de potência que contém objetos conflitantes, quais sejam: os custos do suprimento máximo de carga e o mínimo de geração. As principais áreas de sistema do potência onde já foi aplicada a lógica fuzzy são: planejamento, operação, controle e diagnóstico.

4.2 - HISTÓRICO DA LÓGICA FUZZY

O conceito de conjunto Fuzzy foi introduzido, em 1965, por Lotft A. Zadeh (Universidade da Califórnia, Berkeley). Zadeh observou que os recursos tecnológicos disponíveis, naquela época, eram incapazes de automatizar as atividades relacionadas a problemas de natureza industrial, biológica ou química, que contivessem situações ambiguas, não passíveis de processamento através da lógica computacional fundamentada na lógica booleana.

Procurando solucionar esses problemas o Prof. Zadeh publicou em 1965 um artigo resumindo os conceitos dos conjuntos Fuzzy, revolucionando o assunto com a criação de sistemas Fuzzy.

Em 1974, o Prof. Mamdani, do Queen Mary College, Universidade de Londres, após inúmeras tentativas frustradas em controlar uma máquina a vapor com tipos distintos de controladores, incluindo o P1D, somente conseguiu fazê-lo através da aplicação do raciocínio Fuzzy.

Esse sucesso serviu de alavanca para muitas outras aplicações, como em 1980, no controle Fuzzy da operação de um forno de cimento. Vieram em seguida várias outras aplicações, destacando-se, por exemplo, os controladores Fuzzy de plantas nucleares, refinarias, processos biológicos e químicos, trocador de calor, máquina diesel, tratamento de água e sistema de operação automática de trens.

Estimulados pelo desenvolvimento e pelas enormes possibilidades práticas de aplicações que se apresentaram, os estudos sobre sistemas Fuzzy e controle de processos avançam

rapidamente, culminando com a criação em 1984, da Sociedade Internacional de Sistemas Fuzzy, constituída, principalmente, por pesquisadores dos países mais avançados tecnologicamente.

4.3 - REPRESENTAÇÃO DO SISTEMA FUZZY ADITIVO

Um Sistema Fuzzy aditivo **F** armazena **m** regras fuzzy na forma SE **X = Aj** ENTÃO **Y = Bj** e calcula a saída F(x) como o centróide da parte ENTÃO dos conjuntos fuzzy **B j** parcialmente disparados e somados. Cada entrada x dispara todas as regras paralelamente e em um determinado grau. Logo o Sistema Fuzzy Aditivo F: R^n -> R^p atua como um processador associativo ou como ***uma memória associativa fuzzy*** (FAM - Fuzzy Associativc Memory) global [10]. Cada regra atua como uma FAM local. A ftg. 4.1 abaixo mostra o fluxo de um sistema aditivo para frente.

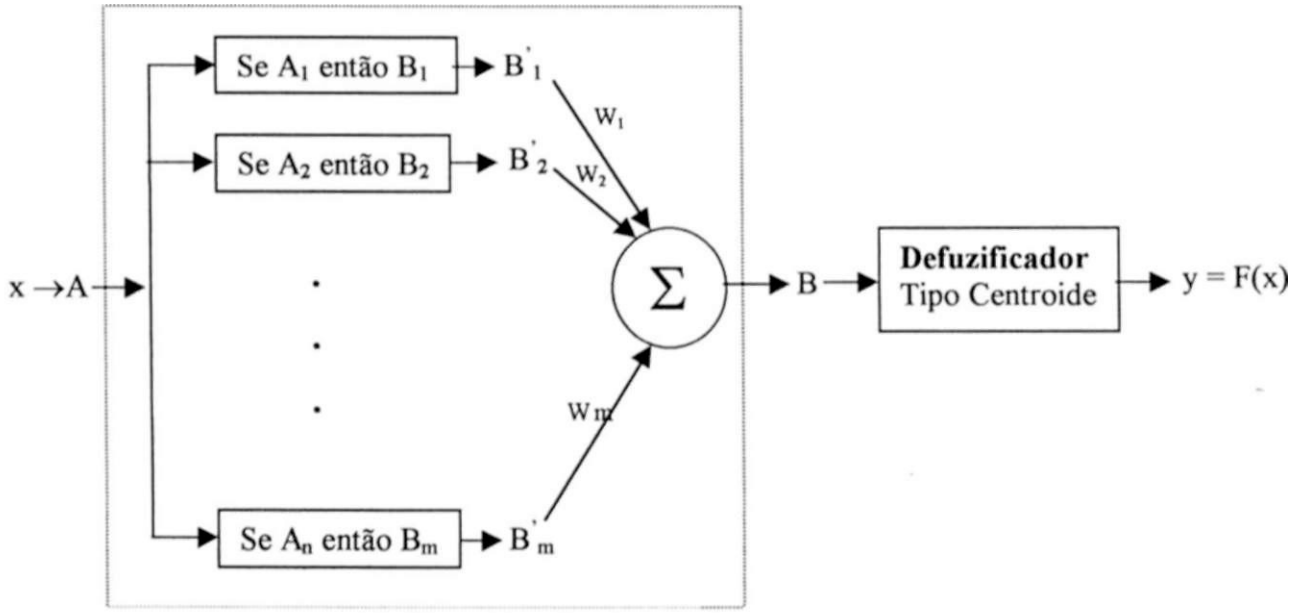


Fig. 4.1 Arquitetura do sistema fuzzy aditivo

A entrada real x dispara paralelamente as partes **Se** de todas as **m** regras em determinado grau **aj (x).** Com isso o sistema escala ou pondera as partes **Então** para produzir os novos conjuntos fuzzy **B j** e os soma para formar o conjunto de saída **B.** O sistema "defuzifica" **B** ou o mapea em um escalar para produzir a saída y = F(x) através do centróide de **B.** No caso discreto, **x** é um vetor de um elemento ou uma coluna de uma matriz identidade **n x n** e cada regra é uma matriz **n x p** ou uma memória associativa fuzzy (FAM). No caso contínuo, **xj** é um pulso delta ô(x - **X j) .** O Sistema Neural ou Genético ou outros adaptativos podem mudar os pesos das regras **wj** ou mudar os conjuntos fuzzy **Aj** ou **Bj** conforme os dados amostrais. O sistema fuzzy F global e suas regras determinam os modelos de FAMs.

O modelo aditivo padrão (SAM - Standard Additive Model) é o caso especial mais importante de um sistema fuzzy aditivo sendo por isto vastamente utilizado nos Simuladores de Lógica Fuzzy. No caso SAM a parte **Então** do conjunto B j disparada é o produto ajustado aj (x) Bj. A estrutura de adição surge da soma B das partes **Então** dos conjuntos disparadas:

$$\overset{m}{\underset{H}{\wedge}} \quad (4.1)$$

$$\overset{m}{} \quad (4.2)$$

A soma se dá sobre todas as **m** regras que mapeam os subconjuntos fuzzy Aj c $\mathbf{R}^n$ nos subconjuntos Bj c $\mathbf{R}^p$**.** O valor ajustado (unidade fuzzy) aj(x) expressa o grau de pertinência da entrada **x** à parte **Se** do conjunto fuzzy Aj. Assim aj : $\mathbf{R}^n$ —> [0, 1] e bj : $\mathbf{R}^p$ —> [0, 1] definem as funções especificadas. Os pesos das regras **wj** podem ser qualquer número real no intervalo [0, 1].

A soma em (4.1) aplica-se a um sistema aditivo geral [10], enquanto aquela em (4.2) representa a forma específica para um sistema SAM. O peso das regras pondera cada termo na soma para refletir a credibilidade ou freqüência das regras e dessa forma fornece um termo extra para um sistema em aprendizagem equilibrar-se. Eles se juntam a estrutura SAM como múltiplos dos valores **aj**(x) dos conjuntos de entrada. Logo para se ignorar os pesos das regras toma-se o valor 1 (um) para todas as unidades: **wi =...= w_m** = 1.

A soma em (4.2) fornece uma SAM se for adicionalmente mapeado o conjunto de B em um número F(x) através do cálculo do seu centróide para cada x.

O Teorema SAM mostra que um sistema fuzzy mapeia entradas x em F(x) como uma soma convexa dos centróides **Cj** das **m** partes **Então** dos conjuntos. O teorema SAM decompõe o centróide global de B nesta soma convexa dos centróides locais. Os **m** coeficientes pi(x),....,p_m(x) são convexos pelo fato de que cada termo é não negativo pj(x) > 0 e a soma de todos esses termos é 1 (um):

Assim, os pesos definem uma função densidade de probabilidade para cada **x.**

Teorema 4.1. (Teorema SAM) Suponha que o sistema fuzzy ***F: R?*** —> ***RF*** seja um ***modelo aditivo padrão.*** ***F(x)*** = Centroide(#) = ***Centroide*** $\int_{f}^{m}$ V r a , ^ ,

Assim ***F(x)*** é uma soma convexa dos centroides das **m** partes **Então** dos conjuntos:

$$F(x) = \qquad (4.3)$$

(4.4)

Os coeficientes convexos de pesos ***pj(x)....,p_m(x)*** dependem da entrada ***x*** através das proporções

$$Pj = \qquad (4.5)$$

Vj é um volume finito positivo (ou uma área ***sep*** •»/ no espaço imagem ***R^p)*** e ***c,è o*** centroide ***fí_j*** das partes **Então** dos conjuntos:

$$\hat{}=L\hat{}_{\ldots\ldots y_p)dyv4>_p>o} \qquad (4.6)$$

$$c. = \qquad (4.7)$$

Para o caso em que ***p 1*** as equações (4.6) e (4.7) ficam reduzidas a

(4.8)

$$c, = \int_{j''}^{foo} j(y)^{d}y \qquad (4.9)$$

A demonstração do Teorema SAM encontra-se no apêndice C.

4.4 - SISTEMA DE INFERÊNCIA TIPO MAMDANI

O método de inferência aplicado neste trabalho é o método Mamdani. Ele é um dos métodos mais utilizados dentro da metodologia fuzzy. O método Mamdani foi um dos primeiros sistemas de controle construído usando-se a teoria dos conjuntos fuzzy. Foi proposto em 1975 por Ebrahim Mamdani [11] como uma tentativa de controlar a combinação de uma máquina a vapor e uma caldeira através da sintetização dos conjuntos de regras lingüísticas de controles, obtidos de operadores humanos experientes. O trabalho de Mamdani foi baseado no artigo de Lotfi Zadeh publicado em 1973 sobre algoritmo fuzzy para processos de decisão e sistemas complexos [12].

O sistema de inferência Mamdani envolve o combinador máximo quando da aplicação da agregação, ou seja, combinação em um só conjunto fuzzy B de todos os conjuntos fuzzy de saída de cada regra B j, e aplica o método do centróide para defuzificação.

O combinador máximo em (4.10) utiliza ***b(y) = rmx^i,(x)b,(y):'} = \,...,m\.*** Um dos grandes problemas com o combinador máximo é que ele é de difícil computação ao contrário do combinador soma. A natureza do combinador máximo em (4.10) não produz uma solução simples se as partes ENTÃO dos conjuntos Bjse sobrepõem. Um caso extremo em que as partes ENTÃO dos conjuntos fuzzy B j são disjuntas reduz a equação (4.10):

visto que a igualdade x + ***y = m\n(x,y)*** + max(x,>') se reduz a ***x + y = max(x,y)*** se a interseção entre os pares for nula: min(x,***y)*** = 0 ou ***B***\ fl ***B'M*** = <***f***>.

4.5 - ETAPAS DO PROCESSO DE INFERÊNCIA

A idéia geral do Sistema de Inferência Fuzzy é interpretar os valores do vetor de entrada e, com base nos conjuntos de regras, atribuir valores para o vetor de saída, como ilustra a figura 4.2.

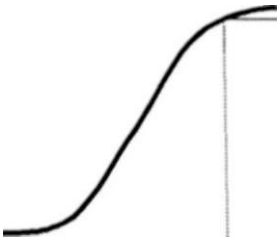
Fig.4.2 Diagrama esquemático do Sistema de Inferência Fuzzy

O processo de inferência fuzzy desenvolve-se nas seguintes etapas : **fuzificação** das variáveis de entrada, **aplicação** do operador fuzzy (E ou OU) no antecedente, **implicação** do antecedente para o conseqüente, **agregação** dos conseqüentes através das regras e **defuzificação.**

Etapa 1. Fuzificação das Entradas

Na primeira etapa, toma-se a variável de entrada e determina-se o seu grau de pertinência a cada um dos conjuntos fuzzy através da função membro (função triangular, trapézio, gaussiana, sigmóide, etc.). A entrada será sempre um valor numérico discreto limitado ao universo da variável de entrada em discurso (no caso abaixo, entre 0 e 1) e a saída fuzzy indica o grau de pertinência do membro (sempre entre 0 e 1). Logo a fuzificação não significa nada mais do que procurar na tabela ou avaliar a função.

O exemplo a seguir (Fig.4.3) utiliza como **variáveis de entrada** a saída de uma rede neural que produz um vetor de duas dimensões cujos valores variam entre 0 (zero) e l(um). Os termos lingüísticos fuzzy das variáveis utilizados para a montagem do sistema de inferência fuzzy são **nulo** e **certeza.** O sistema é constituído por duas regras, e cada uma destas regras depende da transformação das entradas em um número de diferentes conjuntos lingüísticos fuzzy: RN1 é nulo, RN1 é certeza, RN2 é nulo, RN2 é certeza. Antes das regras serem avaliadas, as entradas devem ser fuzificadas nestes conjuntos lingüísticos. Por exemplo, em qual extensão está RN1 com certeza ativado? A figura a seguir mostra o quanto a variável RN1 (0-1) se ajusta a variável lingüística "certeza". Neste caso exemplo, tomou-se para RN1 o valor 0,8, o qual, dado o gráfico de "definição de certeza", corresponde a |i=0,9.



- > 0,9

RESULTADO DA FUZIFICAÇÃO

RN1 é CERTEZA

í

RNI -0.8

Entrada

Fig.4.3 Primeira etapa do processo

Etapa **2.** Aplicação do operador Fuzzy

Uma vez que as entradas foram fuzificadas, sabe-se o grau com que cada parte do antecedente (parte SE das regras) satisfaz cada regra. Se o antecedente de uma regra tem mais que uma parte, então o operador fuzzy é aplicado para que se obtenha um número que represente o resultado do antecedente para aquela regra. Este número será então aplicado para a função de saída. A entrada para o operador fuzzy é dois ou mais valores membros das variáveis de entrada fuzificadas. A saída é um único valor verdade. Para maiores esclarecimentos ver operadores fuzzy no apêndice B.2.

Na figura 4.4, é mostrado um exemplo do operador E aplicado na avaliação da pré-classificação do evento pela rede neural. E avaliado o antecedente da regra 1 para o cálculo da saída da lógica fuzzy na classificação final do evento. As duas diferentes partes do antecedente (RN1 é certeza e RN2 é nulo) resulta num valor fuzzy 0,9 e 0,7 respectivamente. O operador fuzzy E simplesmente seleciona o mínimo dos dois valores, e a operação fuzzy para a regra 1 é completada.

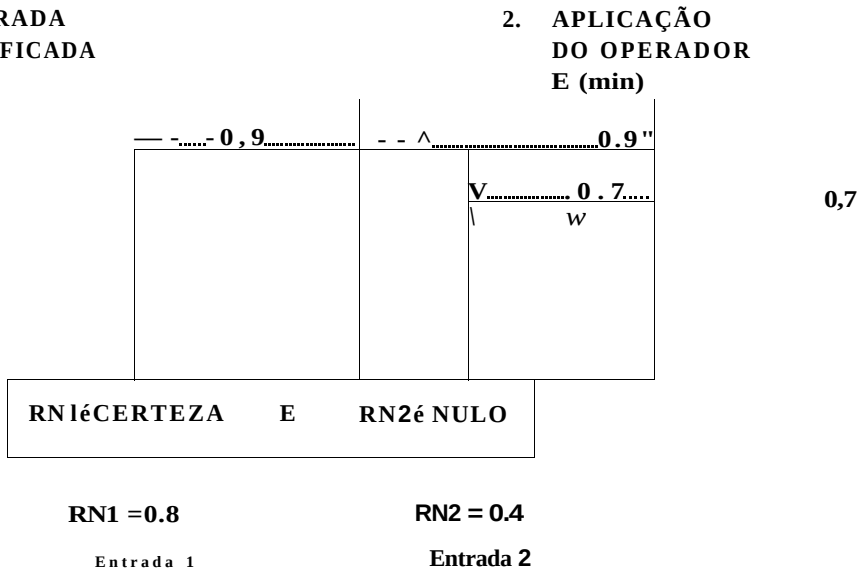


Fig.4.4 Primeira e segunda etapas do processo

Etapa 3. Aplicação do método de implicação

Antes da aplicação do método de implicação, deve-se tomar cuidado com os pesos das regras. Todas as regras têm um peso (um número entre 0 e 1) que se aplica ao número produzido pelo antecedente. Se o peso for 1 (como é para este exemplo) ele não tem nenhum efeito no processo de implicação.

O método de implicação é definido como a modelagem do conseqüente (parte que vem após o ENTÃO das regras) com base no antecedente (um número entre 0 e 1). A entrada para o processo de implicação é um número dado pelo antecedente, e a saída é um conjunto fuzzy. A implicação se dá em todas as regras. Este método utiliza a mesma função E min (mínimo) que trunca a saída do conjunto fuzzy (Ver fig.4.5).

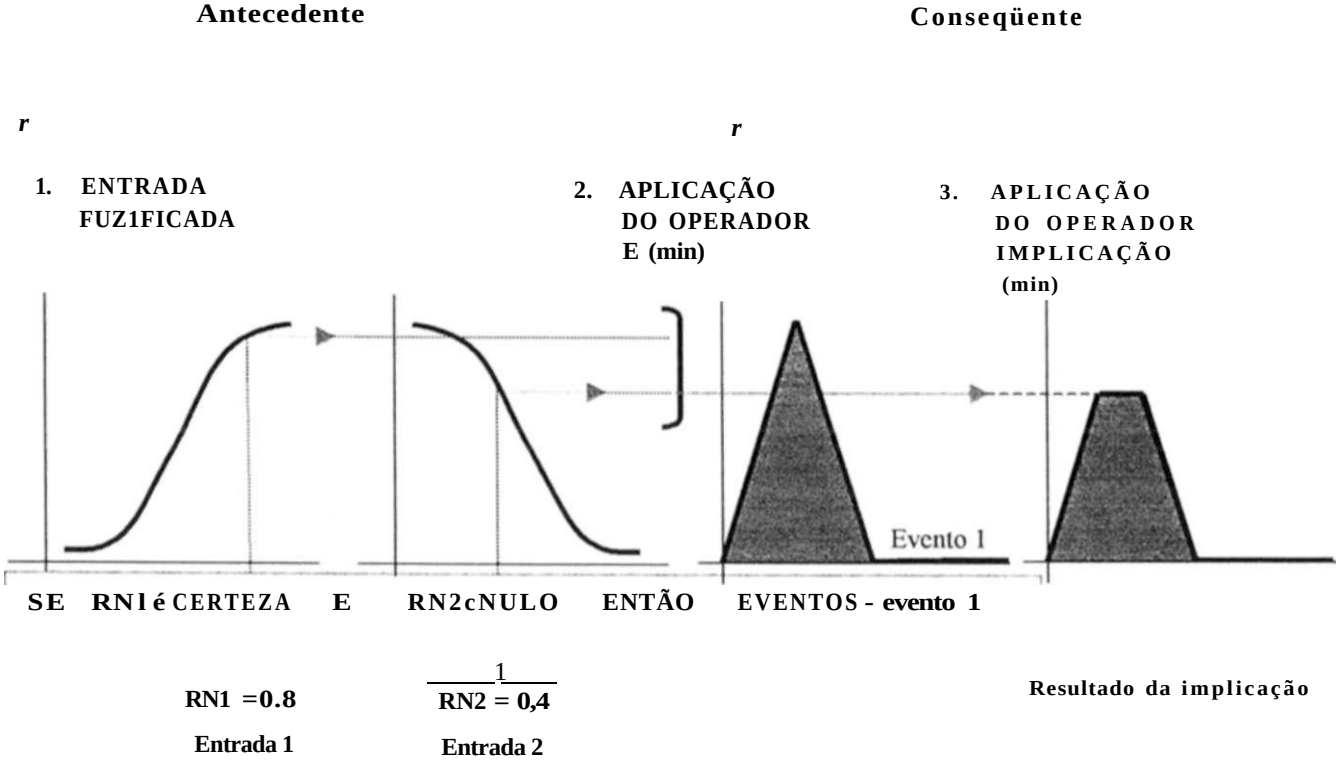


Fig.4.5 Primeira, Segunda e terceira etapas do processo

Etapa 4. Agrega todas as saídas

A agregação ocorre quando unifica-se todas as saídas de cada regra unindo as linhas paralelas. E só tomar todos os conjuntos fuzzy que representam a saída de cada regra e combiná-los em um único conjunto fuzzy que é o preparativo para a etapa final, a defuzificação. A agregação ocorre para cada variável de saída. A entrada para o processo de agregação é a lista das funções de saída truncadas, obtidas no processo de implicação para cada regra. A saída do processo de agregação é um conjunto fuzzy para cada variável de saída.

O processo de agregação é comutativo, logo, a ordem em que as regras são executadas não é importante. O operador usado no processo de agregação é o max (máximo).

No diagrama a seguir (fig.4.6), todas as duas regras são colocadas uma abaixo da outra para mostrar como a saída de cada regra é combinada, ou agregada, em um único conjunto fuzzy.

1.

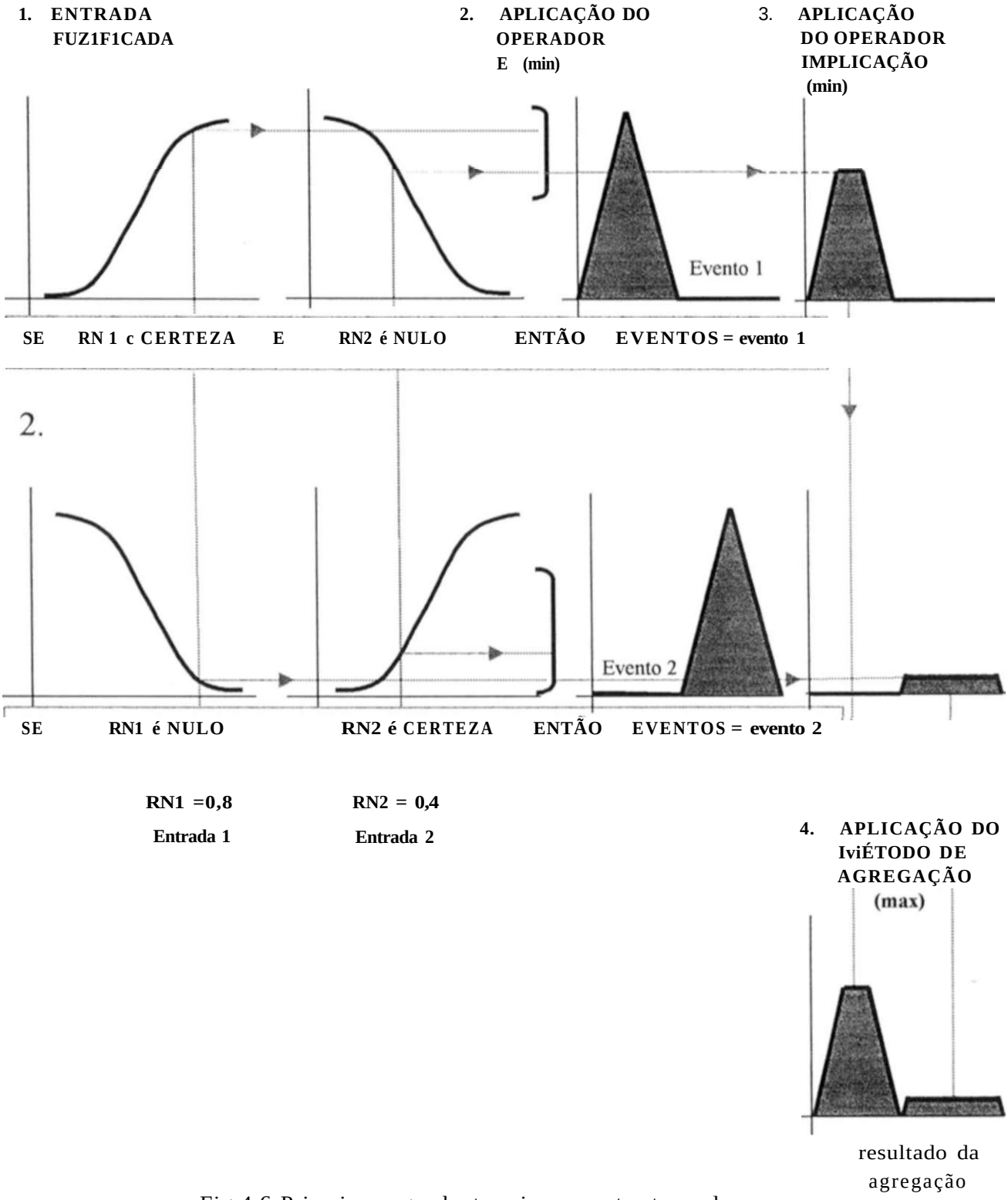


Fig.4.6 Primeira, segunda, terceira e quarta etapas do processo

Etapa **5.** Defuzificação

A entrada para o processo de defuzificação é um conjunto fuzzy (a agregação dos conjuntos fuzzy obtida na etapa **4)** e a saída é um único valor numérico (valor discreto recuperado desse conjunto fuzzy). A fuzificação ajuda na avaliação das regras durante as etapas intermediárias, porém o que geralmente se deseja é uma saída final discreta. Logo, métodos são empregados para transformar um conjunto fuzzy em um valor numérico representativo do problema

O método mais popular utilizado no processo de defuzificação é o método do cálculo do centróide, o qual estabelece o centro da área sobre a curva (fig**.4.7).**

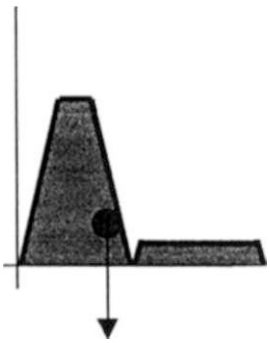


Fig**.4.7** Última etapa do processo

4.6 - VANTAGENS E DESVANTAGENS DA LÓGICA FUZZY

A partir dos resultados obtidos durante o estudo em [9] e confirmadas neste trabalho, destacou-se as seguintes potencialidades:

- Capacidade de representar incertezas inerentes ao conhecimento humano. Ou seja, os dados de entrada podem ser expressos de uma forma semelhante à linguagem (chance do relê de Iª zona "x" atuar muito grande ou chance do relê de sobrecorrente direcional de neutro "y" atuar pequena).
- Esta técnica tem a capacidade de explicar suas conclusões. O usuário pode verificar o processo de inferência da Lógica Fuzzy através do nível de ativação de cada regra quando uma determinada entrada é aplicada ao sistema.

Dentre as principais limitações pode-se citar:

- Em caso de ampliação do Sistema Elétrico ou mudança na topologia do mesmo, a máquina de inferência (base de conhecimento) da Lógica Fuzzy deverá ser revista.
- Não é capaz de generalizar, ou seja, só é capaz de responder ao que estiver escrito em sua base de conhecimento.
- Dificuldade na formação da base de conhecimento que deve ser adquirida através de declarações precisas dos especialistas.

4.7 - PRINCIPAIS CARACTERÍSTICAS DA LÓGICA FUZZY E DAS REDES NEURAIS

A tabela 4.1 mostra de forma resumida as principais características das Redes Neurais e da Lógica Fuzzy.

Tab. 4.1 Principais características das Redes Neurais e da Lógica Fuzzy

	LÓGICA FUZZY	REDES NEURAIS
PROCESSO	INFERÊNCIA	APRENDIZADO
CONHECIMENTO	ESTRUTURADO	NÃO-ESTRUTURADO
ALGORITMO	HEURÍSTICO	ESTATÍSTICO
COMPUTAÇÃO	NUMÉRICA	NUMÉRICA
PROCESSAMENTO DE DADOS	PARALELO	PARALELO
RESULTADOS	DEDUTIVO	INDUTIVO

Processo

A Lógica Fuzzy é um sistema que infere conseqüência a partir de premissas previamente estabelecidas nos Conjuntos de Regras, enquanto que as Redes Neurais implementam um sistema que adquire conhecimento a partir da apresentação de padrões entrada-saída (aprendizagem).

Conhecimento

Os sistemas fuzzy codificam um conhecimento estruturado, mas em uma forma numérica. Entra-se a associação fuzzy **(pesado, mais demorado),** onde os termos **pesado** e **mais demorado** referem-se as variáveis **tráfego** e **tempo do sinal verde** respectivamente, como uma entrada única em uma matriz de regras - FAM. Cada entrada define uma regra de Memória Associativa Fuzzy (FAM) ou uma transformação entrada-saída. O pêndulo invertido, um problema de controle clássico, esclarece de maneira mais clara como funciona o banco de regras FAM (Vide

apêndice A).

Ao contrário do Sistema Fuzzy, que apresenta conhecimento estruturado, a Rede Neural não explica porque obteve uma resposta, pois a representação do conhecimento está distribuída na forma de pesos das conexões entre os neurônios.

Algoritmo

A Lógica Fuzzy é construída por um conjunto de regras heurísticas de controle que são formuladas com base no **conhecimento especialista** de um operador humano para se obter a melhor estratégia de controle. Esta coleção de lógicas de controle é chamada de algoritmo de controle fuzzy ou regras de controle.

A saída de uma Rede Neural tem como base a análise de dados do seu conjunto de treinamento. Com base nesses dados é que a Rede Neural cria seus próprios métodos de tirar conclusões e fazer ilações ou predições. A conclusão disso é que ela aprende da experiência com amostras de dados. Tornando-se capaz de estimar a função sem um modelo matemático de como a saída depende da entrada.

Computação e Processamento de dados

Os sistemas fuzzy **raciocinam** com inferência associativa e paralela. Quando questionada ou lhe fornecida uma entrada, o sistema fuzzy dispara cada regra fuzzy em paralelo, mas em diferentes intensidade para inferir uma conclusão ou saída.

A plataforma numérica das redes neurais deve-se principalmente aos seus algoritmos numéricos. E através desses algoritmos que os pesos das conexões são atualizados de forma distribuída e paralela.

Conclusão: os sistemas neurais e fuzzy codificam informações amostrais em uma plataforma paralela-distribuída e de computação numérica. Isto permite que esses sistemas possam ser implementados em circuitos digitais e VLSI analógico.

Resultados

A lógica Fuzzy fornece uma saída baseada em premissas previamente estabelecidas nos Conjuntos de Regras, ao passo que a Rede Neural estabelece sua saída com base no conhecimento de certo número de dados singulares.

CAPÍTULO 5

APLICAÇÃO DAS TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL NA CONCEPÇÃO DO SISTEMA HÍBRIDO

5.1 - INTRODUÇÃO

Visando implementar um programa de controle a distância de suas subestações, a Companhia Hidrelétrica do São Francisco (CHESF) colocou em desenvolvimento um projeto piloto de teleassistência para a subestação da Mirueira (MRR), a qual atende a área metropolitana do Grande Recife no nível 69kV e supre outras cargas do Sistema Regional Leste. Esta subestação (SE) terá sua assistência remota centralizada no Centro Regional de Operação Leste (CROL) e será automatizada através da modernização do sistema de medição, proteção, comando, controle e supervisão convencional já em operação, substituição da Unidade Terminal Remota (UTR) e implantação de um Sistema Digital (SD) que incorpore funções de automatismo local.

A SE da Mirueira é não abrigada, com arranjo do tipo barra principal e de transferência para os setores 230kV e 69kV, e com configuração de acordo com o diagrama unifilar simplificado do anexo I. Por tratar-se da primeira subestação contemplada por este programa, esta subestação foi selecionada para o estudo com um sistema híbrido inteligente - Redes Neurais e Lógica Fuzzy.

Os sistemas Elétricos de Potência evoluíram ao longo do tempo, tornando-se grandes e envolvendo níveis de tensão de transmissão cada vez mais elevados. Em conseqüência disso o número de alarmes cresce em função do porte do sistema, e são acionados simultaneamente para identificar um determinado tipo de problema. Com a finalidade de diminuir esse número excessivo de alarmes a serem transmitidos ao CROL/CHESF, o qual deverá abrigar não apenas os alarmes desta, mas futuramente os alarmes de todas suas outras subestações a serem contempladas pelo projeto de teleassistência, foi realizada uma filtragem dos sinais de alarmes da SE Mirueira.

5.2 - FUNÇÃO DA REDE NEURAL E DA LÓGICA FUZZY

Os Sistemas Inteligentes podem tornar mais rápida e segura a detecção e o diagnóstico de falhas, especialmente quando múltiplos alarmes do Sistema de Supervisão são acionados simultaneamente.

A função da Rede Neural (RN) neste trabalho é efetuar uma filtragem em um conjunto constituído por um elevado número de sinais de alarmes, denominado de conjunto de

alarmes de entrada. O produto dessa filtragem é um conjunto menor de sinais, cujos elementos são denominados de eventos.

Diante das vantagens que as RN apresentam, principalmente quanto a capacidade de generalização, considerou-se a RN como um sistema de pré-seleção do evento.

A função do Sistema de Inferência Fuzzy é corrigir todos os resultados que porventura venham a ser classificados erroneamente pela Rede Neural, tornando o sistema, agora denominado de Sistema Híbrido um sistema mais confiável.

A Lógica Fuzzy é uma técnica de inteligência artificial que tem como principal característica a manipulação de dados com um determinado grau de incerteza. Através da utilização da Teoria de Conjuntos Fuzzy [13], as incertezas inerentes aos dados de um sistema podem ser representados através de expressões lingüísticas comumente utilizadas.

Além disso, uma outra característica importante é que, assim como em Sistemas Especialistas, um conjunto de regras dotadas de pesos, deve ser definido para se inferir o valor das saídas de um determinado sistema.

A partir de estudos utilizando o Sistema de Inferência Fuzzy constatou-se que o único meio para que o mesmo fosse capaz de classificar **corretamente** um evento classificado **erroneamente** pela Rede Neural seria através da atualização dos pesos das regras em função do estado dos alarmes. Assim, o Sistema de Inferência Fuzzy faria a classificação final, admitindo como entrada a saída da Rede Neural, denominada de pré-classificação do evento, e fornecendo como saída a classificação final do evento. A figura 5.1 mostra esquematicamente o funcionamento do Sistema

Híbrido Inteligente - Rede neural e Lógica Fuzzy utilizado neste trabalho

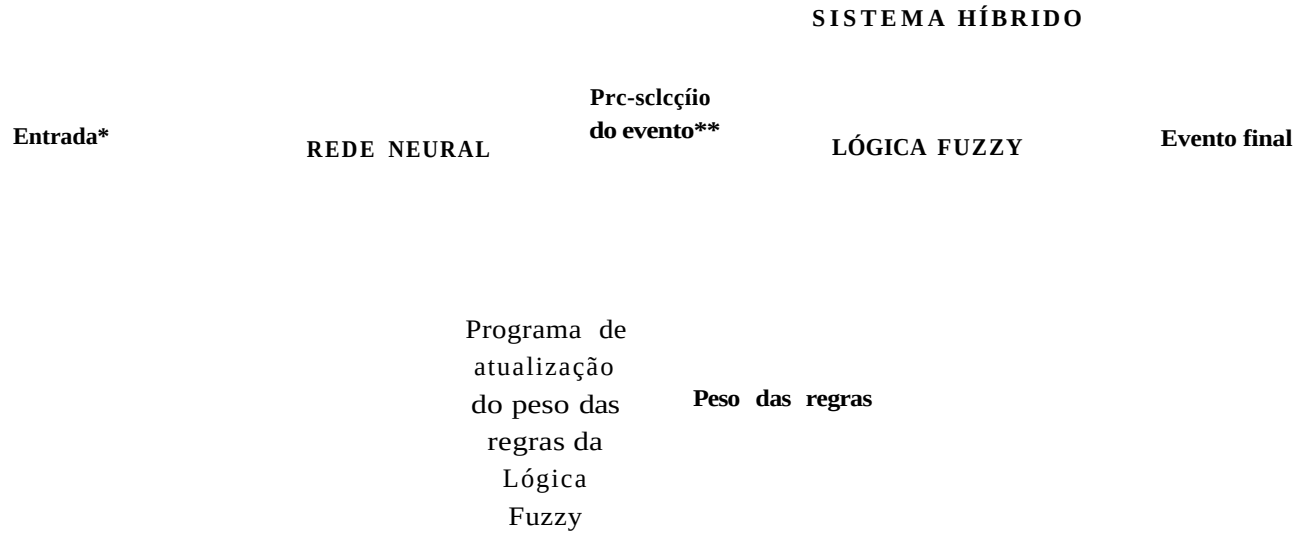


Fig. 5.1 Diagrama esquemático do Sistema Híbrido Inteligente

* Estado dos alarmes indicado por um vetor de elementos zero e um (alarme desativado e ativado respectivamente).

** Saída obtida representativa do código do evento indicado no anexo II.

5.3. - AS REDES NEURAIS E SEU TREINAMENTO

A ferramenta utilizada para treinamento da rede, o NeuroLab [8, 14], é composta por dois módulos:

1) SiReNe - Simulador de Redes Neurais

2) SeReNa - Servidor de Redes Neurais para Aplicações

O primeiro módulo, o simulador propriamente dito, é utilizado em ambiente Windows e oferece recursos e facilidades para criar, configurar, testar e salvar redes neurais para

posterior utilização por aplicações convencionais. O segundo módulo é o servidor de redes neurais que se encarrega de disponibilizar de forma transparente a rede neural gerada pelo SiReNe para aplicações convencionais sob a forma cliente servidor.

Neste trabalho utilizou-se apenas o primeiro módulo - SiReNe, visto que a saída da Rede Neural deve ser conduzida para a entrada do Sistema de Inferência Fuzzy, como mostrado no diagrama esquemático da fig.5.1.

5.3.1 - Preparação do modelo neural

Nos últimos anos, as abordagens de RNA em diagnóstico de falhas em Sistema de Potência têm tirado proveito das vantagens citadas no capítulo 3. Todavia as RNAs empregam os padrões de treinamento derivados simplesmente do estado operacional dos disjuntores e relés. Conforme Kim [15], esta técnica acarreta uma carga severa de trabalho na tarefa de treinamento quando aplicada a um Sistema de Potência complexo de grande porte devido ao elevado número de conexões das Redes Neurais. Essa elevação no número de conexões advém do aumento do número de padrões de falhas e de equipamentos dentro do sistema. Conclui-se desta forma que esta técnica é ineficiente quando aplicada em Sistemas de Potência de grande porte.

Para minimizar esse problema, o Sistema de Potência foi dividido em 4 módulos: Módulo Linha de Transmissão em 230kV (antiga) - LT-04C4; Módulo Linha de transmissão em 230kV (nova) - LT-04F3; Módulo Transformadores - TR-04T1 e Módulo Linha de Transmissão em 69kV - LT-02J1. Cada módulo tem sua Rede Neural representativa, permitindo desta forma construir padrões de treinamento com mais facilidade para estas redes do que para uma única rede

representativa de todo o sistema. Além disso, cm caso de mudança no sistema parcial, um pequeno número de RNAs precisa ser treinada novamente.

Para a preparação de cada uma dessas redes, são necessárias as definições de sua arquitetura e do seu conjunto de dados (exemplos) de treinamento. Além disso, alguns parâmetros precisam ser definidos para o treinamento da rede, tais como: coeficiente de aprendizado, momento e ruído. E necessário que sejam realizados testes com a rede, para avaliar se está pronta para resolver o problema ou não. Os resultados desses testes são apresentados através de dois gráficos. Um gráfico que mostra, nos neurônios da camada de saída da rede, as saídas obtidas com o processamento de cada exemplo de entrada (grupo de entrada) do conjunto de dados representativo do problema, e as saídas desejadas para cada uma dessas entradas. O outro gráfico mostra o erro (diferença entre a saída obtida e as desejada) nos neurônios da camada de saída da rede, obtido com o processamento de cada exemplo de entrada do conjunto de dados representativo do problema.

5.3.2 - Geração dos exemplos de treinamento

Através de estudos preliminares realizados por Engenheiros da CHESF foram estabelecidos os alarmes que, em função da configuração mínima de alarmes necessária para prover os meios para a coordenação da operação e da manutenção do sistema elétrico, deveriam chegar até o CROL/CHESF e foram montados diagramas de blocos para cada um dos módulos da SE. Os diagramas de blocos (Anexo II) montados são nomeados pelos nomes dos módulos que podem ser localizados no diagrama unifilar simplificado do anexo I : LT-04C4, LT-04F3, TR-04T1 e LT-02J1.

Analisando o diagrama de blocos, de cada um dos módulos, e o grupo de alarmes, que através da porta OU causaria a atuação de um único alarme (evento), gerou-se a base de dados

para treinamento e teste. Cada um dos relés no conjunto de entrada c representado pelo número um (1), significando relê ativado ou zero (0) desativado. E as saídas (eventos) são representadas por códigos de ls e Os com 2 ou 4 dígitos dependendo do módulo a ser tratado

Para geração dos padrões ou exemplos de treinamento e teste, utilizou-se o programa Excel para gerar aleatoriamente números ls e Os indicadores do estado de cada um dos relés ligados pela lógica OU que fornecesse o evento correspondente estabelecido no diagrama de blocos (Anexo II). Para maiores esclarecimentos verificar a tabela 5.1 referente aos exemplos criados para treinamento da rede LT-02J1 (fig. 5.2). Pode-se observar que os alarmes 1 e 2 estão ligados por uma porta lógica OU e os alarmes 3 e 4, por uma outra porta lógica OU.

Tab.5.1 Exemplos Entrada / Saída para treinamento do módulo LT-02J1

	E N T R A D A					
Exemplo	Alarm **1**	Alarm2	Alarm3	Alarm4	S A I D A	
1	**1**	**0**	0	0	**1**	**0**
2	**0**	**1**	0	0	**1**	**0**
3	**1**	**1**	0	0	**1**	**0**
4	0	0	**1**	**0**	**0**	**1**
5	0	0	**0**	**1**	**0**	**1**
6	0	0	**1**	**1**	**0**	**1**

Tab. 5.2 Números de alarme de cada módulo, números de eventos e números de dígitos do código do evento que definem os números de neurônios de entrada e saída da rede neural.

Módulo	Números de alarmes de entrada	Números de eventos	Números de dígitos do evento
LT-04C4	19	8	4
LT-04F3	32	11	4
TR-04T1	28	15	4
LT-02J1	4	2	2

A tabela 5.2 mostra o número de alarmes de entrada, número de eventos e número de

dígitos associados a estes eventos. Estes parâmetros são de grande importância para definir a arquitetura da rede de cada um dos módulos. Por exemplo, o número de alarmes de entrada fixa o número de neurônios na camada de entrada da rede, e o número de dígitos do código do evento estabelece o número de neurônios na camada de saída.

Deve-se notar que o número de neurônios das camadas de entrada e saída dependem da aplicação específica do sistema a ser analisado e que a determinação do número de neurônios das camadas intermediárias é empírico, obtido através da análise do desempenho da rede neural.

5.3.3 - Tamanho do conjunto de treinamento

Sabendo-se que o conjunto de treinamento é de fundamental importância para um bom desempenho da rede neural, pois é nesta fase e através deste conjunto que é descoberto o mapeamento da rede, foi escolhido um conjunto de treinamento que contivesse todas as características das classes do problema. Para uma boa representatividade deste conjunto é de fundamental importância o entendimento do problema a ser resolvido.

A relação entre o tamanho do conjunto de treinamento e o número de pesos na rede neural é importante para um treinamento adequado Se o número de exemplos de treinamento é menor que o número de pesos, pode-se esperar que a rede atribua um peso para cada exemplo de treinamento [16]. Isto provavelmente produzirá uma pobre generalização, isto é, a habilidade de associar exemplos não vistos às classes definidas pelos exemplos de treinamento. Recomenda-se que o número de exemplos de treinamento seja no mínimo o dobro do número de pesos da rede, porém nem sempre isto é possível face às dificuldades em se adquirir tais dados. No problema em

questão isto ocorre devido a impossibilidade de gerar mais exemplos por causa da pequena quantidade de alarmes associados pela porta OU a um determinado evento. Para melhor esclarecimento verifique o diagrama da fig 5.2 abaixo referente ao módulo LT-02J1.

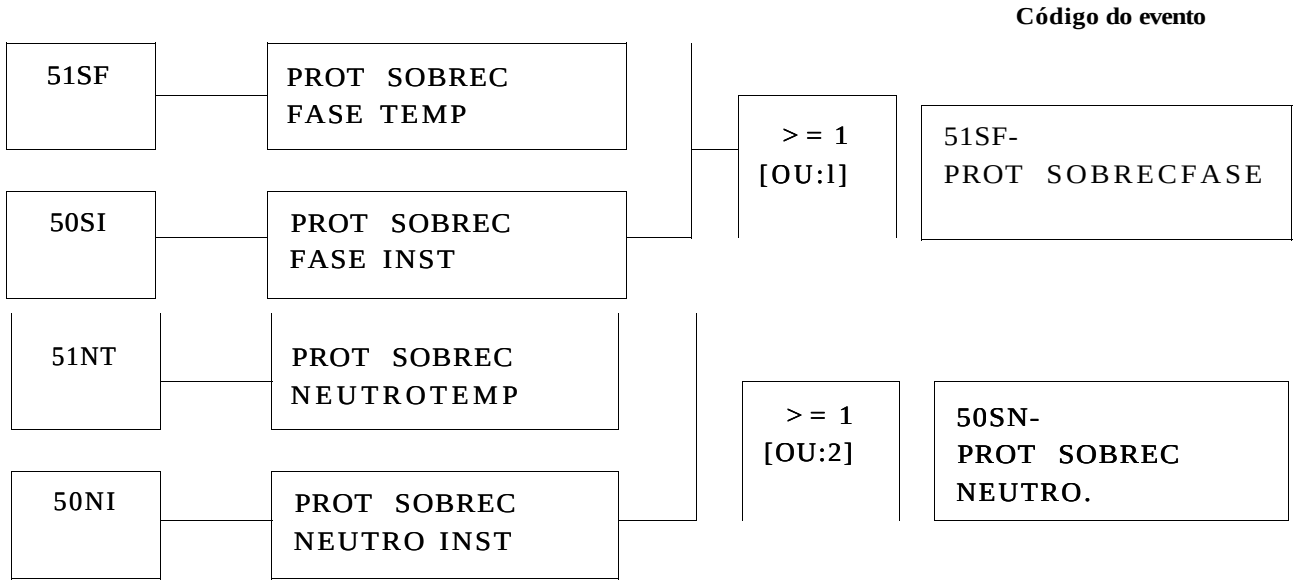


Fig. 5.2 Diagrama esquemático dos alarmes relacionados ao módulo LT-02J1

A tabela 5.3 mostra os números de exemplos associados a cada rede neural

Tab. 5.3 Números de exemplos de treinamento e teste fornecidos à rede neural.

Rede Neural	Números de exemplos de treinamento	Números de exemplos de teste	Total de exemplos
LT-04C4	550	66	616
LT-04F3	1008	120	1128
TR-04T1	180	33	213
LT-02J1*	6	0	6

* Esta rede não possui exemplos de teste, visto que todos os estados possíveis dos seus quatros alarmes de entrada para uma única saída (10 ou 01) estão representados por estes seis exemplos.

5.3.4 - Arquitetura da Rede Neural

O tamanho de uma rede neural é estabelecido por tentativa. O número de neurônios (elementos de processamento) na camada intermediária (escondida) está associada com a habilidade de mapeamento da rede. Quanto maior a rede, mais poderosa ela é. Entretanto, crescendo continuamente o tamanho da rede, haverá um ponto a partir do qual a generalização começará a piorar. O termo generalização refere-se à capacidade que a rede adquire após ter sido treinada, de reconhecer padrões que não lhe foram apresentados no conjunto de treinamento. Esta piora na generalização deve-se ao fato de estar utilizando um número de parâmetros de ajuste (pesos) muito grande para o conjunto de treinamento. Como conseqüência a rede memoriza os padrões que lhes foram apresentados no conjunto de treinamento e responde de forma imprevisível aos padrões nunca visto antes (conjunto de teste). A solução é encontrar o menor grau de liberdade que alcance o desempenho requerido no conjunto de teste.

A prática recomenda que se inicie com uma rede pequena e aumente-se seu tamanho até que seja alcançado um bom desempenho no conjunto de TESTE.

5.3.5 - Construção das Redes no Simulador - SiReNe

Definido o banco de dados (exemplos de treinamento e teste) utilizando o aplicativo Excel, pode-se criar o modelo da rede no SiReNe. Ao executar o programa SiReNe.exe aparecerá uma tela, onde no menu principal e no item arquivo/novo/BP3, o sistema solicita que o usuário selecione qual será o banco de dados e a tabela desse banco de dados utilizado pela nova rede. Após selecionada a tabela, a tela de "Dados" é aberta automaticamente. Nela é possível visualizar os

dados c definir o número de entrada e saída da rede, os quais estabelecem os neurônios de entrada e saída respectivamente.

Na tela principal do modelo backpropagation, representada na figura 5.3, escolhe-se o número de neurônios na camada intermediária e a função de ativação entre as camadas.

Definido a arquitetura da rede, o próximo passo é treiná-la. Na tela, fig. 5.3, ao selecionar o botão treinamento e teste aparecerá a janela apresentada na figura 5.4a. Esta janela é composta de dois ambiente: o de treinamento (fig. 5.4a) e o de teste (fig. 5.4b). No treinamento, define-se alguns parâmetros importantes como taxa de aprendizagem, momento e ruído, cujo objetivo é acelerar o treinamento e reduzir o risco de oscilação. No ambiente de teste seleciona-se a saída para a qual deseja-se ver o gráfico e o tipo de gráfico: o gráfico que mostra a saída desejada e obtida (fig. 5.4b); e o gráfico da diferença (erro) entre a saída desejada e a saída obtida para cada ponto do eixo horizontal que corresponde a um exemplo do banco de dados (fig. 5.4c).

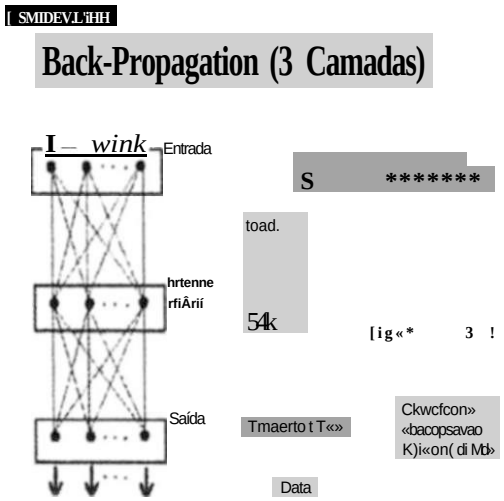


Fig. 5.3 Tela principal do SiReNe, onde pode-se visualizar a arquitetura da rede LT-02J**1.**

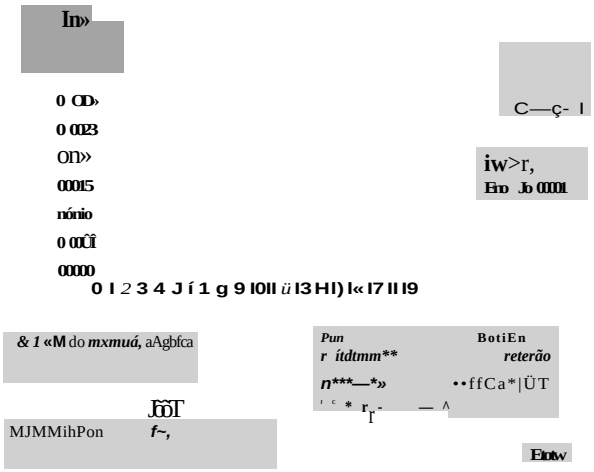


Fig. 5.4a Janela de treinamento e teste

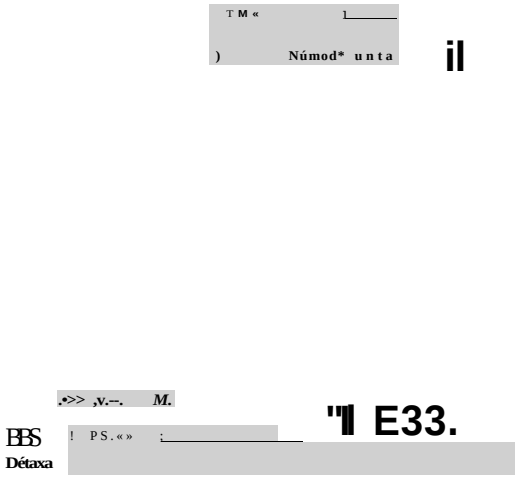


Fig. 5.4b Gráfico da saída desejada c obtida

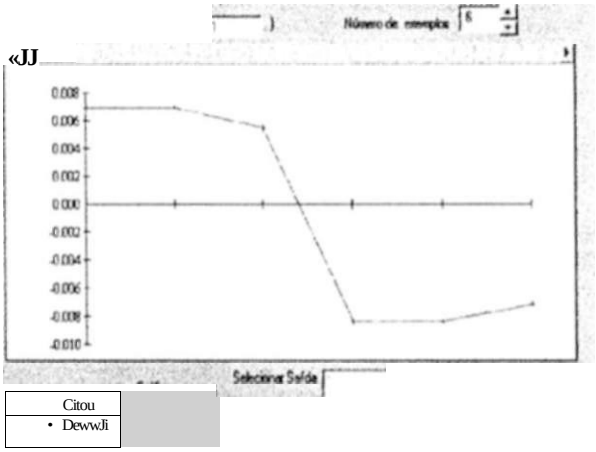


Fig. 5.4c Gráfico do erro entre a saída desejada e a obtida

O erro RMS, mostrado nas telas das figuras 5.4b e 5.4c, c calculado através da equação (5**.1).**

$$ms=-YJ\hat{E} \qquad (5.1)$$

onde ***n*** é o número total dc padrões (exemplos) e E_p é somatório do quadrado da diferença entre a saída real e a desejada dividida por 2 para todos os padrões a serem aprendidos, ou seja,

$\frac{1}{H}^{k}$, onde ***k*** é o número total de neurônios na camada de saída.

Na tabela 5.4 pode-sc verificar a arquitetura final das redes, após várias simulações, de acordo com a variação do número de neurônios na camada intermediária, a alteração da característica da função de ativação entre as camadas e a especificação dos parâmetros com os seguintes valores :

- Momento : variando manualmente entre 0,01 e 0,9
- Coeficiente de aprendizagem : adaptativa
- Ruído : 0
- Forma de treinamento : todos os exemplos (batch)

Este conjunto de experimentos teve como objetivo indicar a configuração da rede que apresentasse melhor desempenho na classificação dos eventos.

Tab. 5.4 Arquitetura das Redes Neurais

REDE NEURAL*	NÚMEROS DE NEURÔNIOS			DESCRIÇÃO DOS MÓDULOS
	Camada de Entrada	Camada Intermediária	Camada de Saída	
LT-04C4	19	10	4	Linha de Transmissão 230KV de Recife II (antiga)
LT-04F3	32	4	4	Linha de Transmissão 230KV de Goianinha (nova)
TR-04T1	28	4	4	Transformador 1000MVA 230/69KV
LT-02J1	4	1	2	Linha de Transmissão 69K.V para Cia Tecidos Paulista

* Todas as redes são do tipo MLP (Backpropagation) com três camadas, c com as funções de ativação da forma sigmoide entre suas camadas.

5.3.6 - Critério de parada

Após a definição da arquitetura, do tamanho do conjunto de treinamento e dos exemplos representativos do problema, a próxima etapa é definir quando parar o treinamento da Rede Neural.

Todos os critérios dc parada são baseados no monitoramento do erro médio quadrático (MSE). A curva do MSE como função do tempo é chamada de curva dc aprendizagem. O critério mais usado é provavelmente o da escolha do número dc iterações (passos), mas pode-se alternativamente apresentar o erro final como critério de parada. Estes dois métodos têm falhas, visto que alcançando-se o número de iterações ou o erro final estipulado pode-se não se ter alcançado o desempenho desejado no conjunto dc Teste.

Um outro critério utilizado na prática é observar o erro entre duas iterações consecutivas e caso o erro não caia significativamente ficando quase que constante, o treinamento deve ser finalizado.

Uma outra possibilidade é monitorar o RMS para o conjunto de teste, como na validação cruzada. Deve-se parar o aprendizado quando o erro no conjunto de teste começar a crescer (fig.5.5). E quando estabelece-se a máxima generalização

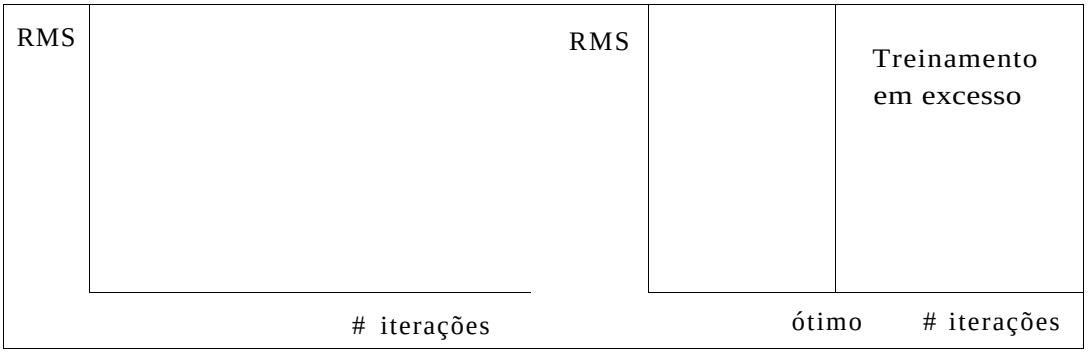


Fig. 5.5 Comportamento do RMS para o conjunto de treinamento e teste respectivamente.

Para implementar este procedimento deve-se treinar a rede durante um certo número de iterações, congelar os pesos e testar o desempenho no conjunto de teste. Então, retornar ao

conjunto dc treinamento c continuar o treinamento caso o desempenho desejado não tenha sido alcançado e o erro continue caindo.

Neste trabalho utilizou-se este último critério de parada por ser um dos métodos mais eficientes atualmente empregado para finalização do treinamento [16].

Após o treinamento, foi realizado um teste no conjunto de exemplos dc treinamento e teste, observando-se um alto percentual de acerto das redes.

5.4 - CONCEPÇÃO DO SISTEMA DE INFERÊNCIA FUZZY

A ferramenta utilizada na concepção do Sistema de Inferência Fuzzy [17], provê os recursos para criar e editar o sistema dentro de uma plataforma do MatLab. A Caixa de Ferramentas da Lógica Fuzzy fornece um número de ferramentas iterativas que permite o acesso a muitas funções através da interface gráfica com o usuário. Juntas, estas interface dispõem de um ambiente para a montagem da arquitetura, implementação e análise do Sistema de Inferência Fuzzy.

As quatro janelas mostradas a seguir (figuras 5.6a, 5.6b, 5.6c, 5.6d) referem-se ao Sistema de Inferência Fuzzy montado para o módulo LT-02J1.

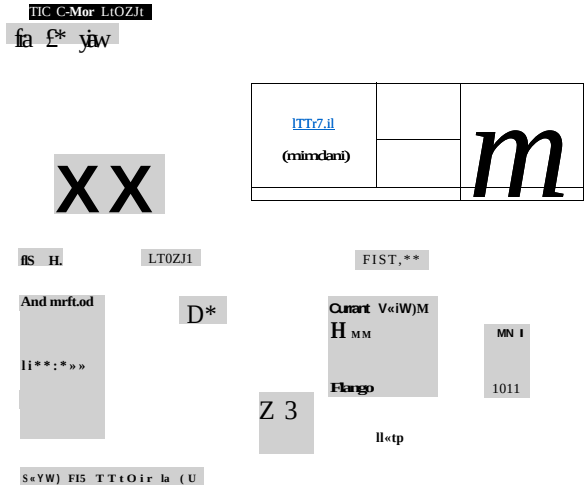


Fig. 5.6a Janela 1 - Editor FIS

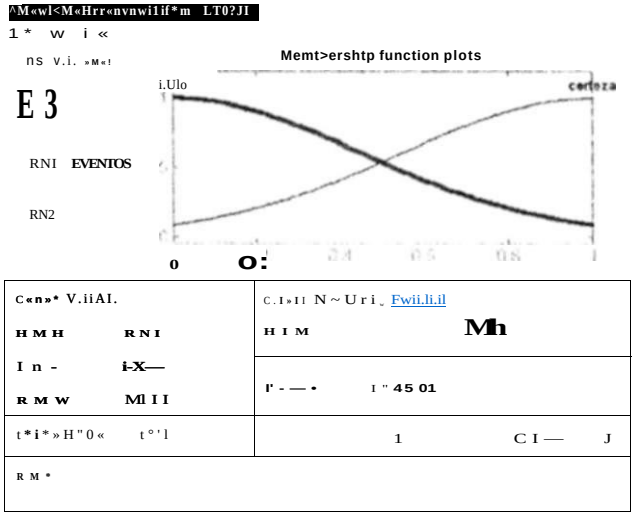


Fig. 5.6b Janela 2 Editor das funções membros

Na janela 1 (Fig. 5.6a), deve-se selecionar primeiro no item File do menu principal o tipo do processo de inferência. Caso contrário, o programa assume automaticamente o tipo Mamdani. Em seguida, seleciona-se no menu principal o item Edit e adiciona-se a quantidade de variáveis de saída, e na tela nomeia-se essas variáveis c seleciona-se os processos das lógicas "E" c "OU", da implicação, agregação c defuzificação.

Na janela 2 (Fig.5.6b), seleciona-sc no menu principal o item Edit/Add MFs...(adicionar funções membros), a forma e a quantidade de funções membros de cada uma das variáveis de entrada e saída definidas na janela 1. Em seguida, define-se na tela o campo de valores das variáveis de entrada e saída, nomeia-se cada uma das funções membros e define-se seus parâmetros. E possível, caso se deseje, selecionar na tela outra forma para a função membro definida no item Edit.

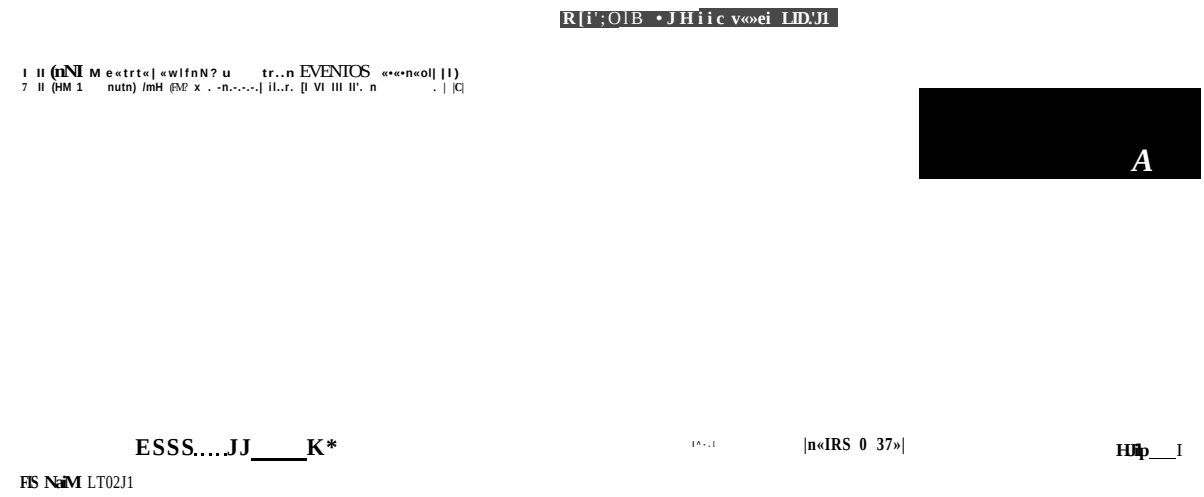


Fig. 5.6c Janela 3 - Editor dc regras

Fig. 5.6d Janela 4 - Visualizador elc regras

Na janela 3 (fig. 5.6c) deve-se escrever as regras com seus respectivos pesos (0-1) Rsses pesos aparecem entre parênteses após a escrita de cada uma das regras Caso não sejam fornecidos, o programa assume automaticamente o valor **I** para todas as regras O valor zero estabelecido para a regra significa que, apesar dela estar escrita, ela não sofre o processo de implicação e, consequentemente, não contribui para processos dc agregação e defuzificação, Isto c visualizado na janela 4, observando-se que a área correspondente à regra 2 da variável de saída (eventos) não foi preenchida, significando que a mesma não sofreu o processo dc implicação, em virtude do valor zero atribuído ao peso da regra na janela 3.

Para que o sistema apresentasse uma saída satisfatória, isto é, classificasse corretamente os eventos mal classificados pela rede neural, foram executados vários experimentos que consistiam na substituição do tipo da função membro das variáveis de entrada e saída, modificação de seus parâmetros c alteração dos pesos das regras. F.sta experimentação foi realizada apenas para o módulo LT-02.I1 por este ser um módulo pequeno, facilitando portanto a análise. Ele possui apenas dois eventos dc saída (fig. 5.2) representados por dois neurônios na saída da Rede Neural (tab.5.4), cujos valores são dados de entrada para o sistema fuzzy. Após estes experimentos

usando os oito tipos de funções membros fornecidos no programa, concluiu-se que qualquer uma das funções pode ser empregada, desde que seus parâmetros sejam devidamente estabelecidos: os parâmetros da curva da função membro do termo lingüístico **nulo** é tal que ela seja decrescente no intervalo (0-1) e em 1 seu valor seja diferente de zero - //,(!) **5*** 0 (fig.5.7a); e os parâmetros da curva para a função membro do termo linguistico certeza é tal que ela seja crescente nesse intervalo e em 0 seu valor seja diferente de zero - //₂(0) 0 (fig.5.7b). A função gaussiana foi a função membro escolhida para os termos lingüísticos das variáveis de entrada , por ela ser uma das funções freqüentemente empregadas em trabalhos desta natureza.

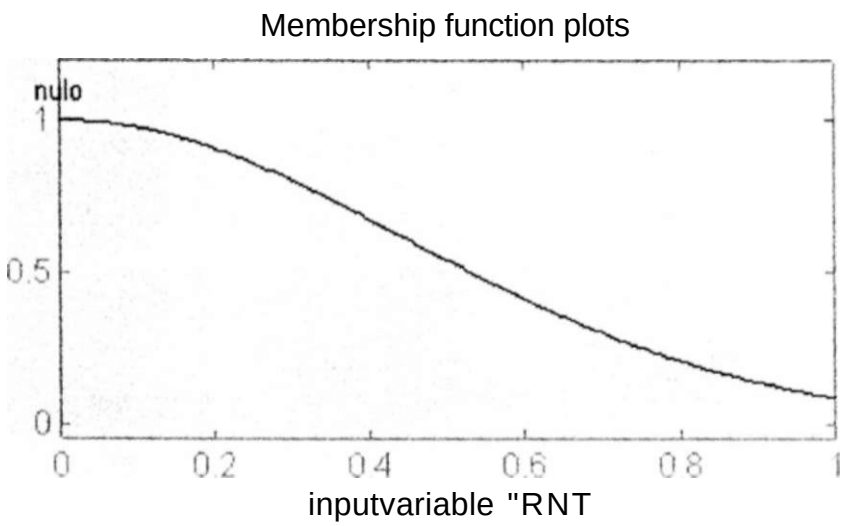


Fig. 5.7a Curva da função membro do termo lingüístico nulo para a variável de entrada RN1.

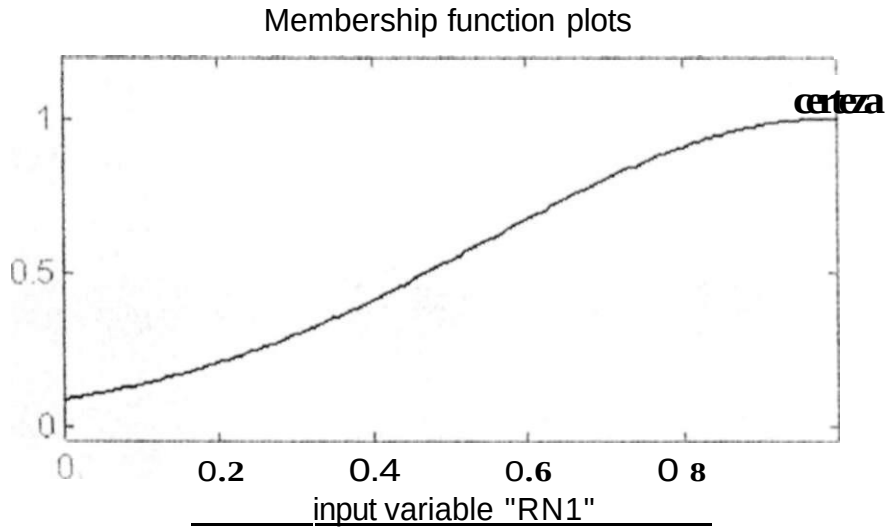


Fig. 5.7b Curva da função membro do termo lingüístico certeza para a variável de entrada RN1

Com relação a função membro dos termos lingüísticos da variável de saída, qualquer função membro também pode ser utilizada, porém essas funções não se devem sobrepor. Isto é possível através do ajuste dos parâmetros da função de cada um dos termos lingüísticos. Cada intervalo, cuja largura é 1 define um único evento (fig.5.8). Para compor a função membro dos termos da variável de saída preferiu-se utilizar a função triangular, também por ser essa um outro tipo de função bastante difundido em trabalhos correlatos.

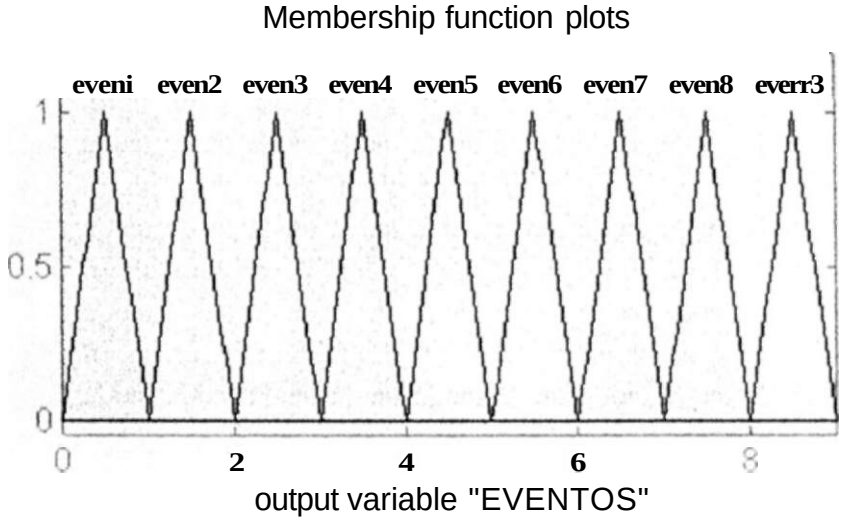


Fig. 5.8 Curvas das funções membros dos termos lingüísticos para a variável de saída EVENTOS.