

**MODELOS ATRAVÉS DE REDES NEURAIAS
ARTIFICIAIS SEM REALIMENTAÇÃO PARA
DISPOSITIVOS E CIRCUITOS DE RF/MICROONDAS**

Paulo Henrique da Fonseca Silva

Tese submetida ao
Colegiado dos Cursos de Pós-Graduação em Engenharia Elétrica
da Universidade Federal da Paraíba,
como parte dos requisitos para a obtenção do título de
Doutor em Ciência no domínio da Engenharia Elétrica.

Área de Concentração:
Processamento de Sinais e Eletromagnetismo Aplicados

Orientador:
Prof. Marcos Antonio Barbosa de Melo, Dr.

Campina Grande, Março de 2002.



S586m Silva, Paulo Henrique da Fonseca
Modelos através de redes neurais artificiais sem
realimentação para dispositivos e circuitos de
RF/microondas / Paulo Henrique da Fonseca Silva. - Campina
Grande, 2002.
166 p. : il.

Tese (Doutorado em Engenharia Elétrica) - Universidade
Federal da Paraíba, Centro de Ciências e Tecnologia.

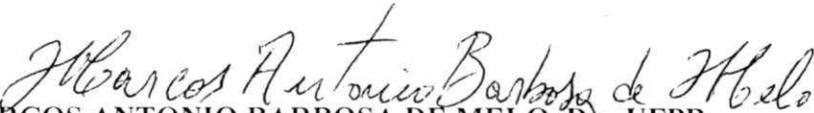
1. Dispositivos de Microondas 2. Circuitos Não-Lineares
3. Redes Neurais Artificiais 4. Tese I. Melo, Marcos
Antonio Barbosa de, Dr. II. Universidade Federal da Paraíba
- Campina Grande (PB) III. Título

CDU 004.032.26(043)

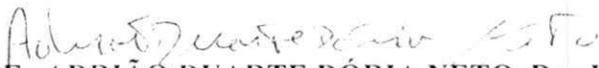
MODELOS ATRAVÉS DE REDES NEURAIS ARTIFICIAIS SEM
REALIMENTAÇÃO PARA DISPOSITIVOS E CIRCUITOS DE RF/MICROONDAS

PAULO HENRIQUE DA FONSECA SILVA

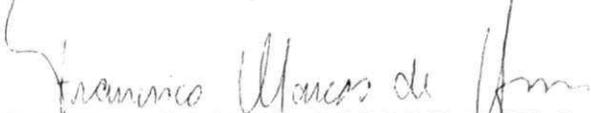
Tese Aprovada em 22.03.2002

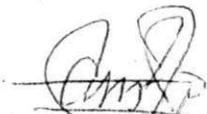

PROF. MARCOS ANTONIO BARBOSA DE MELO, Dr., UFPB
Orientador

PROF. EVANDRO CONFORTI, Dr., UNICAMP
Componente da Banca (Ausente, enviando parecer por escrito)


PROF. ADRIÃO DUARTE DÓRIA NETO, Dr., UFRN
Componente da Banca


PROF. GERVÁSIO PROTÁSIO DOS SANTOS CAVALCANTE, Dr., UFPA
Componente da Banca


PROF. FRANCISCO MARCOS DE ASSIS, Dr., UFPB
Componente da Banca


PROF. FRANCISCO DE ASSIS FERREIRA TEJO, D.Sc., UFPB
Componente da Banca

CAMPINA GRANDE – PB
Março - 2002

DEDICATÓRIA

*A Dona Vilma, minha mãe,
a minha esposa Danielle
e ao meu irmão Carlos Magnus.*

AGRADECIMENTOS

Ao professor Marcos Barbosa, pela excelente orientação;

A Danielle, por sua compreensão e paciência, por incentivar esta realização;

A minha mãe, Maria Vilma, por sua ajuda, seu estímulo;

Ao professor Adrião Duarte, que muito contribuiu nesta pesquisa;

Aos professores e funcionários da COPELE/DEE;

À CAPES, pelo apoio financeiro, através de bolsa, durante estes quatro anos;

Enfim, a todos que, direta ou indiretamente, tomaram parte neste feito;

Eu sou sinceramente grato.

SUMÁRIO

AGRADECIMENTOS	vii
LISTA DE FIGURAS	xiii
LISTA DE SÍMBOLOS	xix
LISTA DE SIGLAS	xxiii
RESUMO	xxv
ABSTRACT	xxvii
CAPÍTULO 1 INTRODUÇÃO	1
CAPÍTULO 2 REDES NEURAIS SEM REALIMENTAÇÃO	7
2.1 Introdução.....	7
2.1.1 Resumo do Método Utilizado.....	8
2.1.2 Modelos de Neurônio Artificial.....	9
2.2 Rede MLP.....	11
2.2.1 Configuração.....	11
2.2.2 Algoritmo da <i>Backpropagation</i>	12
2.2.3 Métodos Alternativos Para o Treinamento do MLP.....	14
2.2.4 Regra de Aprendizado Delta-Bar-Delta.....	17
2.2.5 Algoritmo SuperSAB.....	19
2.2.6 Algoritmo Rprop.....	20
2.3 Redes Neurais e Aproximação de Funções.....	22
2.4 Rede RBF.....	25
2.5 Rede <i>Wavelet</i> Polinomial.....	27
2.6 Rede de Fourier.....	30
2.7 Rede de Funções <i>Sample</i>	32
2.8 Redes Neurais e Aproximação de Funções	
– Resultados de Simulação.....	36

CAPÍTULO 3	MODELOS ATRAVÉS DE REDES NEURAIS BASEADAS EM RESULTADOS ELETROMAGNÉTICOS	43
3.1	Introdução.....	43
3.2	Filtros Distribuídos em Microfita.....	45
3.2.1	Rede MLP Para Projeto de Filtros Distribuídos	45
3.2.2	Rede MLP Para Análise de Filtros Distribuídos.....	47
3.3	Impedância de Antenas Isoladas.....	49
3.4	ANNs Para Dispositivos Planares	53
3.4.1	Resistência AC de uma Fita Metálica.....	53
3.4.2	Indutores Espirais	55
3.4.3	Transistores de Efeito de Campo	59
3.5	Linhas de Microfita Suspensa e Invertida.....	62
3.6	Substrato PBG como Filtro em Linhas de Microfita.....	63
3.7	Terminações Para Calibração de Analisadores de Redes.....	65
CAPÍTULO 4	MODELOS DE GaAs MESFETs ATRAVÉS DE REDES MLPs PARA ANÁLISE TRANSITÓRIA DE CIRCUITOS NÃO LINEARES	71
4.1	Introdução.....	71
4.2	Análise Transitória Não Linear	72
4.2.1	Análise DC Não Linear	72
4.2.2	Análise Transitória Linear	73
4.2.3	Análise Transitória Não Linear	79
4.3	Modelos Neurais Para GaAs MESFETs.....	89
4.4	Análise do Circuito Porta Lógica com GaAs MESFETs	93
4.5	Análise de Sensibilidade do Circuito Porta Lógica.....	96
CAPÍTULO 5	MÉTODO DO BALANÇO HARMÔNICO SIMPLIFICADO: REDES DE FOURIER	99
5.1	Introdução.....	99
5.2	Fundamentos do MBH	100
5.3	MBH Simplificado e Redes de Fourier	102
5.4	Aplicações do MBH Simplificado	103

CAPÍTULO 6	MODELOS ATRAVÉS DE REDES NEURAIAS BASEADAS NO CONHECIMENTO EMPÍRICO PARA CIRCUITOS DE RF/MICROONDAS	111
6.1	Introdução	111
6.2	Métodos de Modelagem Através de Redes Neurais Para Circuitos de RF/Microondas	112
6.2.1	Método Convencional	112
6.2.2	Método Híbrido EM-ANN	113
6.2.3	Método PKI	114
6.2.4	Método KBNN	114
6.2.5	Método SMN	115
6.2.6	Incluindo a Frequência no Método SMN	118
6.2.7	Um Método SM-ANN Alternativo.....	123
6.3	Mapeamento de Espaços Utilizando o MLP com Três Camadas....	123
6.3.1	Exemplos de Treinamento e Iniciação do MLP	124
6.3.2	Função de Ativação Sigmóide.....	127
6.4	Análise de uma Junção-T em Linha de Microfita Através de Métodos SM-ANN.....	128
CAPÍTULO 7	CONCLUSÕES	139
	REFERÊNCIAS	145
APÊNDICE	MÉTODOS ITERATIVOS PARA A MINIMIZAÇÃO DE FUNÇÕES	157
A.1	Método do Gradiente.....	158
A.2	Método do Gradiente Conjugado	158
A.3	Método de Fletcher-Powell	161
A.4	Método de Newton-Raphson.....	162
A.5	Método de Newton.....	163
A.6	Exemplo Numérico	164

LISTA DE FIGURAS

Fig. 2.1	(a) Método convencional (EM-ANN) para o desenvolvimento de modelos através de redes neurais. (b) Modelo através de rede neural.....	9
Fig. 2.2	Modelo do k -ésimo neurônio de uma rede neural sem realimentação.....	10
Fig. 2.3	Representação gráfica de redes sem realimentação. (a) Camada simples. (b) Quatro camadas.....	11
Fig. 2.4	Rede MLP com uma camada oculta.....	13
Fig. 2.5	Influência da escolha da taxa de aprendizado na convergência do algoritmo BP	15
Fig. 2.6	Rede RBF Gaussiana.....	25
Fig. 2.7	Elementos $\varphi_2, \varphi_3, \dots, \varphi_{10}$ pertencentes a família de <i>wavelets</i> polinomiais.....	28
Fig. 2.8	Rede neural baseada em <i>wavelets</i> polinomiais.....	29
Fig. 2.9	Rede de Fourier.....	31
Fig. 2.10	Reconstrução de um sinal a partir de suas amostras.....	32
Fig. 2.11	Formas da função <i>sample</i> (2.68), conforme variações dos parâmetros: (a) largura. (b) largura e centro.....	34
Fig. 2.12	Rede de funções <i>sample</i>	35
Fig. 2.13	Aproximação para (2.77) através da rede MLP, com o emprego de diferentes algoritmos de treinamento.....	37
Fig. 2.14	Evolução dos erros de treinamento.....	38
Fig. 2.15	Aproximação da Rede RBF através de funções RBFs Gaussianas.....	38
Fig. 2.16	Aproximação da rede de <i>wavelets</i> polinomiais para a função exemplo (2.77).....	39
Fig. 2.17	(a) Aproximação através da rede de Fourier : (o) conjunto de treinamento, (—) generalização da rede de Fourier. (b) convergência do algoritmo de treinamento (10 neurônios ocultos).....	40
Fig. 2.18	Aproximações da série de atividade solar através de redes RFS.....	41
Fig. 2.19	Erro durante o treinamento da rede RFS, conforme o número de neurônios ocultos...	42
Fig. 3.1	Resultados de projetos de filtros distribuídos para $f_c = 0,5$ GHz e $f_c = 2,5$ GHz.....	46

Fig. 3.2	Conjunto de treinamento para a rede MLP.....	46
Fig. 3.3	Evolução do aprendizado da rede MLP.....	47
Fig. 3.4	Conjunto de teste e validação da rede MLP (projeto).....	47
Fig. 3.5	Conjunto de treinamento para a rede MLP.....	48
Fig. 3.6	Conjunto de teste e validação da rede MLP (análise).....	48
Fig. 3.7	Modelos de filtros distribuídos através de redes MLPs.....	49
Fig. 3.8	Construção da antena isolada.....	49
Fig. 3.9	Antena isolada em um meio dielétrico dissipativo.....	50
Fig. 3.10	Conjunto de treinamento para a rede MLP.....	51
Fig. 3.11	Processo de aprendizado da rede MLP.....	52
Fig. 3.12	Saída da rede para o conjunto de teste R_o	52
Fig. 3.13	Saída da rede para o conjunto de teste X_o	53
Fig. 3.14	Fita metálica.....	54
Fig. 3.15	Resistência AC de uma fita metálica. (-) Medidas, (+) Eq. (3.3).....	55
Fig. 3.16	Resistência AC de uma fita metálica. (-) Medidas, (+) Modelo MLP.....	55
Fig. 3.17	Indutores espirais planares (a) Retangular. (b) Circular.....	56
Fig. 3.18	Representação de 10% do conjunto de treinamento empregado no caso do indutor retangular. O gráfico reúne (lado a lado, com $esp=1, 2, \dots, 20$) os valores da indutância total.....	57
Fig. 3.19	Convergência da rede MLP usada para o indutor espiral retangular.....	57
Fig. 3.20	Teste da modelo MLP para o indutor retangular: (—) Rede MLP, (---) Modelo EM, (3.4).....	58
Fig. 3.21	Teste do modelo MLP para o indutor circular: (o) Rede MLP, (—) Modelo EM, (3.5).	58
Fig. 3.22	Transistor de efeito de campo metal-semicondutor (MESFET).....	59
Fig. 3.23	Configuração do modelo neural para o FET.....	60
Fig. 3.24	Teste das redes MLP e RBF.....	61
Fig. 3.25	Convergência das redes MLP e RBF.....	61

Fig. 3.26	Resultados para a microfita invertida $\epsilon_{r1} = \epsilon_{r3} = 1$, (SiO ₂ , $\epsilon_{r2} = 3.78$) e (GaAs, $\epsilon_{r2} = 12.9$).....	62
Fig. 3.27	Substrato PBG como filtro em linha de microfita.....	63
Fig. 3.28	Aproximação da resposta de uma estrutura PBG através de modelos RFS.....	64
Fig. 3.29	Coefficiente de reflexão de uma carga padrão medida através da técnica TRL. Comparação entre os modelos: MLP (Jargon e Gupta, 2001) e RFS. (a) Modelo RFS com resolução baixa. (b) Modelo RFS com resolução alta.....	66
Fig. 3.30	Resultados dos modelos MLP e RFS (com três neurônios de resolução alta).....	68
Fig. 3.31	Magnitude de S_{11} para a terminação em curto. Comparação entre as medidas através da técnica TRL e os modelos: MLP com 5 neurônios ocultos, Jargon e Gupta (2001); RFS com 50 neurônios ocultos. (a) RFS com resolução baixa, $\sigma_0 = 10$. (b) RFS com resolução alta, $\sigma_0 = 200$	69
Fig. 3.32	Magnitude de S_{11} para a terminação em curto. Comparação entre as medidas (através da técnica TRL) e os modelos: MLP com 5 neurônios ocultos, Jargon e Gupta (2001); RFS com 50 neurônios ocultos. RFS com resolução baixa, 5 neurônios ocultos e $\sigma_0 = 2$. (b) RFS com resolução média, $\sigma_0 = 20$. (c) RFS com resolução alta, 50 neurônios ocultos e $\sigma_0 = 200$	70
Fig. 4.1	Ponto de operação DC do circuito para polarização de um diodo exponencial ($\eta_d = 1,2$; $I_{SAT} = 1\mu A$; $T_j = 22^\circ C$) através do método de Newton-Raphson.....	74
Fig. 4.2	Circuitos equivalentes Thevenin e Norton para capacitores e indutores resultantes da integração numérica via a regra dos trapézios.....	76
Fig. 4.3	(a) Circuito RLC série. (b) Circuito equivalente Thevenin para análise transitória. (c) Resposta ao degrau do circuito RLC com condições iniciais nulas, $\Delta t = 1,0 \mu s$	77
Fig. 4.4	(a) Circuito <i>beat generation</i> . (b) Equivalente Thevenin para análise transitória. (c) Resposta transitória para $x_4(t)$, dados: condições iniciais nulas, exceto $x_4(0) = 1,0 V$, $L_1 = L_2 = 0,01 H$, $C_1 = 0,2 F$, $C_2 = C_3 = 0,01 F$, $R = 0,025 \Omega$ e $\Delta t = 0,5 \mu s$	79
Fig. 4.5	(a) Circuito limitador a diodo. (b) Resposta transitória.....	81
Fig. 4.6	(a) Circuito chave eletrônica com um BJT. (b) Inclusão do modelo de Ebers-Moll. (c) Circuito equivalente linearizado usado na iteração NR.....	84
Fig. 4.7	Resposta no tempo para o circuito chave eletrônica com um BJT.....	84
Fig. 4.8	(a) Porta lógica com GaAs MESFET sem o circuito de mudança de nível. (b) Circuito equivalente para o GaAs MESFET. (c) Circuito não linear resultante. (d) Circuito equivalente linearizado para análise transitória.....	86

Fig. 4.9	Características $I-V$ para um GaAs MESFET. Modelos – Schichman-Hodges: $V_{Th} = 2,5V$, $\beta_{FET} = 71\mu A/V^2$, $\lambda_{FET} = 0$; Curtice: $V_{Th} = 2,5V$, $\beta_{FET} = 65\mu A/V^2$, $\lambda_{FET} = 0$; programa 2-D: comprimento da porta = $1.0\mu m$, concentração de doadores = $3 \times 10^{16} cm^{-3}$, espessura da camada ativa = $0.25\mu m$, $V_{BI} = 0,5V$. Generalização da rede MLP (MESFET com largura = $10\mu m$).	90
Fig. 4.10	Convergência dos algoritmos de treinamento.....	91
Fig. 4.11	Computações da condutância de dreno.....	92
Fig. 4.12	Transcondutância do MESFET: estimação e aproximação através de dois modelos MLP.....	93
Fig. 4.13	Comparação entre os resultados de simulação do circuito porta lógica com tensão de entrada pulsada para três modelos do GaAs MESFET. Os valores usados são os mesmos dados na Tab. 4.1 e na legenda da Fig. 4.10. (a) Tensões de saída. (b) Correntes de dreno. (c) Características $I-V$ do MESFET driver e trajetórias dinâmicas de tensão-corrente.....	95
Fig. 4.14	Comparação entre os tempos de ocupação da CPU (simulação em Matlab™ (1998) sob um processador AMD-500).....	95
Fig. 4.15	Circuito equivalente para a porta lógica, considerando-se o circuito equivalente completo para os MESFETs.....	96
Fig. 4.16	Respostas pulsadas do circuito porta lógica para diferentes larguras do MESFET funcionado como carga não linear. (—) Respostas usando-se o modelo MLP para os transistores GaAs MESFETs. (- - -) Respostas usando-se um modelo MLP para o circuito porta lógica.....	97
Fig. 5.1	(a) Rede arbitrária. (b) A mesma rede separada em duas sub-redes, cada uma aumentada com fontes apropriadas.....	101
Fig. 5.2	(a) Circuito retificador $\frac{1}{2}$ onda com filtro RC. (b) Inclusão do modelo para o diodo e destaque para a sub-rede não linear. (c) Sub-rede linear aumentada com a fonte $i_{SF}(t)$. Dados: $R_S = 5\Omega$, $R_L = 1K\Omega$, $C = 1000\mu F$, $C_D = 1\mu F$, $I_{SAT} = 1\mu A$, $T = 290K$	104
Fig. 5.3	(a) Aproximação através do MBH para a corrente no diodo. (b) Tensão na carga.....	106
Fig. 5.4	(a) Chave eletrônica a transistor. (b) Inclusão do modelo de Ebers-Moll. (c) Sub-rede linear aumentada. Dados: $R_B = 1K\Omega$, $R_C = 10K\Omega$, $\alpha_F = 0,99$ e $\alpha_R = 0,5$, $V_{cc} = 10V$	107
Fig. 5.5	Resposta em regime do circuito chave eletrônica a transistor para a entrada $v_S(t) = 2 \text{sen}(2\pi \times 10^3 t)$. (a) Aproximação para a corrente de emissor. (b) Tensões no transistor.....	109
Fig. 5.6	Amplificador classe C com entrada $J(t) = 0,1 \text{sen}(2\pi \times 10^8 t)$	109

Fig. 5.7	Resposta em regime para o amplificador classe C. (a) Tensões no transistor. (b) Corrente de emissor. (c) Componentes harmônicas da solução em regime.....	110
Fig. 6.1	Aproximação convencional de um modelo através de rede neural.....	113
Fig. 6.2	(a) Ilustração do método EM-ANN. (b) Modelo híbrido EM-ANN.....	113
Fig. 6.3	(a) Ilustração do método PKI. (b) Modelo PKI.....	114
Fig. 6.4	(a) Ilustração do método KBNN. (b) Modelo KBNN.....	115
Fig. 6.5	Ilustração da técnica SM.....	116
Fig. 6.6	(a) Ilustração do método SMN. (b) Modelo SMN.....	117
Fig. 6.7	(a) Ilustração do método FDSMN. (b) Modelo FDSMN.....	119
Fig. 6.8	(a) Ilustração do método FSMN. (b) Modelo FSMN.....	120
Fig. 6.9	(a) Ilustração do método FMN. (b) Modelo FMN.....	121
Fig. 6.10	(a) Ilustração do método FPSMN. (b) Modelo FPSMN.....	122
Fig. 6.11	(a) Ilustração do método FDPSMN. (b) Modelo FDPSMN.....	123
Fig. 6.12	Rede neural MLP com três camadas para mapeamento de espaços.....	125
Fig. 6.13	(a) Junção-T em linha de microfita. (b) Circuito equivalente.....	129
Fig. 6.14	Respostas típicas dos modelos preciso (Sonnet's <i>em</i> TM) e empírico, Gupta et al (1981), antes do uso dos métodos SM-ANN.....	131
Fig. 6.15	Respostas: (—) modelos SMN; (+) modelo preciso.....	133
Fig. 6.16	Respostas: (—) modelos FSMN; (+) modelo preciso.....	134
Fig. 6.17	Respostas: (—) modelos FPSMN; (+) modelo preciso.....	135
Fig. 6.18	Respostas: (—) modelos FDPSMN; (+) modelo preciso.....	136
Fig. 6.19	Evolução dos erros nos métodos SM baseados em redes MLPs.....	137
Fig. A.1	(a) Função de Rosenbrock, (*) curva $x_2 = x_1^2$. (b) Contornos de nível, direções do vetor gradiente, (+) lugar do mínimo: $x = (1,1)$	165
Fig. A.2	Mínimização da função de Rosenbrock; trajetórias a partir dos pontos $(-1, 0)$ e $(-1, 2)$. MÉTODOS – (a) Gradiente (1000 iterações), $\eta_G = 0,002$. (b) Gradiente conjugado (70 iterações) usando até 100 pesquisas de Fibonacci por iteração. (c) Fletcher-Powell (22 e 17 iterações, respectivamente). (d) Newton (7 iterações)...	166

LISTA DE SÍMBOLOS

9

Símbolo genérico, usado apenas nesta lista.

Letras Romanas

\vec{A}	vetor potencial magnético ($V \cdot m^{-1} \cdot rad \cdot s^{-1}$).
a_m, b_m	coeficientes da série de Fourier.
C_D	capacitância da junção PN de um diodo (F).
C_{DS}, C_{GD}, C_{GS}	capacitância entre eletrodos de um FET (F).
D	coeficiente de difusão ($m^2 \cdot s^{-1}$)
$d_k(n)$	resposta desejada ou sinal de referência (V).
E	função erro (ou custo).
\vec{E}	vetor campo elétrico ($V \cdot m^{-1}$).
$e_k(n)$	erro instantâneo.
esp	número de espiras.
$f(x)$	função arbitrária.
f_c	freqüência de corte (Hz).
f_o	freqüência de operação (Hz).
g_m	transcondutância (S).
G_{DS}	condutância de dreno (S).
G_{eq}	condutância equivalente (S).
h_1, h_2	comprimentos dos elementos da antena isolada (m).
h_s	altura do substrato de uma linha de microfita (m).
\vec{H}	vetor campo magnético ($V \cdot m^{-1}$)
H	matriz Hessiana.
$H(\omega)$	função de transferência (transmissão).
I	matriz identidade.
i_B, i_C, i_E	formas de onda de corrente no BJT (A).
i_D	corrente total através de um diodo de junção PN (A).
I_{DS}	corrente dreno-fonte de um FET (A).
I_{eq}	corrente equivalente (A).
i_F, i_R	correntes exponenciais no modelo de Ebers-Moll (A).
I_{SAT}	parâmetro corrente de saturação no modelo do diodo exponencial (A).
i_{SF}	fonte de corrente na forma de uma série de Fourier (A).
j	número imaginário puro.

J	matriz Jacobiana.
\bar{J}	densidade de corrente: superficial (A/m^2) ou volumétrica (A/m^3).
k_o	número de onda ($rad \cdot m^{-1}$).
k_B	constante de Boltzmann ($1,38 \times 10^{-23} J \cdot K^{-1}$).
L	Comprimento (m).
SSE	erro quadrático somado.
n	número de iterações (ou épocas).
$n(x,y)$	distribuição de portadores
N_D	concentração de impurezas doadoras (m^{-3}).
$net_k(n)$	potencial de ativação do k -ésimo neurônio de uma rede neural.
N_s	número de seções do filtro distribuído em linha de microfita.
P_s	vetor de pesquisa.
q	carga elementar ($1,60 \times 10^{-19} C^{-1}$).
\bar{R}	vetor posição (m).
R_{ac}	resistência elétrica de um condutor devida ao efeito pelicular (Ω).
R_{dc}	resistência elétrica à corrente contínua de um condutor (Ω).
R_{eq}	resistência elétrica equivalente (Ω).
\bar{r}_o	vetor posição da origem até um ponto de campo (m).
s	variável complexa $s = \alpha + j\omega$.
t	tempo (s).
T	período fundamental de um sinal periódico (s).
t_E	espessura de uma fita metálica (m).
T_j	temperatura de uma junção PN (K).
t_h	vetor de centros da h -ésima função de ativação de uma ANN
v	vetor velocidade de deriva ($m \cdot s^{-1}$).
$v(n)$	vetor de pesos de uma ANN.
v_{be}, v_{bc}, v_{ce}	formas de onda de tensão de um BJT (V).
V_{DS}, V_{GS}, V_{DG}	tensões DC de um FET (V).
V_T	tensão térmica de um diodo de junção PN. $V_T = k_B T_j / q$ (V).
w	largura (m)
$w(n)$	vetor de pesos de uma ANN.
$x_k(n)$	vetor de entradas do k -ésimo neurônio de uma rede neural (V).
x	vetor de variáveis independentes (de estado).
X_o	reatância de entrada de uma antena isolada (Ω)
$y_k(n)$	saída do k -ésimo neurônio de uma rede neural.
$z(n)$	vetor de saídas de uma ANN.
$Z(\omega)$	função de transferência (impedância, Ω).
Z_o	impedância característica (Ω).

Letras Gregas

α_D	parâmetro do diodo exponencial ($\alpha_D = k_B T_j / \eta_D q$)
α_F	ganho de corrente direta no modelo de Ebers-Moll de um BJT.
α_R	ganho de corrente inversa no modelo de Ebers-Moll de um BJT.
δ_h	gradiente local na camada oculta de uma rede MLP.
δ_q	gradiente local na camada de saída de uma rede MLP.
ε	permissividade elétrica
ε_r	permissividade elétrica relativa ($F \cdot m^{-1}$).
η	taxa de aprendizado.
η_o	impedância no espaço livre $\eta_o = 120\pi$ (Ω).
λ_o	comprimento de onda no espaço livre (m).
μ	permeabilidade magnética ($H \cdot m^{-1}$).
θ_k	potencial de polarização do k -ésimo neurônio de uma rede neural.
π	constante (3,1415926...).
σ	condutividade elétrica
σ_h	largura da h -ésima função de ativação de uma ANN.
$\varphi(\cdot)$	função de ativação dos neurônios de uma rede neural.
ψ	potencial elétrico escalar (V).
ω	freqüência angular ($rad \cdot s^{-1}$)

Operadores

\mathfrak{Q}^T	transposta de uma matriz ou vetor.
$\hat{\mathfrak{Q}}$	vetor unitário ($\hat{\mathfrak{Q}} = \bar{\mathfrak{Q}}/ \bar{\mathfrak{Q}} $).
$\bar{\mathfrak{Q}}$	vetor localizado num sistema de coordenadas.
$\tilde{\mathfrak{Q}}$	valor estimado.
$ \mathfrak{Q} $	módulo (escalar), $ \mathfrak{Q} = \sqrt{\mathfrak{Q}^2}$.
$\ \mathfrak{Q}\ $	norma Euclidiana, $\ \mathfrak{Q}\ = \sqrt{\mathfrak{Q}^T \cdot \mathfrak{Q}}$.
\mathfrak{Q}^*	conjugado de uma variável complexa.
$\angle \mathfrak{Q}$	fase de uma variável complexa.
$\nabla(\mathfrak{Q})$	gradiente de uma função escalar.
$\nabla \cdot (\mathfrak{Q})$	divergente de um vetor.
$\nabla \times (\mathfrak{Q})$	rotacional de um vetor.
$\partial/\partial \mathfrak{Q}$	derivada parcial.
$\dot{\mathfrak{Q}}$	derivação com respeito ao argumento.
$\mathfrak{Q} * \mathfrak{Q}$	convolução.

LISTA DE SIGLAS

ANN	Artificial Neural Network
BJT	Bipolar Junction Transistor
BP	Backpropagation
CAD	Computer Aided Design
CPW	Coplanar Waveguide
DFT	Discrete Fourier Transform
EDO	Equação Diferencial Ordinária
EM	Electromagnetic
EM-ANN	Electromagnetic-Artificial Neural Network
FDSMN	Frequency Dependent Space Mapping Neuromodeling
FDPSMN	Frequency Dependent Partial Space Mapping Neuromodeling
FDTD	Frequency Difference Time Domain
FET	Field Effect Transistor
FPSMN	Frequency and Partial Space Mapping Neuromodeling
FSMN	Frequency Space Mapping Neuromodeling
KBNN	Knowledge Based Neural Network
LRM	Line-Reflect-Match
LTI	Linear Time Invariant
MBH	Método do Balanço Harmônico
MESFET	Metal-Semiconductor Field Effect Transistor
MIC	Microwave Integrated Circuit
MLP	Multilayer Perceptrons
MMIC	Monolithic Microwave Integrated Circuit
MOSFET	Metal-Oxide-Semiconductor Field Effect Transistor
OSLT	Open-Short-Load-Thru
PBG	Photonic Bandgap
PBGS	Photonic Bandgap Substrate
PKI	Priori Knowledge Input
RBF	Radial Basis Function
RF	Radio Frequency
RFS	Rede de Funções Sample
Rpop	Resilient Backpropagation
SMN	Space Mapping Neuromodeling
SEMS	Smart Electromagnetic Structure
SPICE	Simulation Program with Integrated Circuit Emphasis
STC	Search-Then-Converge
TRL	Thru-Reflect-Line
VNA	Vector Network Analyzer

RESUMO

Esta tese contribui para o desenvolvimento de metodologias através de redes neurais para a modelagem de dispositivos e circuitos de RF/microondas. As aplicações desta técnica neuro-computacional alternativa foram revistas.

O principal objetivo foi o estudo das redes neurais sem realimentação. As estratégias de aprendizado supervisionado, baseadas no método do gradiente, foram formuladas para estas redes neurais. Uma nova rede neural, designada RFS, foi proposto com o emprego de funções de ativação *sample*.

O método convencional, EM-ANN, foi aplicado em diversos problemas de modelagem em RF/microondas e na simulação de circuitos não lineares.

Na análise transitória de circuitos não lineares, aplicações inéditas de modelos MLPs para transistores MESFET de GaAs em simuladores de circuitos, demonstram a viabilidade do uso de redes neurais, como modelos de dispositivos ativos, para a análise de circuitos no domínio do tempo.

Na análise em regime permanente senoidal de circuitos não lineares, aplicou-se as redes de Fourier em conjunto com um método inédito, denominado balanço harmônico simplificado.

Modelos neurais baseados no conhecimento empírico foram desenvolvidos. Uma nova técnica de mapeamento de espaço baseado em redes neurais, denominada FDPSMN, foi proposta.

Todos os algoritmos formulados, bem como, os métodos numéricos empregados nesta tese foram implementados em Matlab™ e C++.

Sugestões para trabalhos futuros foram apresentadas.

ABSTRACT

This thesis contributes to the development of neural network methodologies for modeling of RF/microwave devices and circuits. The applications of this alternative neurocomputational technique are reviewed.

The study of the feedforward neural networks was the main objective. The supervised learning strategies, using the steepest descent method, were formulated for these neural networks. A novel neural network, nominated RFS, was proposed with the use of the *sample* activation function.

The conventional method, EM-ANN, was applied in many RF/microwave modeling problems and in the nonlinear circuit simulation.

In the nonlinear circuit transient analysis, unpublished application examples of MLP models for GaAs MESFET transistors in circuit simulators, demonstrate the viability of the use of neural networks, as active device models, for time domain circuit analysis.

In the nonlinear circuit steady-state analysis, was applied the Fourier network in conjunction with unpublished method, nominated simplified harmonic balance.

Empirical based neural models were developed. A new neural network based space mapping technique, nominated FDPSMN, was proposed.

All formulated algorithms, as well as, the numericals methods used in this subject were implemented in Matlab™ and C++.

Suggestions for further research were presented.

Capítulo 1

INTRODUÇÃO

Com o desenvolvimento dos mercados comerciais de produtos para comunicação sem fio, em ritmo acelerado na última década, aumentou bastante o interesse na melhoria das aproximações, para projetos de circuitos, em frequências de rádio (RF) e na faixa de microondas. Esta indústria, voltada para um mercado dinâmico, com ênfase em baixos custos, demanda, cada vez mais, ferramentas de CAD para dispositivos e circuitos de RF/microondas.

O desenvolvimento de técnicas de simulação eletromagnética (EM) nestas áreas, direcionou a criação de ferramentas de CAD para circuitos híbridos RF/microondas e para circuitos integrados monolíticos de RF/microondas (MMIC). Com isto, uma certa maturidade foi atingida no desenvolvimento destes projetos de circuitos.

O uso de modelos através de redes neurais artificiais (ANNs) é um dos progressos recentes nas áreas de RF e microondas. Treinadas a partir de resultados EMs, as ANNs são bastante apropriadas para o desenvolvimento de CAD nestas áreas, Watson e Gupta (1996), Creech et al (1997) e Veluswami et al (1997). Os modelos através das ANNs mantêm a precisão dos simuladores EMs. Além disso, sua eficiência computacional se equipara à maioria dos modelos empíricos/analíticos.

As redes neurais surgiram como uma técnica poderosa para aproximar relações de entrada/saída arbitrárias, sendo aplicadas à solução de problemas em diversas áreas do conhecimento: telecomunicações, Cooper (1994); processamento de sinais, Rioul e Vetterli (1991); sensoriamento remoto, Goita (1994); controle, Balakrishnan e Weil (1996); reconhecimento de padrões, Smetanin (1995); fabricação, Nunmaker e Sprague (1996); biomédica, Alvager et al (1994); entre outras.

Nos últimos anos, verificou-se o emprego cada vez maior das ANNs na área de projeto em RF/microondas, Litovski et al (1992), Horng (1993), Vai e Prasad (1993), Zaabab et al (1994), Zhang e Creech (1999). Aplicações de modelagem através de redes neurais reportadas na literatura, incluem: dispositivos ativos, Litovski et al (1992), Günes et al (1996), Shirakawa et al (1997); dispositivos passivos, Watson e Gupta (1996), Creech et al (1997), Veluswami et al (1997); linhas de microfita, Patnaik e Mishra (1997); antena *patch*, Mishra e Patnaik (1997); conexões eletro-ópticas, Zhang et al (1997b); estruturas eletromagnéticas espertas (SEMS), Thursby et al (1995); antenas de abertura, Washington (1997);

Além disso, destacam-se os trabalhos pioneiros nas aplicações para: estruturas de RF e microondas, Zhang et al (1999), Wang (1999), Zhang e Gupta (2000); redes baseadas no conhecimento, Rayas-Sanchez (2001), Gupta (1996), Watson et al (1998), Wang e Zhang (1997a) e (1997b); algoritmos de treinamento, Wang (1999), Devabhakruni et al (1999).

Normalmente, entre os modelos baseados em regressão polinomial e os modelos com base em ANNs, estes últimos são mais precisos, Smith e Mason (1997), Tsintikidis et al (1997). Se comparadas a modelos baseados em tabelas de consulta, as ANNs possibilitam modelos com um número maior de dimensões, Zhang et al (1997a). Além disso, para um mesmo modelo neural várias saídas são possíveis.

Os modelos através de ANNs são desenvolvidos a partir de exemplos de treinamento (simulação EM ou medidas). Desta forma, são capazes de “aprender” os mapeamentos arbitrários de entrada/saída para os quais são treinadas. Diversas características interessantes são apresentadas pelas redes neurais, Bishop (1996):

i. Nenhum conhecimento sobre o mapeamento é necessário para o desenvolvimento de uma ANN. As relações são inferidas através dos exemplos de treinamento e, portanto, a rede neural não precisa ser definida explicitamente.

ii. As redes neurais podem generalizar, o que significa que elas são capazes de responder a exemplos novos, dentro da região de interesse, definida na fase de treinamento.

iii. As redes neurais são capazes de aproximar qualquer mapeamento contínuo não linear, como tem sido demonstrado por Hornik et al (1990).

A técnica através de redes neurais constituem uma metodologia geral, que, aplicadas como modelos para dispositivos e circuitos de RF/microondas, são precisas e numericamente eficientes. Após o treinamento com resultados eletromagnéticos, os modelos EM-ANN podem ser integrados em simuladores de circuitos. Além disso, servem como ferramentas alternativas de CAD nas áreas de RF e microondas.

Nesta tese, aborda-se o emprego da tecnologia de redes neurais para dispositivos e circuitos de RF/microondas. Diversas metodologias existentes nestas áreas são exploradas e algumas contribuições são apresentadas. As redes neurais sem realimentação com algoritmos de treinamento supervisionados – usando a estratégia de aprendizado através da correção do erro, são analisadas. Especificamente, estuda-se o uso destas técnicas neuro-computacionais alternativas em duas áreas distintas da engenharia de microondas: modelagem de dispositivos e simulação de circuitos não lineares.

No Capítulo 2, as redes neurais sem realimentação são abordadas e seus algoritmos de treinamento são descritos de forma concisa. Para a rede MLP, além do algoritmo BP, outros são implementados para acelerar o seu treinamento. As demais redes neurais são treinadas diretamente através do método do gradiente. Uma nova função de ativação (*sample*) é proposta para os neurônios ocultos de uma ANN sem realimentação. Esta rede neural é designada rede de funções *sample* (RFS). Para a verificação do desempenho dos algoritmos de treinamento implementados, as redes neurais são aplicadas à tarefa de aproximar funções.

No Capítulo 3, descreve-se várias aplicações do método convencional EM-ANN para dispositivos de RF/microondas. Os métodos eletromagnéticos (usados na geração dos conjuntos de treinamento) e os modelos ANNs são descritos. A capacidade de generalização das ANNs baseadas em resultados EMs é verificada.

No Capítulo 4, descreve-se as técnicas numéricas utilizadas para a análise transitória de circuitos não lineares. Neste caso, para a análise transitória de uma porta lógica, são usados modelos EM-ANN, com base nas redes neurais MLPs, para transistores MESFETs de GaAs. Através dos resultados obtidos na simulação de circuitos no domínio do tempo, demonstra-se que estes modelos são precisos e eficientes.

No Capítulo 5, é apresentada uma variação simplificada do MBH para a análise em regime permanente senoidal de circuitos não lineares. Este método é desenvolvido do ponto de vista das redes de Fourier. Alguns exemplos numéricos são apresentados.

No Capítulo 6, descreve-se diversos métodos de mapeamento de espaço através de redes neurais (SM-ANN), alternativos ao método convencional EM-ANN. As estratégias empregadas com base no conhecimento empírico em projetos de RF/microondas, são resumidas. Para a aplicação destes métodos, algumas modificações necessárias à rede MLP,

são descritas. Uma nova técnica SM-ANN, denominada FDPSMN, é proposta. Resultados para a modelagem de junções-T em microfita, são apresentados.

No Capítulo 7, concluímos esta tese, indicando algumas sugestões para trabalhos futuros.

As contribuições originais do autor, apresentadas nesta tese, são:

- (1) A formulação e implementação, em MatlabTM e C++, de algoritmos, baseados no método do gradiente, para o treinamento supervisionado de redes neurais sem realimentação.
- (2) A formulação e desenvolvimento da rede RFS. Um tipo novo rede neural sem realimentação, que utiliza uma função de ativação do tipo *sample*. é proposta para problemas de modelagem onde outras ANNs sem realimentação falham.
- (3) A modelagem precisa de respostas de dispositivos de RF/microondas (através de parâmetros de espalhamento medidos), empregando a RFS treinada através do método convencional (EM-ANN).
- (4) A formulação e desenvolvimento de redes MLPs como modelos de transistores MESFETs de GaAs, voltadas para análise transitória de circuitos não lineares.
- (5) A formulação e desenvolvimento do método do balanço harmônico simplificado do ponto de vista das redes de Fourier.
- (6) A formulação e desenvolvimento de uma nova técnica de mapeamento de espaço através de redes neurais, denominada FDPSMN.

Capítulo 2

REDES NEURAIS SEM REALIMENTAÇÃO

2.1 INTRODUÇÃO

A rede neural artificial (ANN) é uma máquina, implementada com componentes eletrônicos, cujo projeto tenta imitar a maneira como o cérebro realiza uma tarefa particular ou função de interesse. Por outro lado, o desenvolvimento de uma rede neural pode ser simulado em computadores digitais. Dentre as diversas definições existentes, cita-se Haykin (1999):

–“A rede neural é um processador paralelo distribuído, com uma vocação natural para armazenar o conhecimento experimental, tornando-o disponível para uso posterior. Seu funcionamento é parecido com o do cérebro em dois aspectos: i) o conhecimento é adquirido pela rede por um processo de aprendizado; ii) as amplitudes das conexões entre os neurônios artificiais, conhecidas como pesos sinápticos, são usadas para armazenar o conhecimento”.

Existem diferentes arquiteturas e estratégias de aprendizado (algoritmos treinamento) para o desenvolvimento de uma ANN. Nesta tese, a abordagem é limitada às redes sem realimentação, cujos algoritmos de treinamento são supervisionados, usando a estratégia de aprendizado pela correção do erro.

Em particular, neste capítulo descreve-se de forma concisa alguns paradigmas de redes neurais com propagação direta (sem realimentação), e, seus respectivos algoritmos de treinamento. São analisadas as seguintes redes neurais sem realimentação:

- Rede *Perceptrons* de Múltiplas Camadas (MLP)
- Rede de Funções de Base Radiais (RBF)
- Rede *Wavelet*
- Rede de Fourier
- Rede de Funções *Sample* (RFS)

Dentre estas redes neurais, o MLP, treinado com o algoritmo da *backpropagation*, tornou-se o paradigma mais popular na área de RF/microondas. Contudo, diversas limitações deste modelo têm levado à criação de métodos alternativos de treinamento para o MLP. Alguns dos algoritmos de treinamento estudados são descritos.

Os algoritmos de treinamento para as outras redes com propagação direta são derivados através do método do gradiente. A realização desta tarefa é crucial para formulação e implementação dos modelos através de ANNs. Todos os algoritmos, descritos neste capítulo, foram implementados através de programas elaborados em Matlab™ (1998) e C++.

Uma contribuição na área foi o desenvolvimento de um tipo novo de rede sem realimentação, denominada rede de funções *sample*. Enfim, as redes neurais sem realimentação são analisadas do ponto de vista da teoria de aproximação. Algumas simulações destes paradigmas de redes neurais, aplicadas à aproximação de funções, são apresentadas.

2.1.1 Resumo do Método Utilizado

Neste estudo, analisa-se o emprego de redes neurais sem realimentação para a modelagem de dispositivos e circuitos de RF/microondas. A Fig. 2.1(a) ilustra o método convencional (EM-ANN) para o desenvolvimento de um modelo neural. Na fase de aprendizado supervisionado, dados obtidos por meio de simulação EM ou medidas, assumem a figura de um “professor”, fornecendo os exemplos $\{\mathbf{x}(n); \mathbf{d}(n)\}$, para a rede neural, Fig. 2.1(a). A rede neural é treinada a fim de minimizar-se um erro médio quadrático, $E(n)$,

$$E(n) = \frac{1}{2Q} \sum_{q=1}^Q e_q^2(n) \quad (2.1)$$

em que, Q é o número de neurônios de saída. O erro, na n -ésima época de treinamento, entre a q -ésima saída da rede, $z_q(n)$, e a correspondente resposta desejada, $d_q(n)$, é dado por:

$$e_q(n) = d_q(n) - z_q(n) \quad (2.2)$$

O método do gradiente é a estratégia mais usada na otimização dos parâmetros de uma ANN. Cada ajuste é efetuado na mesma direção, mas no sentido oposto ao vetor gradiente,

$$\Delta w(n) = -\eta \nabla E(n) \tag{2.3}$$

para o qual, o parâmetro $\eta > 0$ determina a amplitude do ajuste aplicado.

Após o treinamento e a validação através de teste, o modelo neural é capaz de generalizar para regiões fora do conjunto de treinamento, onde há pouco (ou nenhum) conhecimento sobre as respostas de um dispositivo. Tendo sido validada, a rede neural pode ser considerada como uma memória associativa, isto é: para entradas novas, estima as saídas que um dado dispositivo apresentaria, Fig. 2.1(b).

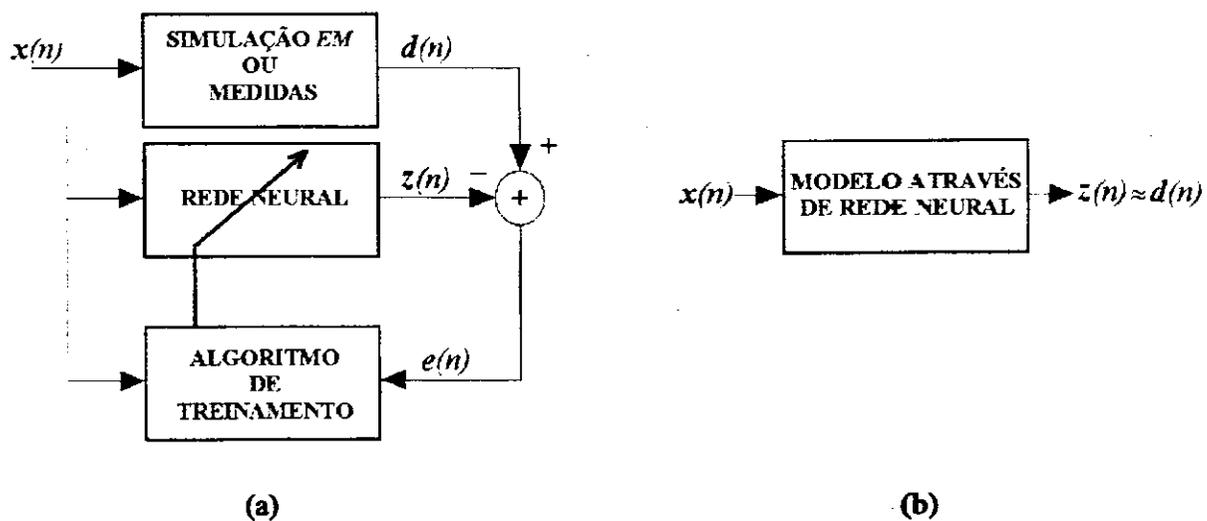


Fig. 2.1 (a) Método convencional (EM-ANN) para o desenvolvimento de modelos através de redes neurais. (b) Modelo através de rede neural.

2.1.2 Modelos de Neurônio Artificial

O neurônio artificial, doravante referido simplesmente como neurônio, é a unidade de processamento de informação de uma rede neural. No modelo de McCulloch-Pitts (1943), um neurônio, Fig. 2.2, é constituído por um conjunto de conexões sinápticas (pesos), um somador e uma função de ativação. As entradas do neurônio são ponderadas pelos respectivos

pesos e aplicadas ao somador. Da saída deste combinador linear, acrescido um valor de limiar (ou polarização), resulta o nível de atividade interna do neurônio. A resposta da função de ativação a este potencial interno corresponde à saída do neurônio. Em termos matemáticos, a operação do k -ésimo neurônio (com P entradas) de uma rede neural é expressa por:

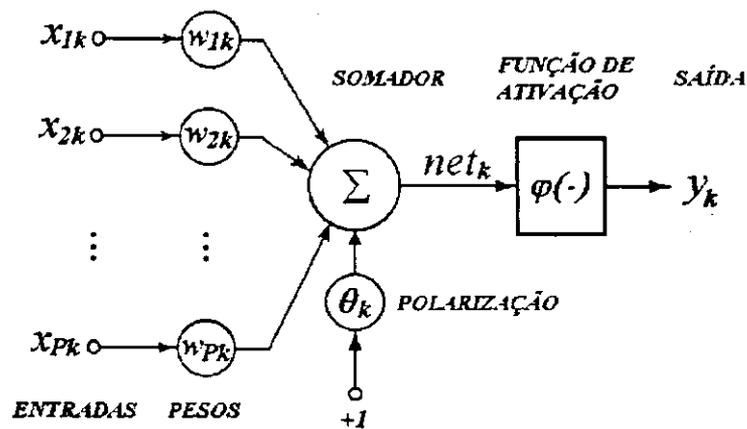


Fig. 2.2 Modelo do k -ésimo neurônio de uma rede neural sem realimentação.

$$net_k = \sum_{j=1}^P w_{kj} x_{jk} - \theta_k \quad (2.4)$$

$$y_k = \varphi(net_k) \quad (2.5)$$

em que, x_{jk} é entrada do neurônio k originada no neurônio j ; w_{kj} é o peso sináptico entre os neurônios k e j ; θ_k é um valor de polarização; net_k é o potencial de ativação interna. A função de ativação $\varphi(\cdot)$ define o modelo para um neurônio. Na formulação de um modelo de neurônio, várias funções podem ser empregadas, o que resulta em diferentes tipos de representação para um dado conjunto de entradas. Por exemplo, na seção 2.7, no desenvolvimento do modelo RFS os neurônios ocultos são definidos através de uma função de ativação do tipo *sample*.

2.2 REDE MLP

2.2.1 Configuração

Quando reunidas em camadas, as unidades de processamento elementares (neurônios) constituem as redes neurais com propagação direta, em camada simples, Fig. 2.3(a), ou em camadas múltiplas, Fig. 2.3(b). Cada neurônio de uma camada oculta está conectado a todos os neurônios das camadas adjacentes (anterior e posterior). Nesta seção, descreve-se as características de um tipo destas redes, denominada *Multilayer Perceptrons* (MLP). O algoritmo da *backpropagation* (BP) e outras estratégias de treinamento serão abordadas.

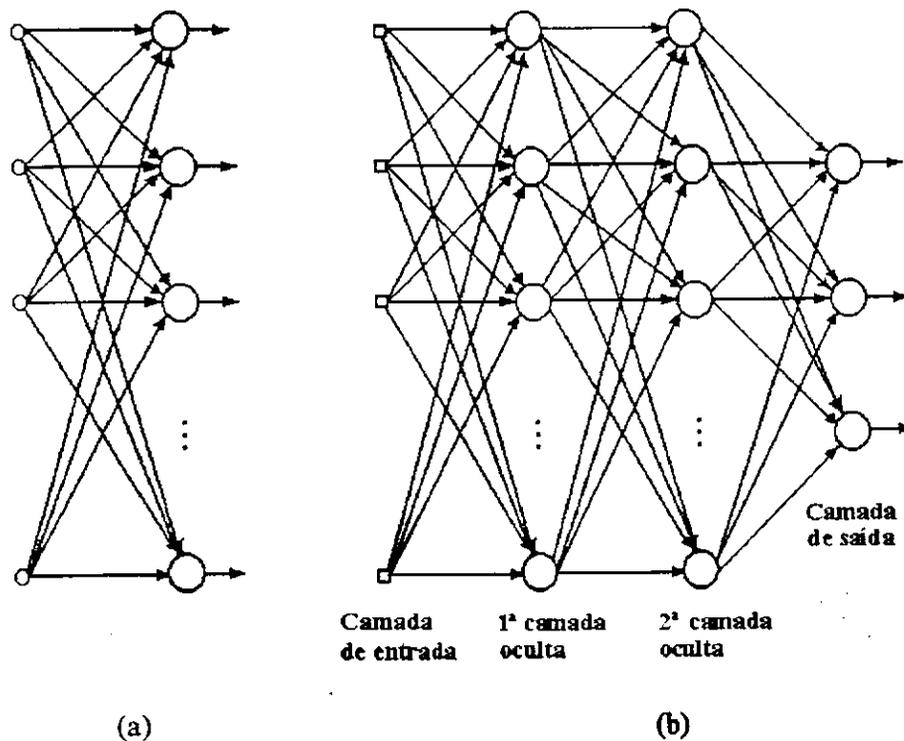


Fig. 2.3 Representação gráfica de redes sem realimentação:
(a) duas camadas; (b) quatro camadas.

2.2.2 Algoritmo da *Backpropagation*

A rede MLP treinada através do algoritmo BP, ou retropropagação do erro, é o tipo mais popular de rede neural. Descrita pela primeira vez por Werbos (1974), ficou praticamente desconhecida da comunidade científica por mais de uma década. Os estudos reportados em meados da década de oitenta por Parker (1985), Lecun (1985) e Rumelhart et al (1986), representaram um marco no desenvolvimento das ANNs.

Os conceitos básicos do algoritmo BP são facilmente compreendidos. Infelizmente estas idéias simples são obscurecidas por uma notação intrincada. Derivações formais deste algoritmo sempre são tediosas. Widrow e Lehr (1990), apresentaram um desenvolvimento informal. Na formulação do algoritmo BP, adota-se a configuração mais simples para a rede MLP (com uma camada oculta) e a notação apresentada na Fig. 2.4.

De um modo geral, os neurônios não lineares da rede MLP são definidos com uma função de ativação *sigmoidal* (em forma de 'S'). Em particular, utiliza-se a função logística (2.10). A propagação direta dos sinais através da rede MLP é dada através das expressões:

$$net_h = \sum_{p=0}^P w_{hp} x_p, \quad h = 1, 2, \dots, H \quad (2.6)$$

$$y_h = \varphi(net_h) \quad (2.7)$$

$$net_q = \sum_{h=0}^H v_{qh} y_h, \quad q = 1, 2, \dots, Q \quad (2.8)$$

$$z_q = \varphi(net_q) \quad (2.9)$$

$$\varphi(net) = \frac{1}{1 + \exp(-net)} \quad (2.10)$$

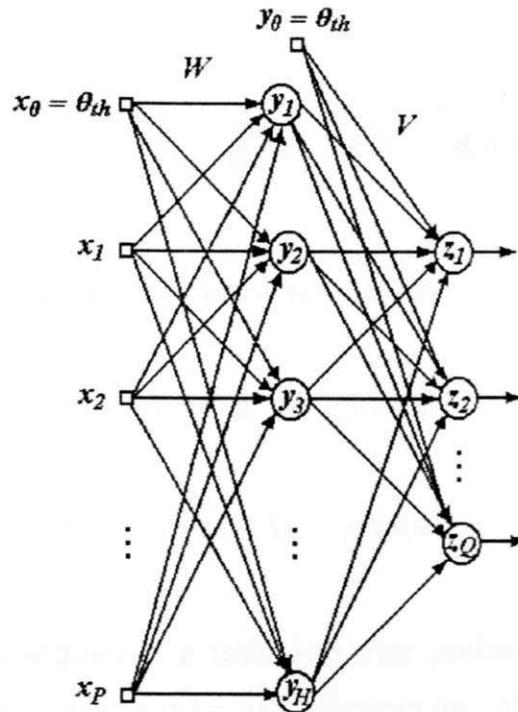


Fig. 2.4 Rede MLP com uma camada oculta.

para as quais, conforme a Fig. 2.4, tem-se: $x^T = [x_0, x_1, \dots, x_P]$, as entradas da rede ($x_0 = \theta_{th}$); $\theta_{th} = -1$, um valor de limiar; W e V , as matrizes de pesos da rede; net_h e net_o , os potenciais de ativação; $\varphi(\cdot)$, a função de ativação de cada neurônio; $y^T = [y_0, y_1, \dots, y_H]$, as saídas dos neurônios ocultos ($y_0 = \theta$); $z^T = [z_1, z_2, \dots, z_Q]$, o vetor de saídas da rede.

Na fase de treinamento, a cada iteração do algoritmo BP, o ajuste dos pesos é efetuado com a retropropagação dos sinais de erro, (2.2). O objetivo é minimizar a média do erro quadrático somado,

$$SSE = \frac{1}{N} \sum_{n=1}^N E(n) \tag{2.11}$$

em que, $E(n)$ foi definido em (2.1). O treinamento do MLP com o algoritmo BP é realizado com base no cálculo dos gradientes locais, Haykin (1999). Com o uso de (2.10), resulta as seguintes expressões para os gradientes locais nas camadas de saída e oculta:

$$\delta_q = -\frac{\partial E}{\partial e_q} \frac{\partial e_q}{\partial z} \frac{\partial z}{\partial net_q} = \varphi'(net_q) e_q = z_q(1-z_q)e_q, \quad q=1,2,\dots,Q \quad (2.12)$$

$$\delta_h = \varphi'(net_h) \sum_{q=1}^Q \delta_q v_{qh} = y_h(1-y_h) \sum_{q=1}^Q \delta_q v_{qh}, \quad h=0,2,\dots,H \quad (2.13)$$

respectivamente. Os valores dos ajustes aplicados aos pesos são dados por:

$$\Delta v_{qh} = \eta \delta_q y_h, \quad q=1,2,\dots,Q \text{ e } h=0,2,\dots,H \quad (2.14)$$

$$\Delta w_{hp} = \eta \delta_h x_p, \quad h=1,2,\dots,H \text{ e } p=0,2,\dots,P \quad (2.15)$$

Na formulação acima, para simplificar a formulação, o índice de iterações, n , foi omitido. A cada iteração, um exemplo, $s(n) = \{\mathbf{x}(n); \mathbf{d}(n)\}$, do conjunto de treinamento é apresentado à rede neural. Neste contexto, uma época corresponde à apresentação de todos os exemplos de um conjunto de treinamento. As expressões acima podem ser reformuladas para o caso mais geral, com mais camadas ocultas. Através do conceito de gradiente local, a inclusão de novas camadas ocultas torna-se um procedimento direto.

2.2.3 Métodos Alternativos Para o Treinamento do MLP

Esta seção aborda o conceito de aprendizado supervisionado para as redes neurais MLPs. Em algumas aplicações de modelos MLPs na área de RF/microondas, verifica-se a convergência lenta do algoritmo BP. Para a melhoria de sua eficiência computacional, são formulados alguns métodos alternativos para o treinamento de MLPs. Através de modificações no algoritmo BP, o processo de treinamento destas redes neurais é acelerado.

O algoritmo BP apresenta duas propriedades distintas: uma formulação simples para o cálculo local; um ajuste dos pesos a cada exemplo (ou por época) no sentido oposto ao gradiente de uma dada função erro. Estas duas propriedades do aprendizado das redes MLPs

são responsáveis por suas vantagens e desvantagens, Haykin (1999). Por sua vez, Jacobs (1988) identifica duas causas fundamentais para a convergência lenta do algoritmo BP:

1. Quando uma superfície de erro apresentar uma variação pequena com relação a um dado peso, sua derivada com relação a este peso terá uma magnitude pequena. Por conseguinte, o ajuste aplicado ao peso também será pequeno. Portanto, para produzir uma redução significativa da função erro, muitas iterações do algoritmo serão necessárias. Por outro lado, se a superfície de erro apresentar uma variação grande com relação a um dado peso, a respectiva derivada terá uma magnitude elevada, bem como, o ajuste aplicado ao peso. Isto pode acarretar numa ultrapassagem do mínimo da superfície de erro, Fig. 2.5.

2. O vetor oposto ao vetor gradiente pode apontar para longe do mínimo da superfície de erro; portanto, o algoritmo ajustará os pesos numa direção ruim.

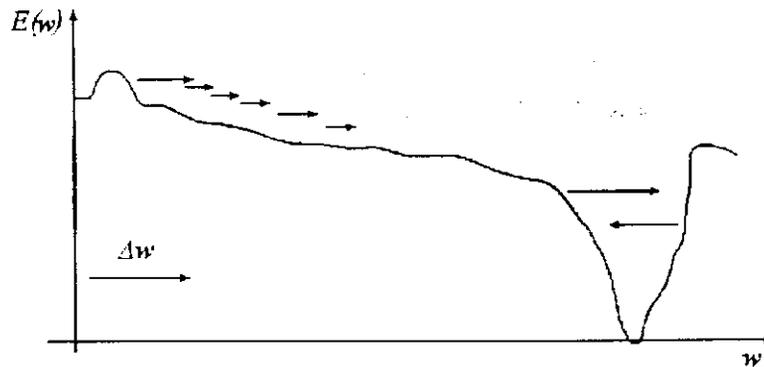


Fig. 2.5 Influência da escolha da taxa de aprendizado na convergência do algoritmo BP.

Uma vez conhecidas as derivadas parciais, a próxima etapa no algoritmo BP é o cálculo do ajuste dos pesos. Na sua forma mais elementar, este ajuste é um passo escalonado no sentido oposto ao gradiente da função erro, (2.1).

Apesar desta regra de aprendizado ser bastante simples, a escolha de uma taxa de aprendizado apropriada é uma tarefa difícil. Uma boa escolha depende da forma de (2.1), que muda conforme a tarefa de aprendizado. Outro problema do método do gradiente é a influência da derivada parcial nos valores dos ajustes calculados. Além disso, mesmo que sob certas circunstâncias a convergência para um mínimo (local) possa ser provada, não há garantias que o mínimo global da função custo seja encontrado, Riedmiller (1994). Uma idéia

inicial para tornar o aprendizado mais estável, reduzindo as oscilações dos pesos durante o treinamento da rede MLP, foi a inclusão do termo momento:

$$\Delta w(n) = -\eta \frac{\partial E}{\partial w}(n) + \alpha \Delta w(n-1) \quad (2.16)$$

A constante positiva α controla a influência do ajuste anterior sobre o ajuste atual dos pesos. A regra expressa por (2.16) denomina-se regra delta generalizada, Rumelhart et al (1986). Deve-se observar que, apesar de ser bem aplicada em muitas tarefas de aprendizado, esta não é uma técnica geral para ganhos de estabilidade e aceleração da convergência. É comum, no uso do método do gradiente com o termo momento, reduzir a taxa de aprendizado para evitar instabilidade no processo de aprendizado, Riedmiller (1994).

Uma forma importante de melhorar a convergência do algoritmo BP é otimizar a taxa de aprendizado, bem como, o termo momento, durante o treinamento do MLP. Uma regra empregada é o método do gradiente estocástico, Robins e Monro (1951):

$$\eta(n) = \kappa / n \quad (2.17)$$

em que κ é uma constante; n , o número de épocas. O esquema “*pesquisa-então-converge*” (STC), proposto por Darken et al (1992), sugere a seguinte adaptação:

$$\eta(n) = \eta_0 \frac{1 + \left(\frac{\kappa}{\eta_0}\right) + \left(\frac{n}{\tau}\right)}{1 + \left(\frac{\kappa}{\eta_0}\right) \cdot \left(\frac{n}{\tau}\right) + \tau \cdot \left(\frac{n}{\tau}\right)^2} \quad (2.18)$$

em que, η_0 , τ e κ são constantes positivas.

Um dos trabalhos mais interessantes nesta área é a regra delta-bar-delta, proposta por Jacobs (1988). Esta estratégia de treinamento foi desenvolvida baseada nas seguintes heurísticas:

- i. Cada parâmetro ajustável da rede, que contribui no cálculo da função erro, deve ter sua própria taxa de aprendizado.
- ii. Deve-se permitir que cada taxa de aprendizado possa variar durante o treinamento.
- iii. Quando a derivada da função erro, em relação a um determinado peso, apresenta o mesmo sinal algébrico entre duas iterações consecutivas do algoritmo, a taxa de aprendizado para aquele peso em particular deve ser aumentada.
- iv. Quando o sinal algébrico da derivada da função erro, em relação a um determinado peso, muda entre duas iterações consecutivas do algoritmo, a taxa de aprendizado para aquele peso deve ser diminuída.

2.2.4 Regra de Aprendizado Delta-Bar-Delta

Para proceder as modificações do algoritmo BP através da adaptação da taxa de aprendizado, emprega-se a função (2.1). O cálculo local num neurônio da rede MLP pode ser expresso através das seguintes expressões:

$$v_j(n) = \sum_i w_{ji}(n) y_i(n) \quad (2.19)$$

$$w_{ji}(n) = w_{ji}(n-1) - \eta_{ji}(n) \frac{\partial E(n-1)}{\partial w_{ji}(n-1)} \quad (2.20)$$

para as quais, na n -ésima iteração, $y_j(n)$ é a saída do neurônio j ; $\eta_{ji}(n)$ é a taxa de aprendizado associada ao peso $w_{ji}(n)$. Aplicando-se a regra da cadeia para calcular a derivada de $E(n)$ em relação à taxa de aprendizado, resulta:

$$\frac{\partial E(n)}{\partial \eta_{ji}(n)} = \frac{\partial E(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial \eta_{ji}(n)} \quad (2.21)$$

Resolvendo-se as derivadas no lado direito de (2.21), obtém-se,

$$\frac{\partial E(n)}{\partial \eta_{ji}(n)} = - \frac{\partial E(n)}{\partial w_{ji}(n)} \frac{\partial E(n-1)}{\partial w_{ji}(n-1)} \quad (2.22)$$

Logo, (2.22) define a derivada da função erro com relação à taxa de aprendizado $\eta_{ji}(n)$, admitindo que o neurônio j encontra-se na camada de saída de rede. De fato, demonstra-se que, (2.22) também se aplica quando o neurônio j está localizado numa camada oculta, Jacobs (1988). Conhecendo estas direções de pesquisa, (2.22), a regra para a adaptação da taxa de aprendizado é formulada. Especificamente, através do método do gradiente:

$$\Delta \eta_{ji}(n+1) = -\gamma \frac{\partial E(n)}{\partial \eta_{ji}(n)} = \gamma \frac{\partial E(n)}{\partial w_{ji}(n)} \frac{\partial E(n-1)}{\partial w_{ji}(n-1)} \quad (2.23)$$

em que γ é uma constante positiva. O procedimento descrito por (2.23) é chamado de regra de aprendizado delta-delta, Jacobs (1988), que atende as heurísticas 3 e 4 anteriores.

A principal limitação desta regra é a escolha adequada do parâmetro γ . Isto deve-se à influência das magnitudes das derivadas parciais na otimização da taxa de aprendizado, (2.23). Devido a esta limitação, algumas modificações foram feitas por Jacobs (1988). Neste sentido, a regra de ajuste para a taxa de aprendizado $\eta_{ji}(n)$ relativa ao peso $w_{ji}(n)$ é reformulada como se segue:

$$\Delta \eta_{ji}(n+1) = \begin{cases} \kappa & , S_{ji}(n-1)D_{ji}(n) > 0 \\ -\beta \eta_{ji}(n) & , S_{ji}(n-1)D_{ji}(n) < 0 \\ 0 & , \text{c.c.} \end{cases} \quad (2.24)$$

para a qual, $D_{ji}(n)$ e $S_{ji}(n)$ são obtidas por:

$$D_{ji}(n) = \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (2.25)$$

$$S_{ji}(n) = (1 - \xi)D_{ji}(n-1) + \xi S_{ji}(n-1) \quad (2.26)$$

em que, ξ é uma constante positiva. A estratégia de adaptação da taxa de aprendizado, definida através de (2.24)-(2.26), denomina-se regra delta-bar-delta, Jacobs (1988). Os valores usuais para os parâmetros do algoritmo são: $\kappa = 0,01, \beta = 0,2, \xi = 0,7$. Haykin (1999). Se os parâmetros κ e β forem nulos, o algoritmo resultante equívale ao da *backpropagation*.

2.2.5 Algoritmo SuperSAB

O algoritmo SuperSAB, Tollenaere (1990), também baseia-se na dependência da adaptação da taxa de aprendizado com relação aos sinais das derivadas parciais, justo como descrito na regra delta-bar-delta. A mudança principal é o aumento exponencial da taxa de aprendizado, (2.27), que na regra delta-bar-delta é linear. Isto é feito para aumentar a amplitude de variação das taxas de aprendizado.

$$\eta_{ji}(n) = \begin{cases} \eta^+ \eta_{ji}(n-1) & , \quad \frac{\partial E(n-1)}{\partial w_{ji}(n-1)} \frac{\partial E(n)}{\partial w_{ji}(n)} > 0 \\ \eta^- \eta_{ji}(n-1) & , \quad \frac{\partial E(n-1)}{\partial w_{ji}(n-1)} \frac{\partial E(n)}{\partial w_{ji}(n)} < 0 \\ \eta_{ji}(n-1) & , \quad c.c. \end{cases} \quad (2.27)$$

com $0 < \eta^- < 1 < \eta^+$. Além disso, no caso de mudança de sinal de duas derivadas sucessivas, o ajuste anterior dos pesos é revertido.

Se comparados em termos de convergência, o algoritmo SuperSAB apresenta um desempenho superior ao algoritmo BP. Uma desvantagem do algoritmo SuperSAB é o

número elevado de parâmetros a se determinar: taxa de aprendizado inicial, termo momento e os fatores de aumento e diminuição da taxa de aprendizado. Por outro lado, os algoritmos de adaptação da taxa de aprendizado, descritos até agora, sofrem com a influência indesejável da amplitude da derivada parcial sobre o ajuste dos pesos, isto é:

$$\Delta w_{ji}(n) = -\eta_{ji}(n) \frac{\partial E(n)}{\partial w_{ji}(n)} + \alpha \Delta w_{ji}(n-1) \quad (2.28)$$

Durante o processo de aprendizado do MLP, a amplitude da derivada em (2.28) pode ter um efeito imprevisível sobre a direção de pesquisa do mínimo da função erro. Na seção seguinte, aborda-se uma estratégia para transpor esta limitação.

2.2.6 Algoritmo Rprop

O Rprop (*Resilient backpropagation*) é um algoritmo de treinamento com taxa de aprendizado adaptativa, Riedmiller e Braun (1993), Riedmiller (1994), Moreira e Fiesler (1995). O princípio básico do Rprop é a eliminação da influência prejudicial das derivadas parciais na otimização dos pesos do MLP. Como consequência, para indicar a direção de ajuste dos pesos, apenas o sinal destas derivadas é levado em conta. Nesta otimização, as amplitudes dos ajustes são determinadas através do valor de ajuste Δ_{ji} , exclusivo para cada peso:

$$\Delta w_{ji}(n) = \begin{cases} -\Delta_{ji}(n) & , \quad \frac{\partial E(n)}{\partial w_{ji}(n)} > 0 \\ +\Delta_{ji}(n) & , \quad \frac{\partial E(n)}{\partial w_{ji}(n)} < 0 \\ 0 & , \quad c.c. \end{cases} \quad (2.29)$$

O segundo passo do aprendizado Rprop é a determinação dos novos valores de ajuste $\Delta_{ji}(n)$. Baseado no processo de adaptação dependente do sinal, semelhante a adaptação da taxa de aprendizado em (2.27), estes valores são determinados por:

$$\Delta_{ji}(n) = \begin{cases} \eta^+ \Delta_{ji}(n-1) & , \quad \frac{\partial E(n-1)}{\partial w_{ji}(n-1)} \frac{\partial E(n)}{\partial w_{ji}(n)} > 0 \\ \eta^- \Delta_{ji}(n-1) & , \quad \frac{\partial E(n-1)}{\partial w_{ji}(n-1)} \frac{\partial E(n)}{\partial w_{ji}(n)} < 0 \\ \Delta_{ji}(n-1) & , \quad c.c. \end{cases} \quad (2.30)$$

em que, $0 < \eta^- < 1 < \eta^+$.

Inicialmente, todos os valores de ajuste são iguais a constante Δ_0 , que é um dos parâmetros do Rprop. Desde que Δ_0 determina diretamente a amplitude do primeiro ajuste dos pesos, ele pode ser escolhido de acordo com a magnitude dos pesos iniciais, por exemplo $\Delta_0 = 0.1$, Riedmiller (1994). A escolha deste valor não é crítica, já que seu valor é adaptado enquanto o treinamento prossegue.

No treinamento do MLP através do algoritmo Rprop, a fim de se evitar uma variação excessiva dos pesos, define-se o parâmetro valor máximo de ajuste, Δ_{max} . O valor $\Delta_{max} = 50$ foi sugerido por Riedmiller e Braun (1993). A dependência da convergência do algoritmo sobre este parâmetro é mínima. Contudo, em alguns casos, é preferível reduzir este valor. a fim de se prevenir uma possível convergência antecipada para algum mínimo local sub-ótimo.

Os fatores de acréscimo e decréscimo são fixados em $\eta^+ = 1,2$ e $\eta^- = 0,5$, respectivamente. Estes valores são baseados em considerações teóricas e empíricas. Portanto, o número de parâmetros fica reduzido a dois, Δ_0 e Δ_{max} , Riedmiller (1994), Riedmiller e Braun (1993), Schiffmann et al (1994).

Em resumo, o princípio básico do Rprop é a adaptação direta dos valores de ajuste para os pesos, Δ_{ji} . Com uma estratégia diferente dos algoritmos baseados na taxa de aprendizado descritos anteriormente, o Rprop modifica diretamente a amplitude de ajuste aplicada aos pesos, introduzindo o conceito de valores de ajuste resilientes. Como resultado, o esforço de adaptação não é atrapalhado pelo comportamento imprevisível do gradiente. Devido à simplicidade das leis de aprendizado, há apenas um ligeiro aumento no custo computacional com relação ao algoritmo BP.

2.3 REDES NEURAI E APROXIMAÇÃO DE FUNÇÕES

A idéia de aproximar funções através da superposição de funções básicas é antiga. Desde o início do século XIX, já se sabia, que, uma combinação linear de senos e cossenos podia ser utilizada para representar outras funções periódicas.

A fim de estabelecer o problema de interpolação (ou aproximação), considera-se as funções: $f(\mathbf{x})$, uma função contínua de n variáveis, $\mathbf{x}^T = [x_1, x_2, \dots, x_n]$; $F(\mathbf{W}, \mathbf{x})$, função definida para aproximar $f(\mathbf{x})$, com um número fixo de parâmetros \mathbf{W} , pertencentes a um dado conjunto P . Com a escolha de uma função F particular, o problema será encontrar $\mathbf{W} = (w_1, w_2, \dots, w_n)$, tal que, dado os exemplos $(\mathbf{x}; f(\mathbf{x}))$, a melhor aproximação possível para $f(\mathbf{x})$ seja encontrada, Poggio e Girosi (1989). Para tanto, a rede deve ser submetida a um processo de aprendizado. Define-se a aproximação ótima através da relação:

$$E[F(\mathbf{W}^*, \mathbf{x}), f(\mathbf{x})] \leq E[F(\mathbf{W}, \mathbf{x}), f(\mathbf{x})] \quad , \quad \forall \mathbf{W} \in P \quad (2.31)$$

em que, $E(\cdot)$ é uma medida do erro da aproximação.

Convém ressaltar a importância da escolha da função de aproximação F , que possa representar f , tão bem quanto possível. Uma má escolha de F torna o treinamento da rede lento, resultando normalmente num valor final elevado para o erro da aproximação. Na tarefa de aproximar funções, destacam-se três problemas principais:

- **Representação**: que tipo de aproximação usar? Isto é, que classes de funções $f(\mathbf{x})$ podem ser efetivamente aproximadas por $F(\mathbf{W}, \mathbf{x})$?
- **Treinamento**: para uma dada escolha de F , que algoritmo utilizar para a determinação dos valores ótimos dos parâmetros \mathbf{W} ?
- **Implementação**: como executar o algoritmo de forma eficiente?

Do ponto de vista de redes neurais para aproximação de funções, supõe-se que a rede funciona como memória associativa, que generaliza quando novas entradas são apresentadas. Quase todo esquema de aproximação pode ser mapeado em algum tipo de rede, que então é chamada de “rede neural”. Neste sentido, as redes neurais podem ser consideradas como uma notação gráfica para uma classe ampla de algoritmos. Neste contexto, uma rede neural serve para representar uma função como uma composição de muitas funções básicas.

Neste estudo, emprega-se a configuração com uma camada oculta, utilizada na maioria das aplicações. Definida com esta configuração, a rede apresentará a seguinte resposta:

$$z = F(\mathbf{W}, \mathbf{x}) = \sum_{h=1}^H v_h \varphi(\cdot) + v_0 \quad (2.32)$$

em que $\varphi(\cdot)$ é a função de ativação das unidades ocultas, v é o vetor de pesos entre a camada oculta e o nó de saída (linear). Definida a configuração da rede, pode-se modificar $F(\mathbf{W}, \mathbf{x})$ através de uma escolha apropriada de $\varphi(\cdot)$. Algumas funções de ativação, empregadas para diferentes tipos de redes sem realimentação, são dadas abaixo:

• Rede MLP:

– *sigmóide* $\varphi(\mathbf{x}) = \text{sig}(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x})} \quad (2.33)$

– *tangente hiperbólica* $\varphi(\mathbf{x}) = \frac{(\exp(\mathbf{x}) - \exp(-\mathbf{x}))}{(\exp(\mathbf{x}) + \exp(-\mathbf{x}))} \quad (2.34)$

– *arcotangente* $\varphi(\mathbf{x}) = \frac{2}{\pi} \arctan(\mathbf{x}) \quad (2.35)$

- Rede RBF:

– *Gaussiana* $\varphi(\mathbf{x}, \mathbf{t}_h) = \exp\left(-\frac{1}{\sigma_h^2} \|\mathbf{x} - \mathbf{t}_h\|\right)$ (2.36)

– *Multiquadratic* $\varphi(\mathbf{x}) = \frac{1}{(c^2 + \mathbf{x}^2)^a}, \quad a > 0$ (2.37)

- Rede de Wavelets:

– *wavelets polinomiais* $\varphi_n(\mathbf{x}) = \frac{d^n}{d\mathbf{x}^2} [\text{sig}(\mathbf{x})]$ (2.38)

– *função de Haar* $\varphi(\mathbf{x}) = \begin{cases} +1, & \text{se } 0 \leq \mathbf{x} \leq \frac{1}{2} \\ -1, & \text{se } \frac{1}{2} < \mathbf{x} \leq 1 \\ 0, & \text{c.c.} \end{cases}$ (2.39)

– *função chapéu mexicano* $\varphi(\mathbf{x}) = (1 - \mathbf{x}^2) \exp(-\frac{1}{2} \mathbf{x}^2)$ (2.40)

– *função de Littlewood-Paley* $\varphi(\mathbf{x}) = \frac{(\text{sen}(2\pi\mathbf{x}) - \text{sen}(\pi\mathbf{x}))}{\pi\mathbf{x}}$ (2.41)

- Rede de Fourier: $\varphi(\mathbf{x}, \theta_h) = \text{sen}(n\mathbf{x} + \theta_h)$ (2.42)

- Rede de Funções *Sample*: $\varphi(\mathbf{x}) = \frac{\text{sen}(\mathbf{x})}{\mathbf{x}}$ (2.43)

Em seguida, apresentar-se-á uma descrição resumida de outras redes neurais com propagação direta que foram implementadas. Destaca-se as funções de ativação empregadas em cada caso, bem como, as relações recursivas para adaptação dos seus parâmetros livres.

2.4 REDE RBF

A configuração de uma rede de funções de base radiais, na sua forma mais básica (com um nó de saída) é apresentada na Fig. 2.6. Ela consiste de uma camada de entrada (sensores), uma camada oculta com unidades de processamento não lineares, e, uma camada de saída linear. De acordo com a notação utilizada na Fig. 2.6, a saída da rede RBF é dada por:

$$z = \sum_{h=1}^H v_h \varphi(\mathbf{x}, \mathbf{t}_h) + v_0 \tag{2.44}$$

$$\varphi(\mathbf{x}, \mathbf{t}_h) = \exp\left(-\frac{1}{\sigma_h^2} \|\mathbf{x} - \mathbf{t}_h\|\right) \tag{2.45}$$

em que, o símbolo $\|\cdot\|$ representa a norma Euclidiana, \mathbf{t}_h é o vetor de centros, e σ_h é o desvio padrão (uma medida de espalhamento dos centros), referentes à h -ésima função de base radial Gaussiana, Poggio e Girosi (1990), Poggio et al (1993).

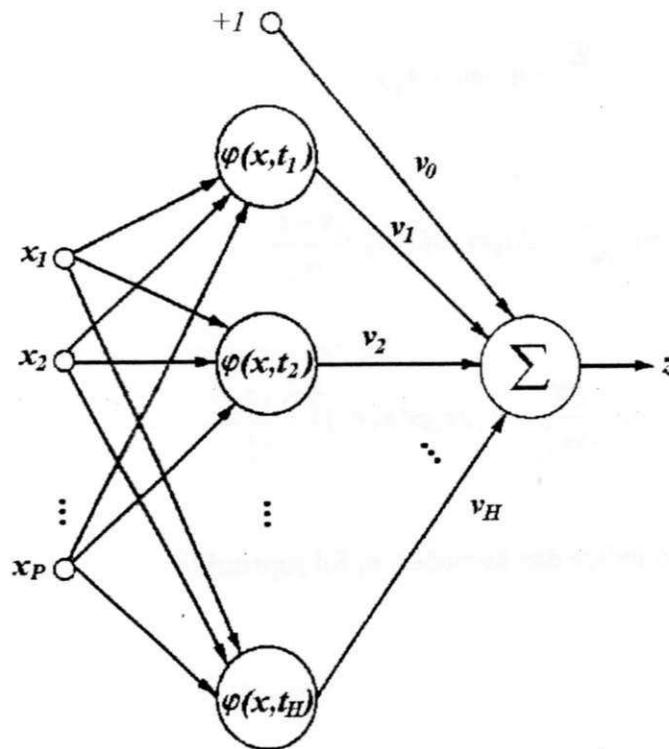


Fig. 2.6 Rede RBF Gaussiana.

A rede RBF possui apenas uma camada oculta não linear e a camada de saída é linear. O potencial de ativação de um neurônio oculto na rede RBF é dado em termos da distância Euclidiana entre o vetor de entradas e o vetor de centros das RBFs. Duas propriedades principais são observadas para esta rede: a construção de um mapa entrada/saída com características locais; um treinamento rápido, Haykin (1996) e (1999).

Dentre os tipos de treinamento da rede RBF Gaussiana, destacam-se: i) com seleção aleatória dos centros; ii) com seleção auto-organizada dos centros; iii) com seleção supervisionada dos centros, Poggio e Girosi (1990). Esta última estratégia, a mais geral, será utilizada aqui. A rede RBF é formulada com apenas um nó de saída, Fig. 2.6, sem perda de generalidade. Neste tipo de treinamento, os parâmetros livres da rede RBF são ajustados durante um processo de aprendizado supervisionado, Lowe (1989). O objetivo é minimizar uma função erro, por exemplo (2.1).

O método do gradiente é empregado no aprendizado supervisionado pela correção do erro, Poggio e Girosi (1990). Com o uso de (2.45), resultam as seguintes relações recursivas para otimizar os parâmetros da rede RBF:

$$\Delta v_h = -\eta_v \frac{\partial E}{\partial v_h} = \eta_v e \varphi(\mathbf{x}, \mathbf{t}_h) \quad (2.46)$$

$$\Delta \mathbf{t}_h = -\eta_t \frac{\partial E}{\partial \mathbf{t}_h} = 2\eta_h e v_h \varphi(\mathbf{x}, \mathbf{t}_h) \frac{\mathbf{x} - \mathbf{t}_h}{\sigma_h^2} \quad (2.47)$$

$$\Delta \sigma_h^2 = -\eta_\sigma \frac{\partial E}{\partial \sigma_h^2} = \eta_\sigma e v_h \varphi(\mathbf{x}, \mathbf{t}_h) \frac{\|\mathbf{x} - \mathbf{t}_h\|^2}{\sigma_h^2} \quad (2.48)$$

para as quais, o índice das iterações, n , foi suprimido.

2.5 REDE *WAVELET* POLINOMIAL

A idéia de combinar a teoria *wavelet* com redes neurais, Zhang (1997), Zhang e Benvensite (1992), Pati e Krishnaprasad (1993), resultou num tipo novo de rede neural, denominada rede *wavelet*. Esta rede é desenvolvida com o uso de funções de ativação *wavelets* para seus neurônios ocultos. As redes *wavelet* têm sido aplicadas à modelagem de dispositivos ativos e passivos de microondas, Harkouss et al (1998), Bila et al (1999) e Harkouss et al (1999).

As *wavelets* são funções que satisfazem certas propriedades matemáticas, sendo usadas para representar dados e outras funções. Na análise *wavelet*, a escala de resolução, que é usada para observar os dados, desempenha um papel fundamental. Os algoritmos *wavelet* processam os dados em diferentes resoluções. Por exemplo, se uma função for observada a partir de uma janela larga, seu aspecto geral é destacado. Por outro lado, empregando-se uma janela estreita, pequenos detalhes são realçados. O resultado da análise *wavelet* é, por assim dizer, “ver a floresta e as árvores”.

Dentre os diversos tipos de *wavelets*, aborda-se a construção de uma família de *wavelets* polinomiais, baseada em PPS (polinômios de potências de *sigmóide*). Isto requer alguns resultados pertencentes ao espaço de Hilbert e a idéia de *frames* para *wavelets*, que correspondem a uma generalização de bases ortogonais para o espaço de Hilbert, Marar et al (1996a).

Sucintamente, a partir dos resultados da teoria *wavelet*, dada uma função mãe $\varphi(\cdot)$, densa no espaço das funções quadrado integráveis, $L^2(\mathcal{R})$, qualquer função contínua arbitrária pode ser aproximada por uma soma finita de *wavelets*, que correspondem a translações, dilatações e ganhos da função mãe, Szu et al (1992):

$$z = \tilde{f}(x) = \sum_{h=1}^H a_h \varphi\left(\frac{x-t_h}{r_h}\right) \quad (2.49)$$

A construção de *wavelets* polinomiais baseadas em PPS é feita de modo formal através da teoria de séries de Taylor, Marar et al (1996a). Contudo, pode-se obter tais *wavelets* através de derivações sucessivas da função *sigmóide*, (2.33). A derivada primeira é dada por:

$$\varphi'(x) = \text{sig}(x)[1 - \text{sig}(x)] \tag{2.50}$$

A primeira função *wavelet* polinomial é gerada através da derivada segunda de $\varphi(x)$:

$$\varphi_2(x) = 2\text{sig}(x)^3 - 3\text{sig}(x)^2 + \text{sig}(x) \tag{2.51}$$

A Fig. 2.7 ilustra as nove primeiras funções polinomiais $\varphi_2, \varphi_3, \dots, \varphi_9$.

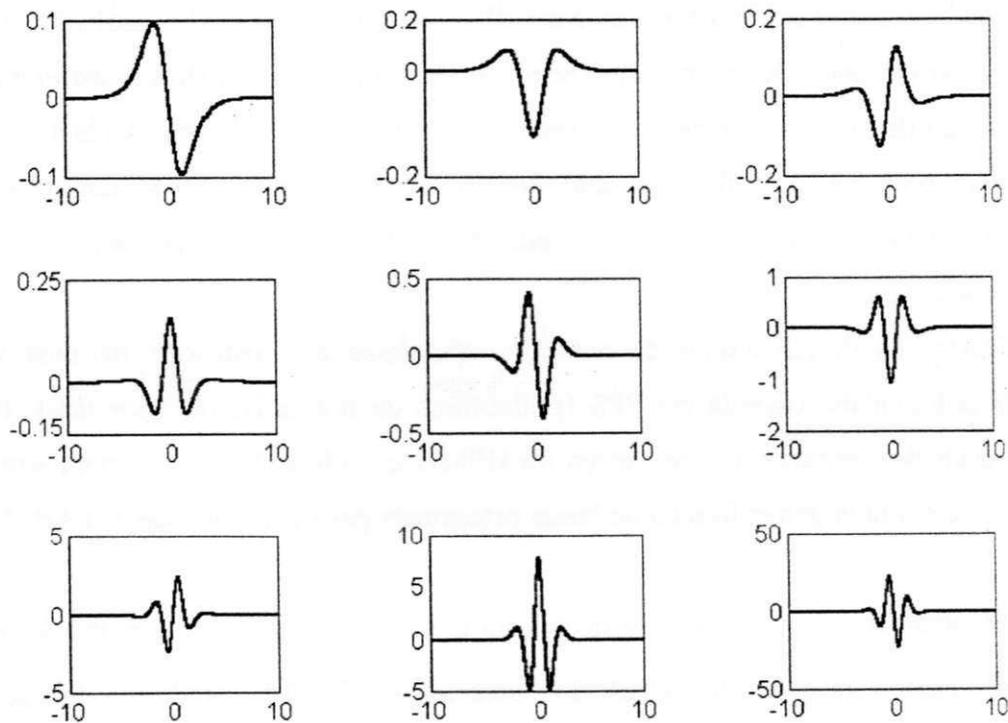


Fig. 2.7 Elementos $\varphi_2, \varphi_3, \dots, \varphi_{10}$ pertencentes a família de *wavelets* polinomiais.

As técnicas que combinam redes neurais e *wavelets* têm sido analisada, Daugman (1988), Pati e Krishnaprasad (1993), Marar et al (1996b) e Szu et al (1992). Por exemplo, no processamento de imagens, as redes neurais são empregadas para a determinação dos coeficientes da *wavelet* de Gabor, Daugman (1988). A seguinte *wavelet* foi empregada na classificação de fonemas, Szu et al (1992),

$$\varphi(x) = \cos(1.75x) \cdot \exp(-x^2 / 2) \tag{2.52}$$

A representação de uma função contínua como proposto por Szu et al (1992), (2.49), sugere a construção da rede neural, representada na Fig. 2.8.

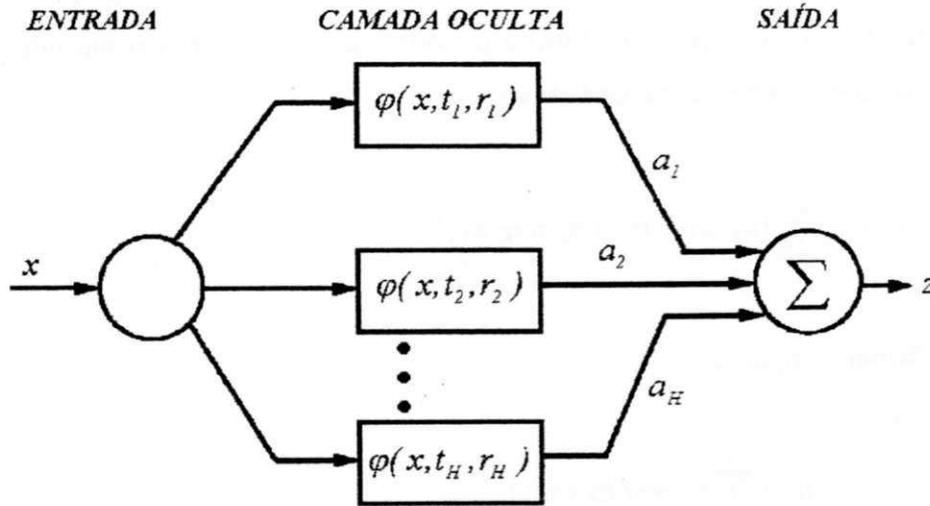


Fig. 2.8 Rede neural baseada em *wavelets* polinomiais.

Os parâmetros da rede (a_h, t_h, r_h) , $h = 1, 2, \dots, H$, são adaptados na direção do mínimo da função erro, (2.1). O processo de aprendizado requer o uso de algum método de otimização. Através do método gradiente e usando-se o elemento φ_2 da família de *wavelets* polinomiais, resultam as seguintes relações recursivas para otimizar os parâmetros livres da rede:

$$\Delta a_h = -\eta \nabla_{a_h} E = -\eta e \varphi_2(\theta) \quad (2.53)$$

$$\Delta t_h = -\eta \nabla_{t_h} E = -\eta \frac{e a_h}{r_h} (-6 \text{sig}^4(\theta) + 12 \text{sig}^3(\theta) - 7 \text{sig}^2(\theta) + \text{sig}(\theta)) \quad (2.54)$$

$$\Delta r_h = -\eta \nabla_{r_h} E = -\eta \frac{e a_h (x - t_h)}{r_h^2} (1 - \text{sig}(\theta)) (\text{sig}(\theta) - 6 \text{sig}^2(\theta) + 6 \text{sig}^3(\theta)) \quad (2.55)$$

em que,

$$\theta = (x - t_h) / r_h \quad (2.56)$$

2.6 REDE DE FOURIER

Apesar das funções *sigmóides* serem as mais usuais, outros tipos de funções de ativação são possíveis no projeto de redes neurais sem realimentação. Por exemplo, através da análise de Fourier, sabe-se que, qualquer função periódica pode ser escrita como uma série infinita de termos seno e cosseno (série de Fourier),

$$f(x) = \sum_{h=0}^{\infty} (a_h \cos(hx) + b_h \sin(hx)) \quad (2.57)$$

ou em forma compacta:

$$f(x) = a_0 + \sum_{h=1}^{\infty} r_h \sin(hx + q_h) \quad (2.58)$$

em que,

$$r_h = \sqrt{a_h^2 + b_h^2} \quad (2.59)$$

$$q_h = \arctan(b_h / a_h) \quad (2.60)$$

Esta tipo de representação pode ser visto como uma rede neural, com uma entrada, x ; unidades ocultas com função de ativação do tipo (2.42); e, uma saída $z = f(x)$. A representação da rede de Fourier, com H unidades ocultas (harmônicas) é ilustrada na Fig. 2.9. O fator a_0 corresponde à polarização da unidade de saída; os elementos do vetor de pesos, p , são inteiros ($p_h = h$); o parâmetro q_h , corresponde à polarização do h -ésimo neurônio oculto; v_h é o peso do entre o h -ésimo neurônio oculto e o neurônio de saída.

Uma diferença entre as configurações da rede de Fourier e do MLP está na definição dos pesos entre as camadas de entrada e oculta. Para a rede de Fourier, uma parte destes pesos, p , é definida a priori, (2.58). Enquanto isso, para o MLP todos os pesos são ajustados durante a fase de treinamento.

Admitindo-se uma função erro quadrática, (2.1), a função de ativação, (2.42), e, aplicando-se o método do gradiente, foram obtidas as seguintes relações recursivas para a otimização dos parâmetros livres da rede de Fourier:

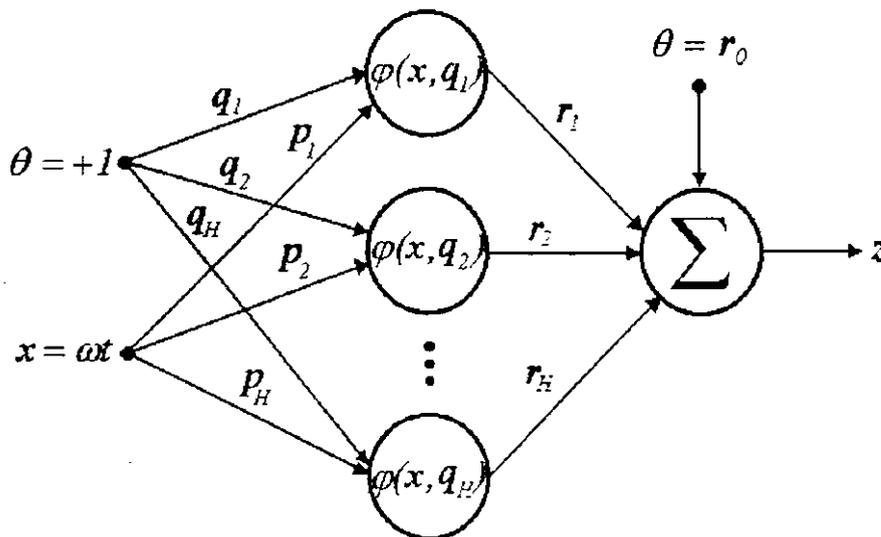


Fig. 2.9 Rede de Fourier.

$$\Delta q_h = -\eta \nabla_{b_h} E = r_h \eta e \cos(h\omega t + q_h) \tag{2.61}$$

$$\Delta r_h = -\eta \nabla_{c_h} E = \eta e \text{sen}(h\omega t + q_h) \quad h = 1, 2, \dots, H \tag{2.62}$$

A simplicidade de (2.61)-(2.62) resulta num algoritmo eficiente para o treinamento da rede de Fourier. Em particular, destaca-se sua aplicação à aproximação de funções periódicas. Nas simulações, calculou-se o parâmetro a_0 como a média das amostras de treinamento.

Na seção 2.8, aplicar-se-á a rede de Fourier para aproximar uma função senoidal. No Capítulo 5, desenvolve-se uma técnica de balanço harmônico simplificada, que emprega estas redes. Neste caso, a função erro (2.1) é redefinida e o treinamento se dá no modo *batch*.

2.7 REDE DE FUNÇÕES *SAMPLE*

Na modelagem de dispositivos de RF/microondas, há certas situações, para as quais, devido a imprecisão das aproximações obtidas, as redes neurais mais populares tornam-se inadequadas. Por exemplo, no emprego das redes MLPs, encontra-se dificuldades para aproximar funções que apresentem comportamentos localizados, isto é: regiões com não linearidades suaves e regiões com ondulações e não linearidades abruptas.

Este problema de representação das redes neurais usuais serviu como motivação para proceder com a modificação da resposta dos seus neurônios ocultos. Desta forma, consegue-se uma mudança de representação das redes neurais sem realimentação, através da definição de outras funções de ativação para seus neurônios.

Nesta seção, desenvolve-se uma rede neural com uma função de ativação baseada na função *sample*. Esta escolha foi originalmente motivada pela constatação das propriedades da função *sinc* na reconstrução de sinais, bem como, sua importância em processamento de sinais, Oppenheim e Willsky (1983). Estas propriedades de são descritas em seguida.

A interpolação é um procedimento comum para se reconstruir uma função, aproximada ou exatamente, a partir de suas amostras. Conforme a teoria de sinais e sistemas, Oppenheim e Schaffer (1975), Oppenheim e Willsky (1983), um sinal limitado em banda, amostrado numa frequência adequada, pode ser reconstruído exatamente a partir de suas amostras. Para isto, emprega-se um filtro passa-baixas, Fig. 2.10. Quando considera-se o efeito no tempo de um filtro passa-baixas ideal, a interpretação da reconstrução do sinal como um processo de interpolação torna-se evidente. Em particular, a saída $x_r(t)$ do filtro vale

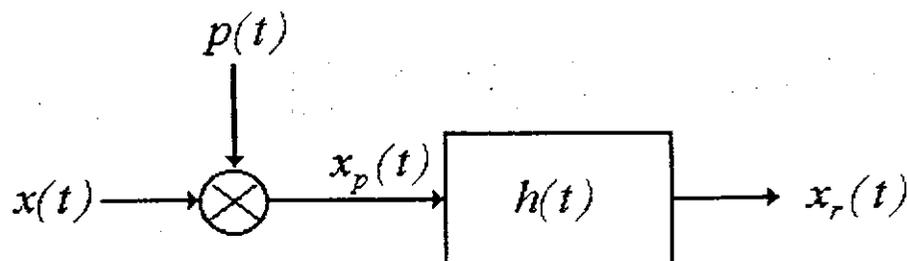


Fig. 2.10 Reconstrução de um sinal a partir de suas amostras.

$$x_r(t) = x_p(t) * h(t) = \sum_{n=-\infty}^{+\infty} x(nT)h(t - nT) \quad (2.63)$$

$$p(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT), \quad (2.64)$$

$$x_p(t) = \sum_{n=-\infty}^{+\infty} x(nT) \delta(t - nT) \quad (2.65)$$

Para um filtro passa-baixas ideal, tem-se:

$$h(t) = T \frac{\omega_c}{\pi} \operatorname{sinc}\left(\frac{\omega_c t}{\pi}\right) \quad (2.66)$$

resultando,

$$x_r(t) = T \frac{\omega_c}{\pi} \sum_{n=-\infty}^{+\infty} x(nT) \operatorname{sinc}\left(\frac{\omega_c(t - nT)}{\pi}\right) \quad (2.67)$$

em que, o sinal original é recuperado, a partir de suas amostras, através de uma combinação linear de funções *sinc*.

Com base no resultado (2.67), escolheu-se a função *sample*, bastante semelhante à função *sinc*, como função de ativação da ANN. Em seguida, atribuiu-se parâmetros ao seu argumento, como é feito para os casos da rede RBF (que usa centros espalhados para funções Gaussianas) e da rede *wavelet* (que usa translações e dilatações de uma *wavelet* mãe). Assim, através da distância Euclidiana entre os vetores de entradas \mathbf{x} e de centros \mathbf{t}_h , as funções *sample* são deslocadas; enquanto isso, σ_h determina o seu espalhamento. Deste modo, tem-se:

$$\varphi(\mathbf{x}, \mathbf{t}_h) = \operatorname{sample}(\sigma_h \|\mathbf{x} - \mathbf{t}_h\|^2) = \frac{\operatorname{sen}(\sigma_h \|\mathbf{x} - \mathbf{t}_h\|^2)}{\sigma_h \|\mathbf{x} - \mathbf{t}_h\|^2} \quad (2.68)$$

Com a influência destes parâmetros, a função (2.68) pode apresentar diferentes formas, Fig. 2.11. Esta é uma característica importante desta função, que torna bastante interessante seu

emprego como função de ativação dos neurônios de uma ANN. Por exemplo, esta característica aumenta a classe de mapeamentos, que uma ANN, assim definida, é capaz de aproximar.

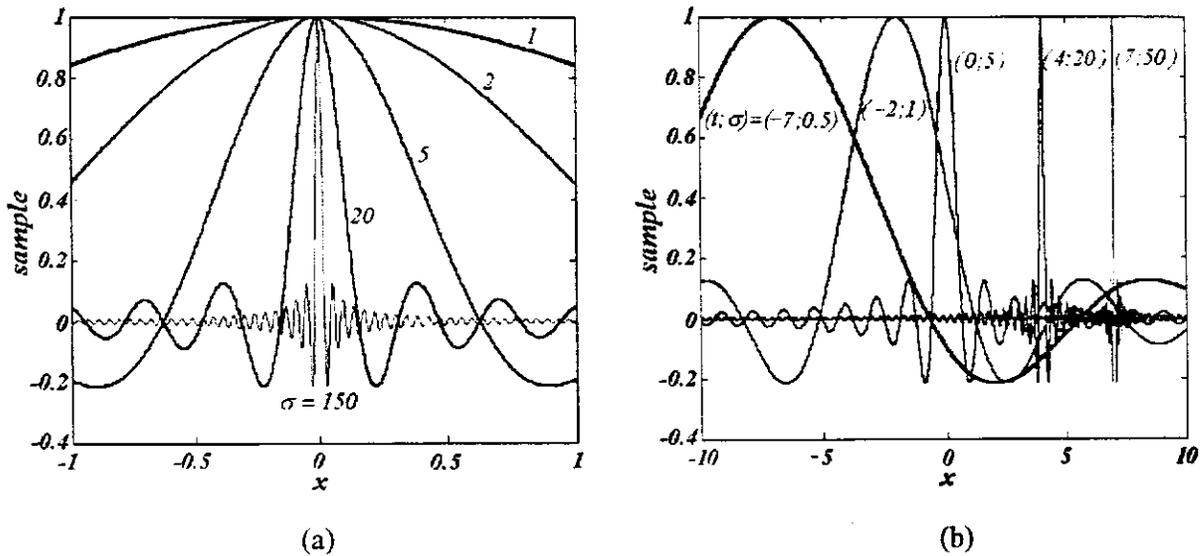


Fig. 2.11 Formas da função *sample* (2.68), conforme variações dos parâmetros:

(a) largura. (b) largura e centro.

A configuração de uma rede de funções *sample*, na sua forma mais básica (com um nó de saída) é representada na Fig. 2.12. Ela consiste de uma camada de entrada (sensores), uma camada oculta de unidades de processamento não lineares e uma camada de saída linear. De acordo com a notação da Fig. 2.12, a saída da RFS é dada por:

$$z = \sum_{h=1}^H v_h \varphi(\mathbf{x}, \mathbf{t}_h) + v_0 \quad (2.69)$$

$$y_h = \varphi(\mathbf{x}, \mathbf{t}_h) = \text{sample}(\text{net}_h) \quad (2.70)$$

$$\text{sample}(x) = \frac{\text{sen}(x)}{x} \quad (2.71)$$

$$\text{net}_h = \sigma_h \|\mathbf{x} - \mathbf{t}_h\|^2 = \sigma_h (\mathbf{x} - \mathbf{t}_h)^T \cdot (\mathbf{x} - \mathbf{t}_h) \quad (2.72)$$

Com a utilização do método do gradiente, resultam as seguintes relações recursivas para a otimização dos parâmetros livres da rede RFS:

$$\Delta v_0 = \eta_v e \tag{2.73}$$

$$\Delta v_h = \eta_v e y_h \tag{2.74}$$

$$\Delta \sigma_h = \eta_\sigma e v_h (\cos(\text{net}_h) - y) / \sigma_h \tag{2.75}$$

$$\Delta t_h = 2\eta_t e v_h \sigma_h (x - t_h) (y - \cos(\text{net}_h)) / \text{net}_h \tag{2.76}$$

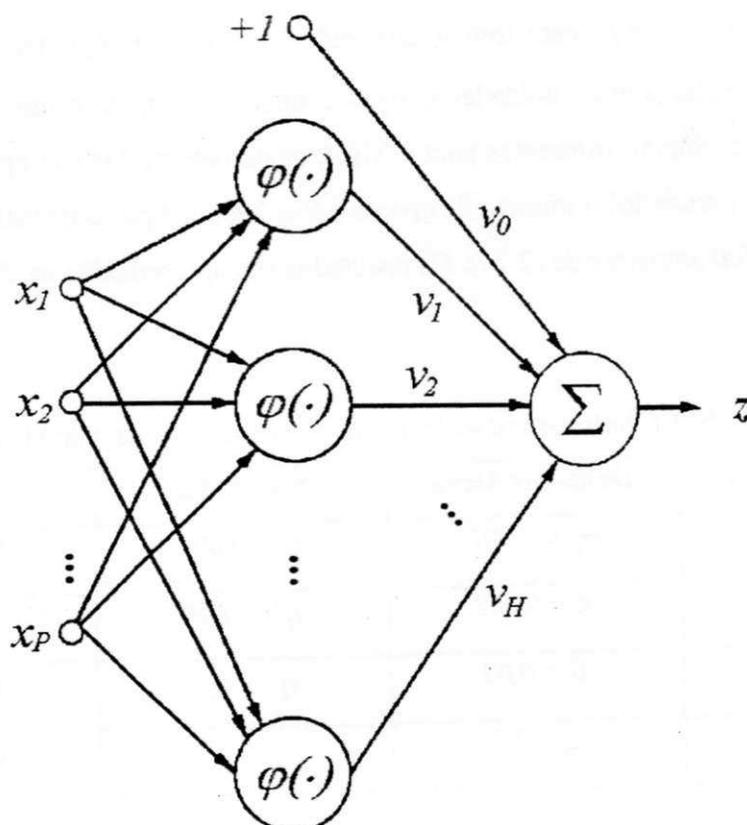


Fig. 2.12 Rede de funções *sample*.

2.8 REDES NEURAIIS E APROXIMAÇÃO DE FUNÇÕES – RESULTADOS DE SIMULAÇÃO

Nesta seção, verifica-se as propriedades de convergência dos algoritmos de treinamento descritos neste capítulo. Através de tarefas simples de aproximação, verifica-se o desempenho de cada algoritmo.

Por exemplo, para verificar o desempenho dos algoritmos desenvolvidos para a rede MLP, utilizou-se 50 exemplos da função

$$y(x) = \text{sig}(x)[\cos(x) - \text{sen}(x)] \quad (2.77)$$

no intervalo $x \in [-6, 6]$, para treinar uma rede MLP com a seguinte configuração: duas unidades de entrada, quinze unidades ocultas e uma unidade de saída. Utilizou-se quatro algoritmos de treinamento diferentes para o MLP, cujos parâmetros são apresentados na Tab. 2.1. O critério de parada foi o número de épocas ($N = 2000$). Após o treinamento, a rede MLP foi testada para 200 amostras de (2.77). Os resultados são apresentados na Fig. 2.13.

Tab. 2.1 Parâmetros dos algoritmos de treinamento da rede MLP.

<i>Backpropagation</i>	Delta-Bar-Delta	SuperSAB	RPROP
$\eta = 0,007$	$\eta_0 = 0,007$	$\eta_0 = 0,007$	$\Delta_0 = 0,007$
	$\kappa = 0,002$	$\eta^+ = 1,05$	$\Delta_{max} = 2$
	$\beta = 0,03$	$\eta^- = 0,7$	$\eta^+ = 1,2$
	$\xi = 0,7$		$\eta^- = 0,5$

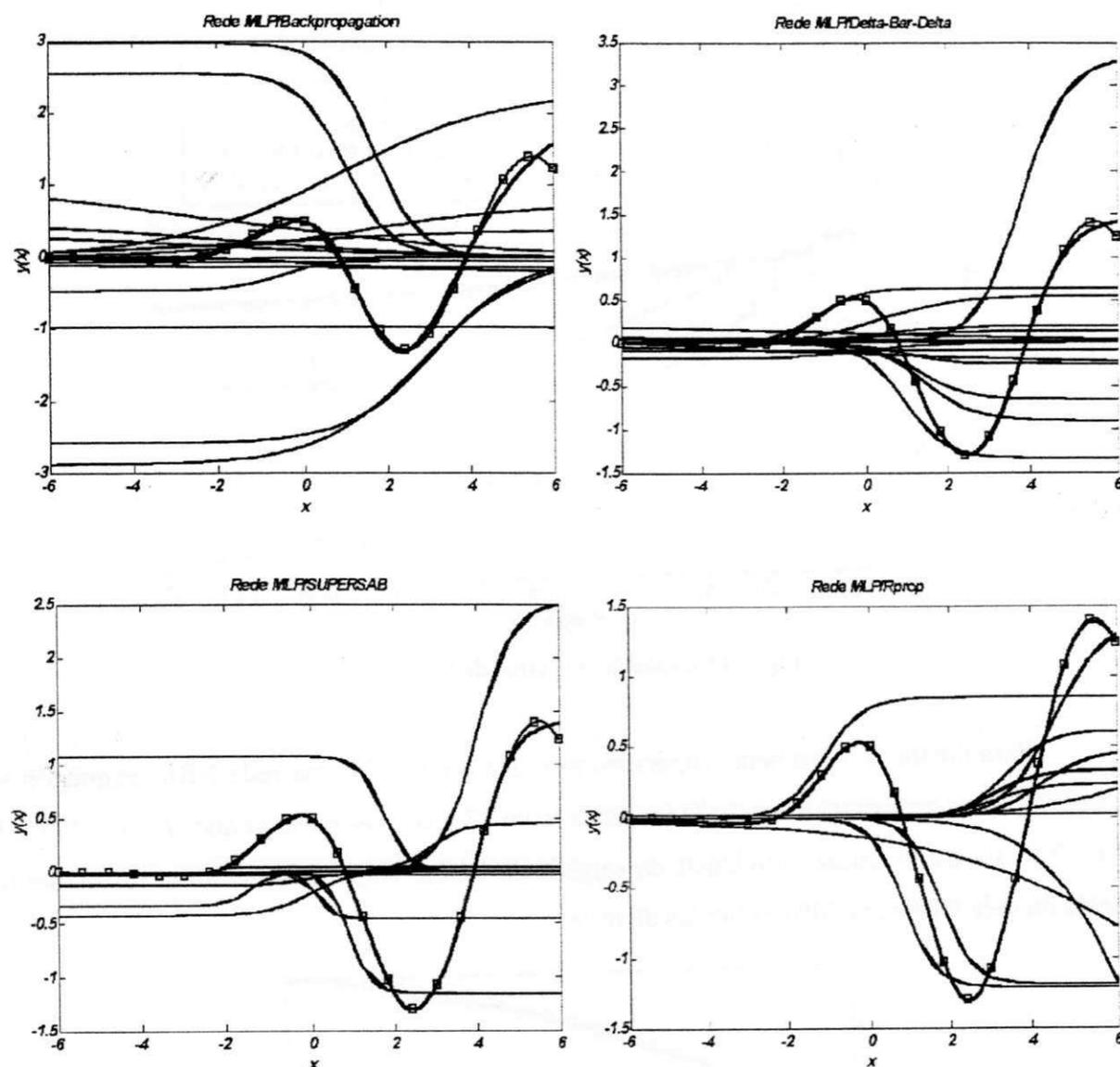


Fig. 2.13 Aproximação para (2.77) através da rede MLP, com o emprego de diferentes algoritmos de treinamento.

Na Fig. 2.14 apresenta-se a evolução dos erros das redes MLPs, quando os algoritmos descritos acima são empregados. Destaca-se as oscilações do algoritmo BP, devido a sua trajetória aleatória na busca do mínimo da função erro. Além disso, com relação às outras estratégias de treinamento, verifica-se a convergência superior do algoritmo Rprop.

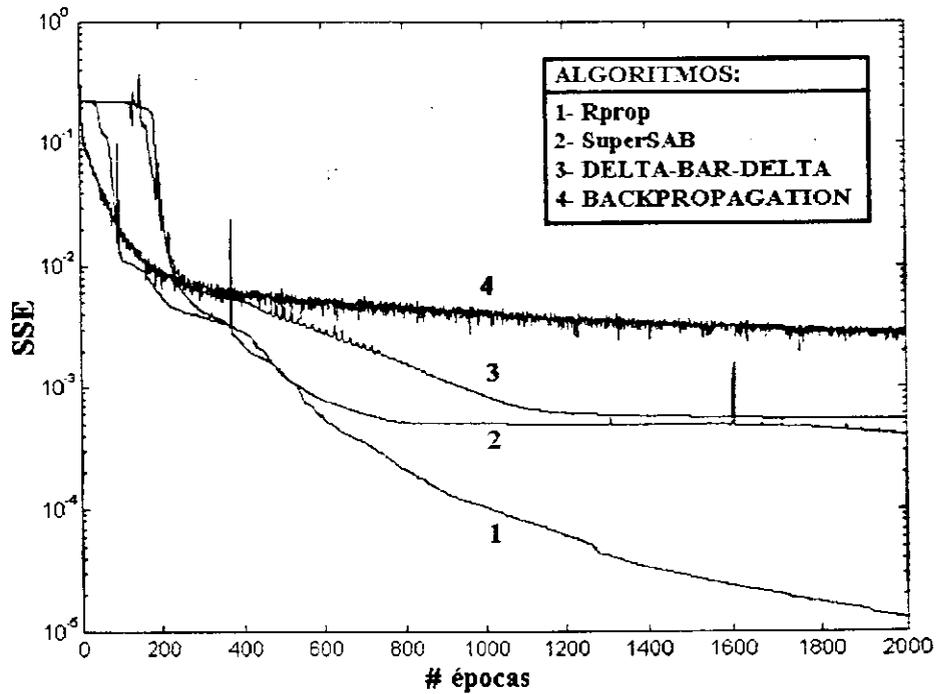


Fig. 2.14 Evolução dos erros de treinamento.

Para ilustrar o treinamento supervisionado dos centros de uma rede RBF, empregou-se dez RBFs para aproximar a função logarítmica (com 17 amostras no intervalo $1 \leq x \leq 100$). A Fig. 2.15 ilustra as saídas individuais da camada de RBFs, bem como, a soma resultante na saída da rede RBF para 1000 exemplos de teste.

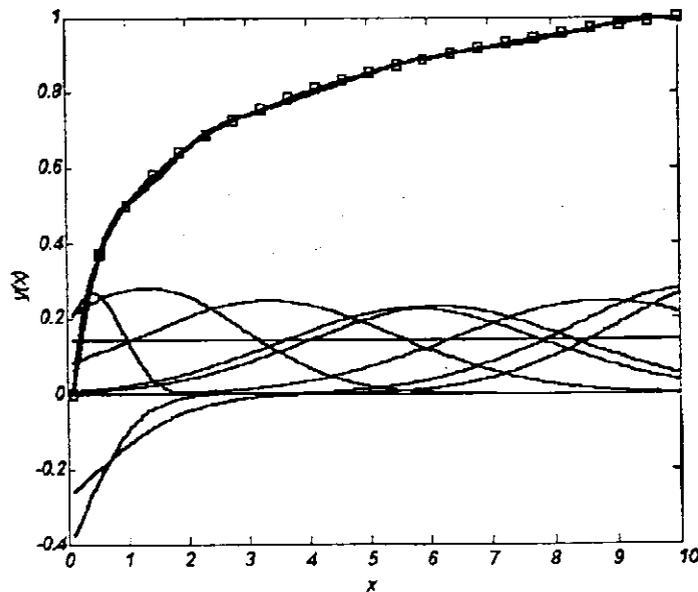


Fig. 2.15 Aproximação da Rede RBF através de funções RBFs Gaussianas.

Para a função exemplo (2.77), com 90 amostras no intervalo $[-6, 6]$, usou-se (2.53)-(2.56) no treinamento supervisionado da rede de *wavelets* polinomiais. Destaca-se a excelente aproximação obtida neste caso, Fig. 2.16.

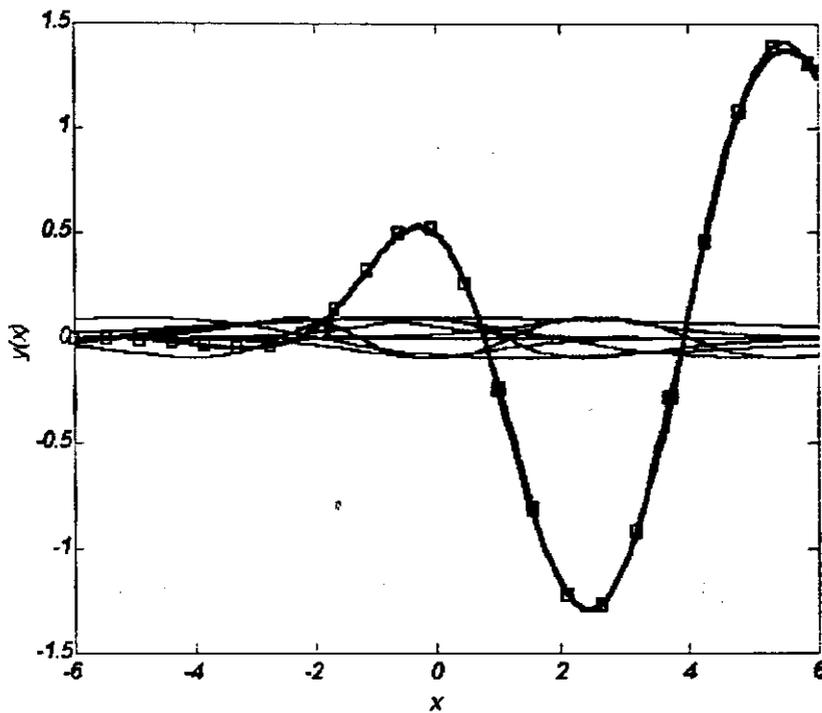


Fig. 2.16 Aproximação da rede de *wavelets* polinomiais para a função exemplo (2.77).

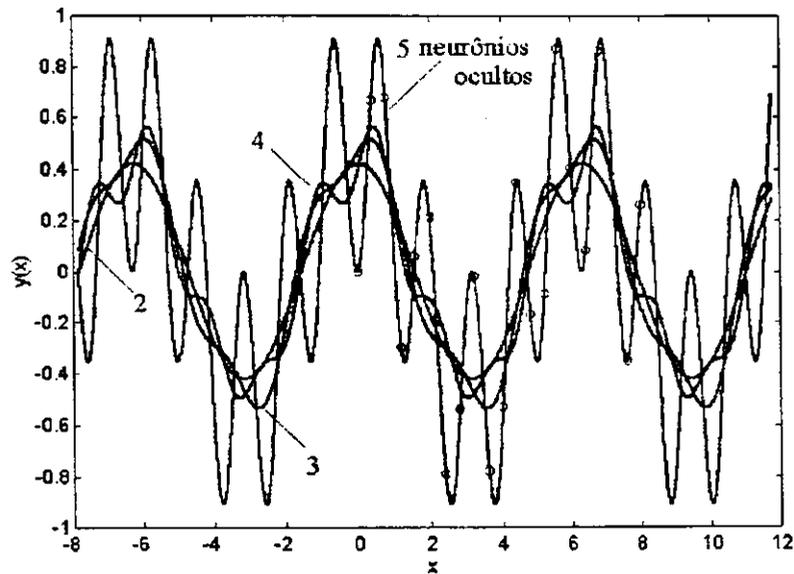
Para a rede de Fourier, um exemplo simples demonstra a capacidade desta rede, treinada através das relações (2.61)-(2.62), para aproximar e extrapolar funções periódicas. Por exemplo, para a função,

$$y(x) = \text{sen}(2x)\text{sen}(3x) \quad (2.78)$$

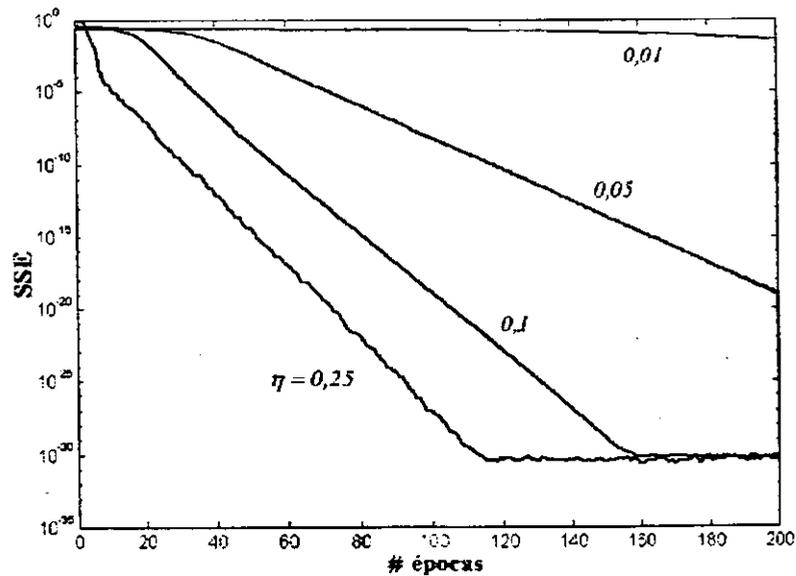
com 20 amostras no intervalo $x \in [0, 8]$. A Fig. 2.17(a) ilustra a aproximação através da rede de Fourier com duas, três, quatro e cinco unidades ocultas, após 200 épocas.

Para os resultados da aproximação de (2.78) através da rede de Fourier, Fig. 2.17, pode-se concluir que: (i) Com amostras suficientes da função periódica, abrangendo pelo menos um período, e, com um número apropriado de unidades ocultas, pode-se aproximar a

função periódica (2.78), através de uma rede de Fourier, até o nível de precisão numérica do computador digital onde (2.61)-(2.62) são calculadas. De acordo, nesta aproximação, o erro se estabiliza em torno de $SSE \approx 10^{-30}$, Fig. 2.17(b). (ii) Como era de se esperar, a rede de Fourier consegue extrapolar a função periódica, Fig. 2.17(a), com apenas cinco neurônios.



(a)



(b)

Fig. 2.17 (a) Aproximação através da rede de Fourier : (o) conjunto de treinamento, (—) generalização da rede de Fourier. (b) convergência do algoritmo de treinamento (10 neurônios ocultos).

Para se verificar o treinamento da rede RFS (principalmente naqueles mapeamentos, para os quais, outros tipos de redes sem realimentação falham), considera-se a aproximação dos dados do nível de atividade solar, entre os anos de 1700 e 1979 (280 exemplos). Na aproximação desta série através da rede RFS, empregou-se quatro configurações para a camada oculta (com 10, 50, 100 e 200 neurônios). A Fig. 2.18 apresenta os resultados obtidos, através dos quais, observa-se as propriedades de resolução da rede RFS. Conforme emprega-se um número maior de neurônios ocultos, maior resolução na aproximação é obtida. Em cada caso, as larguras estreitas são iniciadas com o mesmo valor, $\sigma_0 = 200$.

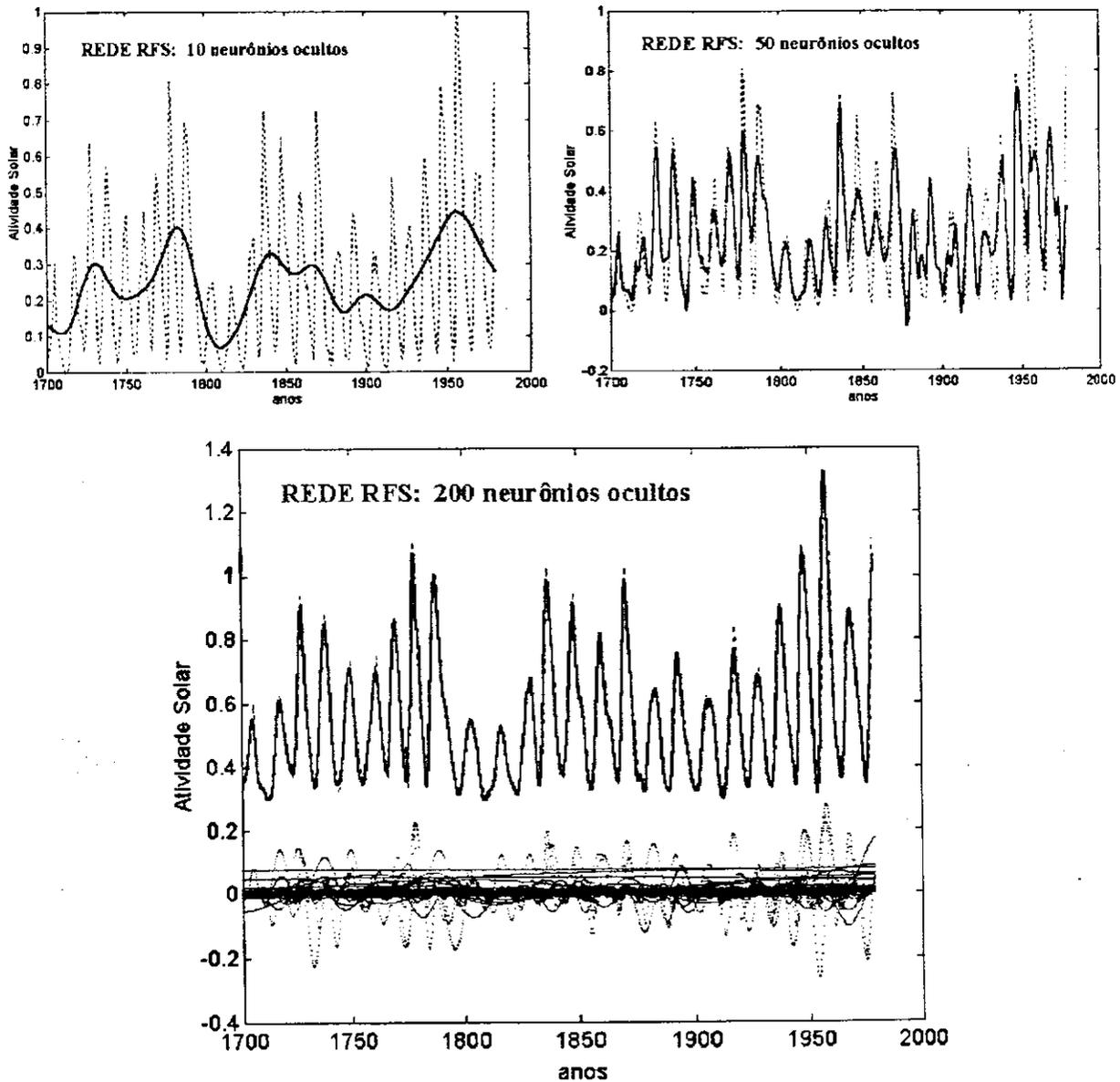


Fig. 2.18 Aproximações da série de atividade solar através de redes RFS.

A inspeção das curvas de erro de treinamento, Fig. 2.19, mostra que, quanto maior o número de neurônios ocultos, menor o erro da aproximação rede após o treinamento. Esta é uma propriedade desta rede, que pode ser justificada, através do resultado (2.67), em que, um sinal pode ser recuperado, a partir de suas amostras, através da combinação de infinitas funções *sinc*.

Com o emprego da rede RFS foram obtidos resultados excelentes de aproximação do nível de atividade solar, que são conseguidos apenas através de arquiteturas mais complexas, como a rede TDNN (*Time-Delay Neural Network*), Svarer et al (1993).

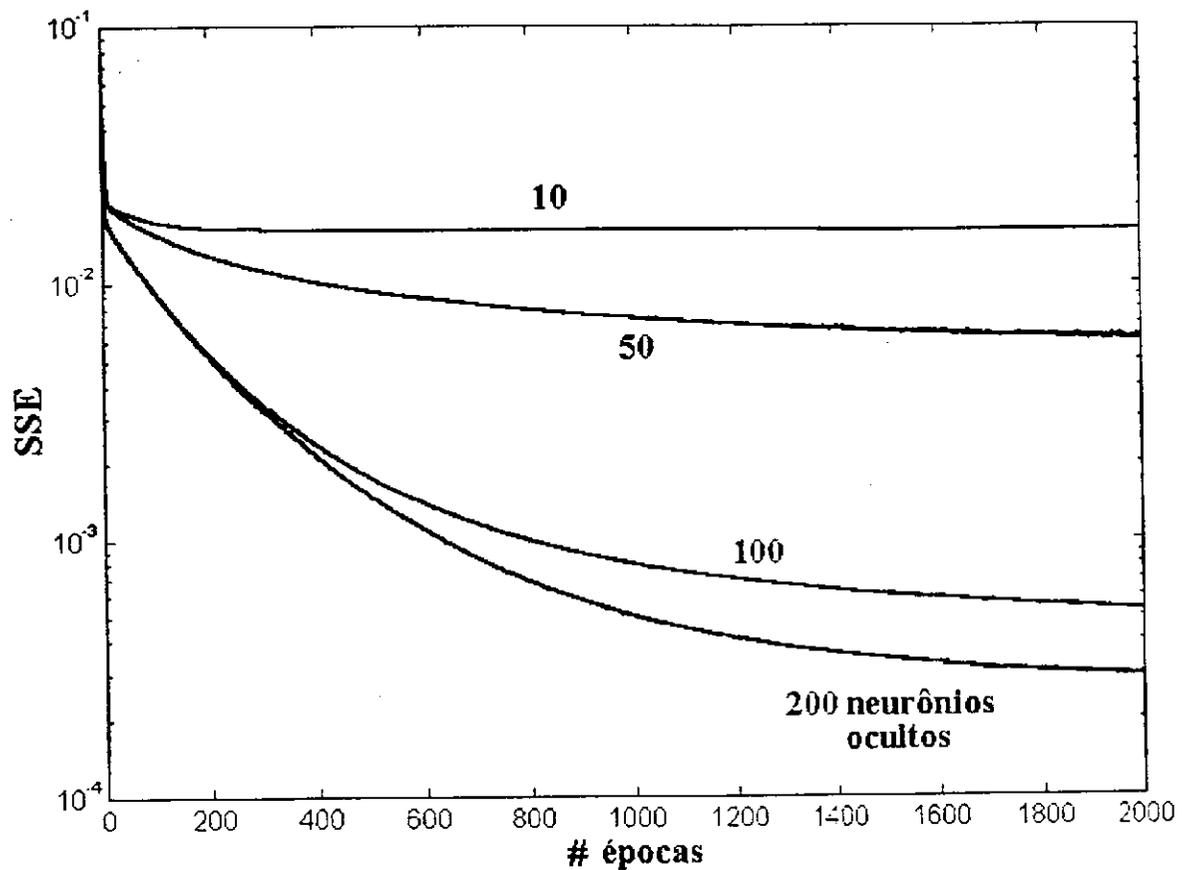


Fig. 2.19 Erro durante o treinamento da rede RFS, conforme o número de neurônios ocultos.

Capítulo 3

MODELOS ATRAVÉS DE REDES NEURAIIS BASEADAS EM RESULTADOS ELETROMAGNÉTICOS

3.1 INTRODUÇÃO

A precisão e a rapidez de modelos de dispositivos e circuitos de microondas são características fundamentais no projeto, na análise e em problemas de otimização. Contudo, precisão e rapidez nem sempre são possíveis na computação destes modelos.

Por exemplo, em simulações de dispositivos ativos, passivos e linhas de transmissão, a precisão dos modelos físicos é crucial. Neste caso, a complexidade das equações físicas, que envolvem as equações de Maxwell, aumenta bastante o custo computacional.

Por outro lado, na simulação de circuitos, para se encontrar a resposta de interesse, um sistema de equações deve ser solucionado, relacionando-se as tensões e as correntes em cada nó do circuito. A solução destas equações requer um grande esforço computacional, principalmente em circuitos mais complexos. Além disso, a análise e a otimização de circuitos são processos naturalmente iterativos, requisitando simulações repetitivas, e, aumentando ainda mais o tempo de computação.

Para contornar estes problemas, aproximações através de modelos polinomiais Barby et al (1988) e de tabelas de consultas, Jain et al (1983), têm sido previamente utilizadas durante a otimização, substituindo a computação dos modelos EMs de MOSFETs. Os modelos polinomiais limitam-se à aproximação de mapeamentos não lineares suaves. Por outro lado, uma desvantagem das tabelas de consulta é o seu crescimento exponencial com as dimensões do problema.

Recentemente, várias aproximações de CAD, baseadas em modelos através de redes neurais, têm sido introduzidas nas áreas de RF e microondas. Esta ferramenta

numérica tem possibilitado a modelagem eficiente de dispositivos e circuitos de microondas. Estes modelos são muito mais simples, enquanto mantêm a precisão das equações físicas e/ou empíricas dos modelos eletromagnéticos, Silva et al (1999).

Do ponto de vista numérico, as ANNs são ferramentas computacionais, que, aprendem através da experiência (treinamento) e generalizam exemplos previamente apresentados. Assim, são capazes de extrair as características essenciais dos exemplos de treinamento, que sejam representativos para um dado problema, Patnaik e Mishra (2000).

Neste capítulo, descreve-se alguns modelos desenvolvidos com base em redes neurais para dispositivos de RF/microondas. O método convencional (EM-ANN) é empregado no desenvolvimento dos modelos, Fig. 2.1. Os exemplos de aplicação ilustram aspectos desta técnica de modelagem, tais como: geração de dados de treinamento; seleção da configuração da ANN (número de entradas, neurônios ocultos e saídas); convergência de algoritmos de treinamento; teste e validação de modelos EM-ANN.

Quando um modelo EM-ANN for treinado e testado, a próxima etapa é a sua integração, por exemplo, em simuladores de circuitos. Os modelos EM-ANN apresentam duas vantagens principais: sua precisão é comparável a dos modelos EMs; o esforço computacional requerido é menor que o de um modelo EM. Por estes motivos, são apropriados para o projeto auxiliado por computador de dispositivos de RF/microondas. O método EM-ANN foi aplicado aos seguintes casos:

- Filtros distribuídos em linhas de microfita;
- Antena cilíndrica isolada.;
- Fita condutora;
- Indutores planares;
- Transistores de efeito de campo;
- Linhas de microfita.
- Substrato PBG como filtro em linhas de microfita
- Terminações padrões para calibração de analisadores de rede

Nas seções seguintes serão apresentados alguns exemplos de aplicação dos modelos EM-ANN. Descreve-se de forma concisa os modelos EMs envolvidos na geração dos dados de treinamento.

3.2 FILTROS DISTRIBUÍDOS EM MICROFITA

Em filtros distribuídos construídos em linhas de microfita, as seções de linha de microfita comportam-se como os elementos reativos do filtro (indutores e capacitores). Os principais parâmetros envolvidos no projeto destes filtros são: o número de seções, N_s ; a largura, w ; o comprimento, L , de cada seção de microfita; a impedância de operação, Z_o ; a frequência de corte f_c ; a espessura da fita condutora, t_E ; o material, μ_r , ϵ_r ; a altura do substrato, h_s .

3.2.1 Rede MLP para Projeto de Filtros Distribuídos

A aproximação CAD original para projeto/otimização de filtros distribuídos, baseada em modelos empíricos Fusco (1987), Gupta et al (1981), foi implementada. Os resultados obtidos foram empregados no treinamento de uma rede MLP (com uma entrada, três camadas ocultas com oito neurônios cada, sete neurônios na camada de saída).

O conjunto de treinamento foi gerado a partir de projetos otimizados de 21 filtros com 7 seções cada, na região de interesse, $f_c = [0,5 - 2,5]$ GHz, em intervalos $\Delta f_c = 0,1$ GHz. A Fig. 3.1 apresenta os resultados para dois projetos, nos extremos desta faixa de frequências, para operação em circuitos com $Z_o = 50 \Omega$. As seções de linha de microfita são construídas sobre um substrato de fibra de vidro, com as seguintes características: ($t = 0,001$, $h = 1,5$) mm, $\epsilon_r = 4,8$ e $\mu_r = 1$.

Estes projetos foram realizados visando a minimização das áreas de construção dos filtros, com a seguinte restrição: larguras de seções de linha constantes para todos os projetos. Desta forma, a frequência de corte (entrada do MLP) e os comprimentos de cada seção de microfita (saídas do MLP) são os únicos parâmetros do projeto otimizado, sendo usados no aprendizado da rede MLP através do algoritmo BP.

Na Fig. 3.2, tem-se o conjunto de exemplos de treinamento, para filtros distribuídos passa-baixa com característica de Butterworth. O aprendizado de uma rede neural, durante o processo de treinamento pela correção do erro, pode ser visualizado através do gráfico de evolução do erro médio quadrático, (2.11). Para a rede MLP implementada, este gráfico é mostrado na Fig. 3.3. Neste caso, o algoritmo BP desenvolvido, convergiu após 4403 épocas. Na execução deste algoritmo, escrito em Matlab™ (1998), o tempo de treinamento foi de 31,2 min.



Seção	Zo [ohm]	w [mm]	L [mm]
1	50	2.687	6.847
2	18.78	10.7	10.21
3	81.98	1	61.39
4	18.78	10.7	37.06
5	81.98	1	61.39
6	18.78	10.7	10.21
7	50	2.687	6.847



Seção	Zo [ohm]	w [mm]	L [mm]
1	50	2.687	1.369
2	18.78	10.7	2.043
3	81.98	1	12.28
4	18.78	10.7	7.411
5	81.98	1	12.28
6	18.78	10.7	2.043
7	50	2.687	1.369

Fig. 3.1 Resultados de projetos de filtros distribuídos para $f_c = 0,5$ GHz e $f_c = 2,5$ GHz.

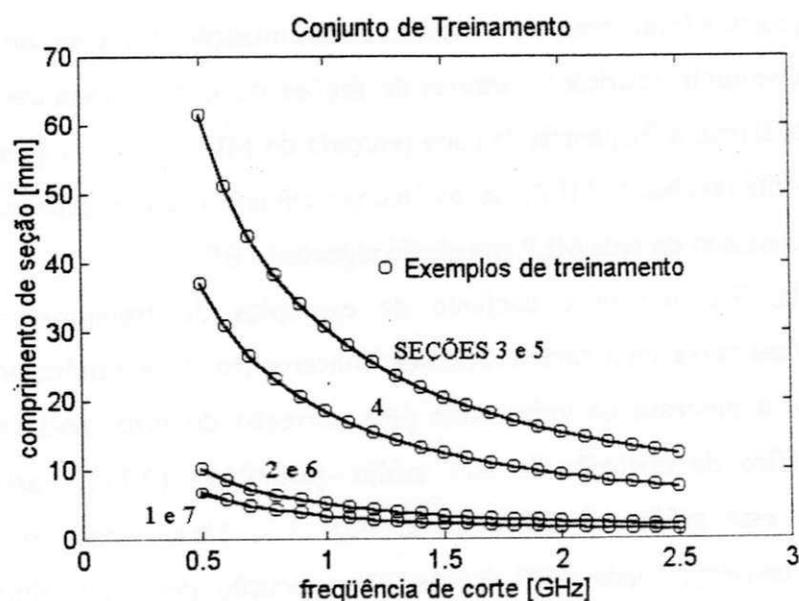


Fig. 3.2 Conjunto de treinamento para a rede MLP.

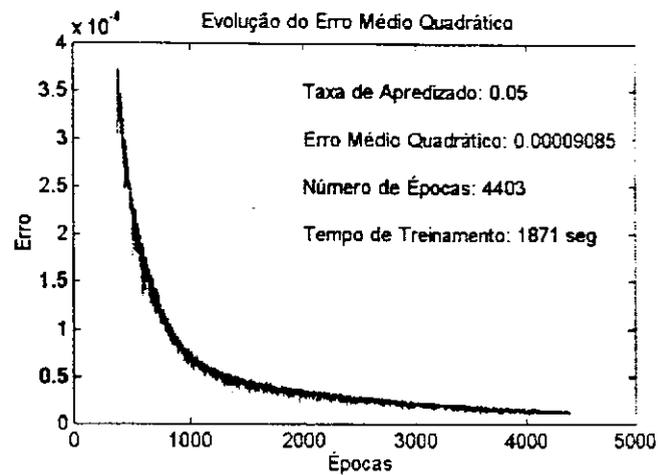


Fig. 3.3 Evolução do aprendizado da rede MLP.

Após o treinamento, a rede MLP foi submetida a um conjunto de teste, para o qual, verificou-se a sua capacidade de generalização, Fig. 3.4. Considerando-se esta resposta satisfatória, a rede MLP foi validada. Neste caso, a rede pode substituir os modelos empíricos para o projeto dos filtros distribuídos, dentro da região de interesse. O conjunto de teste foi gerado na faixa $f_c = [0,53 - 2,43]$ GHz, com passo $\Delta f_c = 0,1$ GHz.

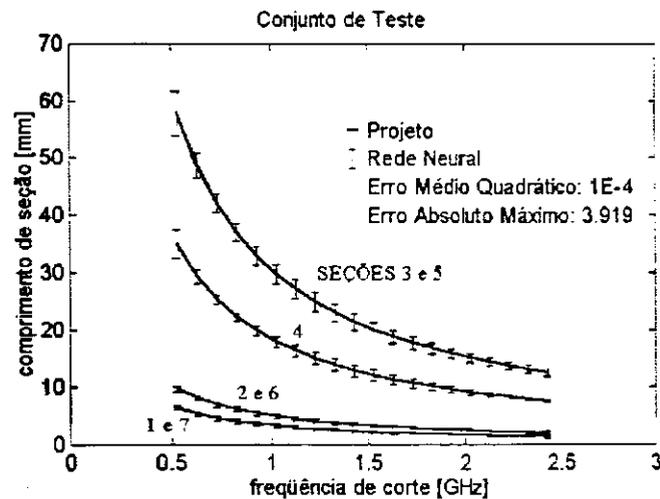


Fig. 3.4 Conjunto de teste e validação da rede MLP (projeto).

3.2.2 Rede MLP para Análise de Filtros Distribuídos

Tendo concluído as etapas de projeto e otimização, o projetista dispõe dos parâmetros geométricos e de operação de um dado filtro distribuído. Contudo, para proceder com a etapa de construção, uma análise precisa deve ser realizada, garantindo que, as especificações de desempenho sejam atendidas pelo filtro projetado.

Para a análise das descontinuidades das seções de microfita dos filtros projetados, utilizou-se o método da conservação da potência complexa, Neto et al (1997). Através deste método, obtém-se os parâmetros de espalhamento dos filtros distribuídos em linhas de microfita. Os resultados deste método foram usados para o desenvolvimento do MLP (com duas entradas, cinco camadas ocultas com oito neurônios cada, uma de saída).

Os dados de treinamento da rede MLP foram obtidos através da análise dos 21 filtros projetados anteriormente. O parâmetro de espalhamento S_{11} (coeficiente de reflexão) foi utilizado para a análise destes filtros, Fig. 3.5. Estas curvas possuem 50 exemplos por filtro, tal que, o conjunto de treinamento completo possui 1050 exemplos. Após a fase de treinamento, um conjunto de teste foi utilizado para validar a rede MLP. A resposta da rede para três filtros deste conjunto de teste é apresentada na Fig. 3.6.

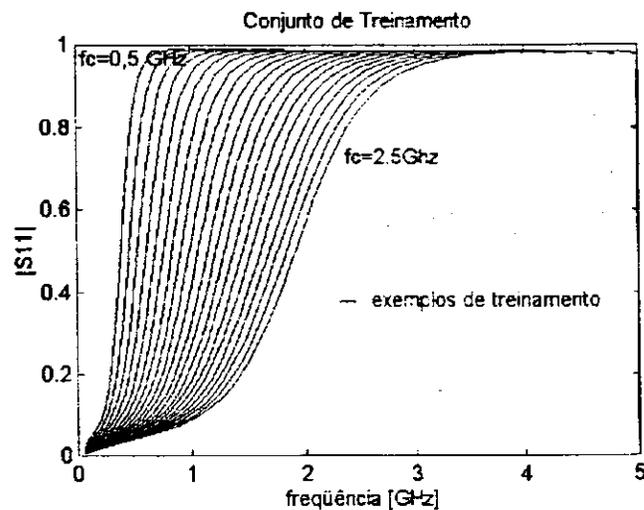


Fig. 3.5 Conjunto de treinamento para a rede MLP.

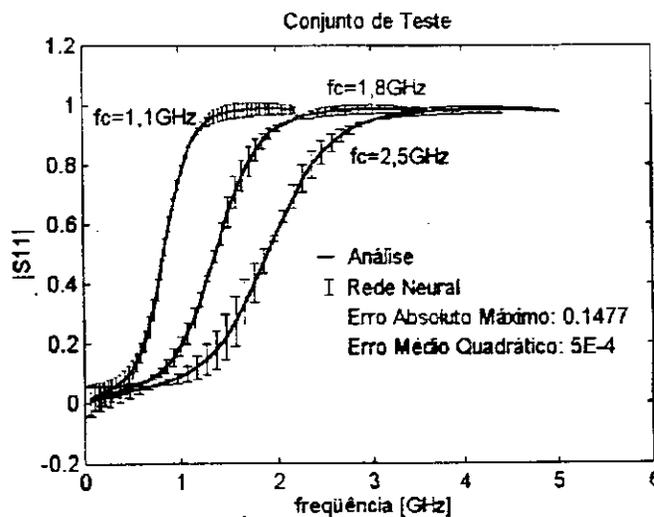


Fig. 3.6 Conjunto de teste e validação da rede MLP (análise).

Uma vez que as redes MLPs sejam validadas, uma nova aproximação pode ser desenvolvida para CAD de filtros distribuídos em linhas de microfita. Desta forma, os modelos MLPs podem ser usados nas etapas de projeto e análise, Fig. 3.7.

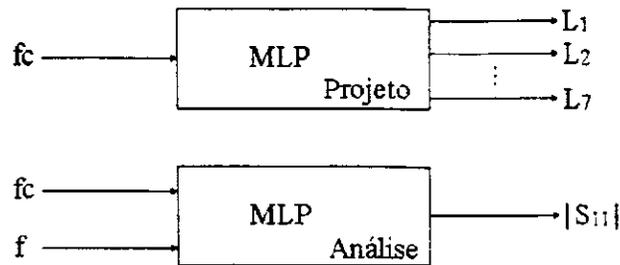


Fig. 3.7 Modelos de filtros distribuídos através de redes MLPs.

3.3 IMPEDÂNCIA DE ANTENAS ISOLADAS

As antenas isoladas, simétricas e assimétricas, são úteis no aquecimento localizado por deposição controlada de energia de microondas. Sua aplicação abrange inúmeros ramos da indústria, tais como: alimentícia, médica, em telemetria, comunicações subterrâneas, etc.

Uma antena isolada é construída com a própria linha coaxial de alimentação, Fig. 3.8, onde os condutores externos da linha coaxial são seccionados no ponto de junção da antena, e, os condutores interno e externo são eletricamente conectados na terminação da linha coaxial, Silva et al (1998).

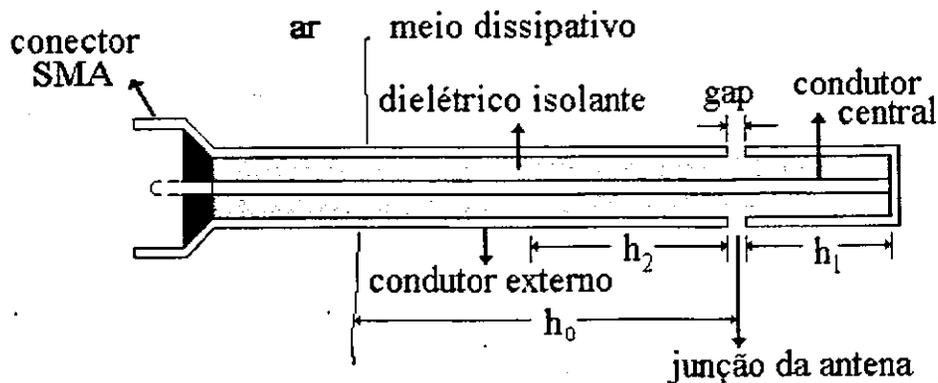


Fig. 3.8 Construção da antena isolada.

A antena isolada consiste de dois condutores centrais (região 1) envolvidos por uma casca cilíndrica dielétrica, consistindo de uma ou duas camadas (regiões 2 e 3), com

raios externos b e c , respectivamente. Fora da casca isolante está o meio dissipativo (região 4), Fig. 3.9.

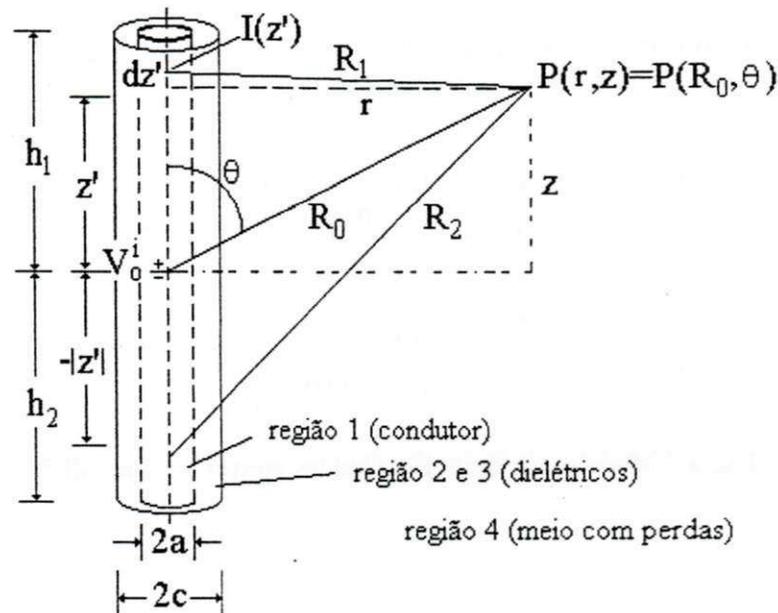


Fig. 3.9 Antena isolada em um meio dielétrico dissipativo.

Para encontrar a distribuição de corrente de uma antena cilíndrica, equações integrais derivadas das equações de Maxwell precisam ser solucionadas. Duas, das mais usuais, destas equações integrais são as equações de Hallén (3.1) e de Pocklington (3.2), King et al (1983). Atualmente, suas soluções são obtidas utilizando-se o método dos momentos, Da Mata (1993), Silva (1997), Silva et al (1998).

$$\int_{-h/2}^{h/2} I_z(z') \frac{e^{-jkR}}{4\pi R} dz' = -\frac{j}{2} YV_0^i \text{senk}|z| + C \cos kz \quad (3.1)$$

$$\left(k^2 + \frac{\partial^2}{\partial z^2} \right) \int_{-h/2}^{h/2} I_z(z') \frac{e^{-jkR}}{4\pi R} dz' = -j\omega\mu\epsilon E_z^i(z) \quad (3.2)$$

Contudo, em problemas de otimização e na análise do aquecimento localizado através de antenas isoladas, onde as características elétricas do meio variam com a temperatura, computações repetidas do método dos momentos tornam seu uso proibitivo. Neste caso, um modelo de rede neural pode ser utilizado em substituição a este modelo EM.

Como exemplo desta aplicação, uma rede MLP foi treinada para modelar a impedância de entrada de uma antena isolada simétrica, $Z_o = R_o - jX_o \Omega$, quando os comprimentos dos seus braços variam na região de interesse $h \in [1; 24]$ cm. A rede é definida com uma entrada (a frequência), vinte neurônios ocultos e duas saídas $[R_o; X_o]$. Os seguintes parâmetros do meio e da antena isolada são considerados:

- Frequência de operação: $f_o = 915$ MHz
- Braços da antena: $h_1 = h_2 = h = [1-24]$ cm
- Raio do condutor: $a = 0,47$ mm
- Raio dielétrico da região 2: $b = 0,584$ mm
- Raio dielétrico da região 3: $c = 0,8$ mm
- Permissividade da região 2: $\epsilon_{r2} = 1,0$
- Permissividade da região 3: $\epsilon_{r3} = 1,78$
- Permissividade da região 4: $\epsilon_{r4} = 42,5$
- Condutividade da região 4: $\sigma_4 = 0,88$ S/m

O conjunto de 49 exemplos de treinamento para o aprendizado da rede MLP é apresentado na Fig. 3.10.

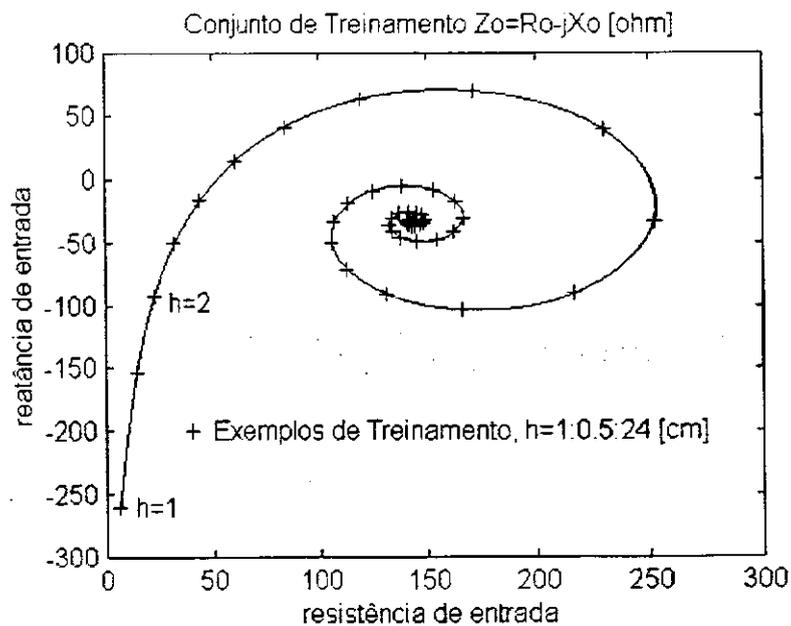


Fig. 3.10 Conjunto de treinamento para a rede MLP.

A evolução do processo de aprendizado da rede MLP é apresentado na Fig. 3.11. Através da apresentação de exemplos de teste, a capacidade de generalização desta rede é

verificada, Figs. 3.12, 3.13. Neste exemplo, funções disponíveis no ambiente do MatlabTM *Neural Network Toolbox* (1998), foram utilizadas para a implementação das redes MLPs.

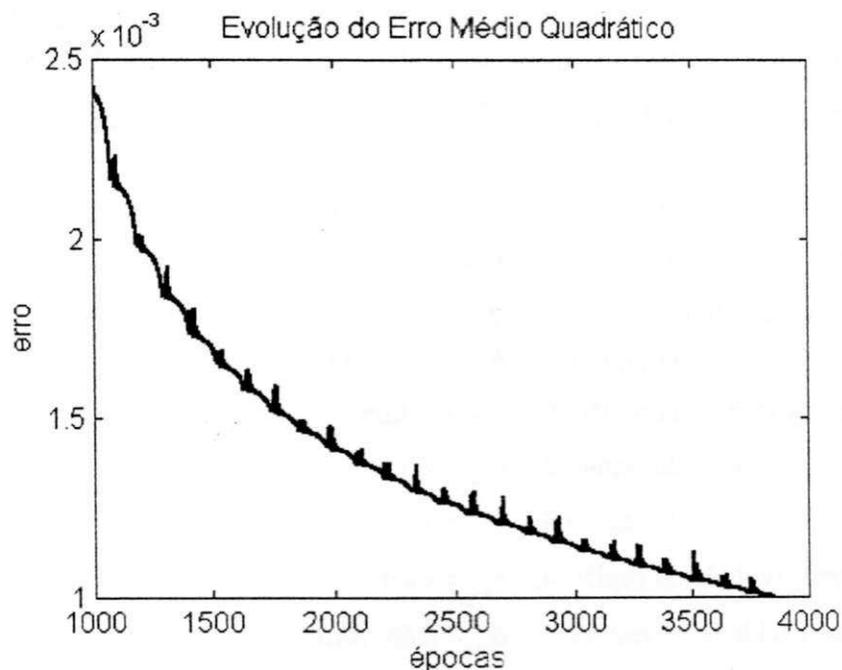


Fig. 3.11 Processo de aprendizado da rede MLP.

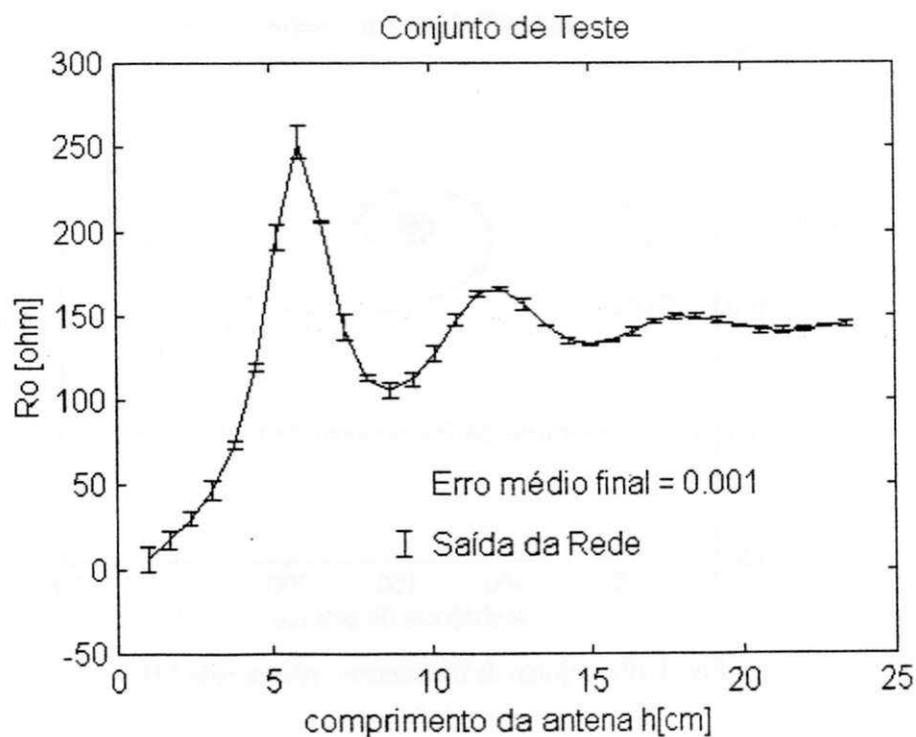


Fig. 3.12 Saída da rede para o conjunto de teste R_o .

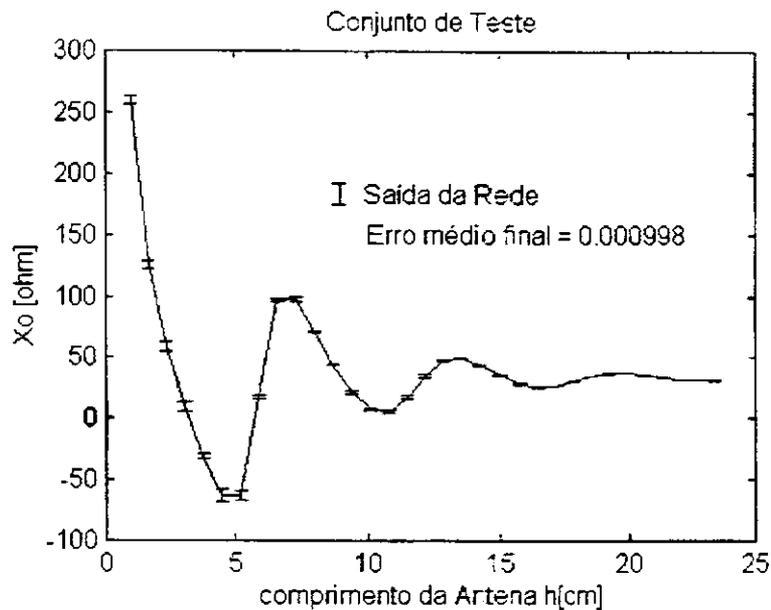


Fig. 3.13 Saída da rede para o conjunto de teste X_o .

3.4 ANNs PARA DISPOSITIVOS PLANARES

No projeto de um circuito integrado monolítico de microondas (MMIC), dispositivos planares como capacitores interdigitais, indutores espirais e FETs, entre outros, podem ser construídos sob uma mesma pastilha semicondutora. Pucel (1981), Pettenpaul et al (1988). Algumas características comuns dos MMICs são: baixas perdas, tamanho e peso reduzidos, baixo custo. Em seguida, trata-se do uso de modelos EM-ANN para dois parâmetros importantes no projeto de indutores espirais: as perdas ôhmicas e a indutância total.

3.4.1 Resistência AC de uma Fita Metálica

Elementos passivos discretos em MMICs são construídos com seções de linhas de microfita, com largura, w , da mesma ordem de grandeza da espessura de metalização, t_E . Com isto, o cálculo aproximado das perdas. Pucel et al (1968), normalmente resultam em fatores de qualidade elevados se comparados aos medidos. O cálculo exato da variação (com relação à frequência) da resistência elétrica de uma fita metálica, Fig. 3.14, é uma tarefa bastante difícil, sendo feita apenas por meio de sofisticados métodos numéricos, Waldow e Wolff (1987). Para uma aplicação em programas de CAD, medidas publicadas, Haefner (1937), têm sido interpoladas através da fórmula:

$$Rac = \frac{L}{\sigma w t_E} \left\{ \frac{0,43093 x_w}{1 + 0,041 (w/t_E)^{1,19}} + \frac{1,1147 + 1,2868 x_w}{1,2296 + 1,287 x_w^3} + 0,035 (w/t_E - 1)^{1,8} \right\} \quad (3.3)$$

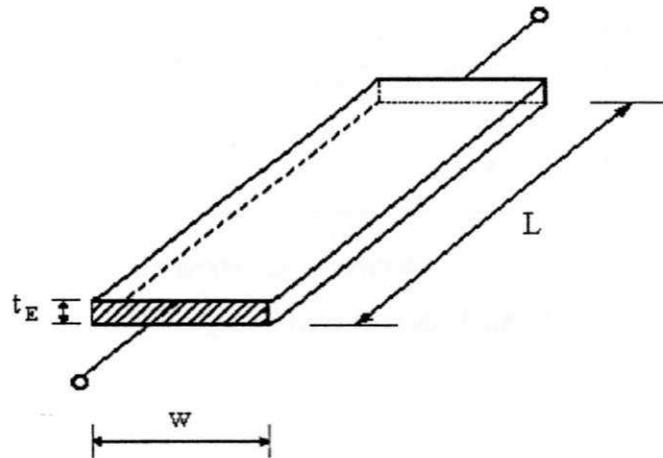


Figura 3.14 Fita metálica.

em que,

$$x_w = (2 f \sigma \mu w t_E)^{1/2} \quad (3.4)$$

é a frequência normalizada, L é o comprimento, σ é a condutividade elétrica e, μ , a permeabilidade magnética do condutor.

Uma rede MLP com 17 neurônios ocultos foi empregada como um interpolador alternativo à expressão (3.3). A rede possui duas entradas: $x_w \in [0,5; 12]$ e $w/t_E \in [1; 18]$. A sua saída é uma aproximação para a relação Rac/Rdc , em que $Rdc = L/\sigma w t_E$. A aproximação através de (3.3) é ilustrada na Fig. 3.15, e, através do modelo MLP, Fig. 3.16. Uma comparação é feita entre os valores calculados e medidos.

A interpolação através de (3.3) introduz erros em torno de 20% para $w/t_E = 18$. Pode-se verificar que a rede MLP apresenta maiores erros para os valores de $x_w < 3$, Fig. 3.16. Entretanto, para a aproximação dos valores medidos de Rac , o modelo MLP apresenta precisão superior àquela dada por (3.3).

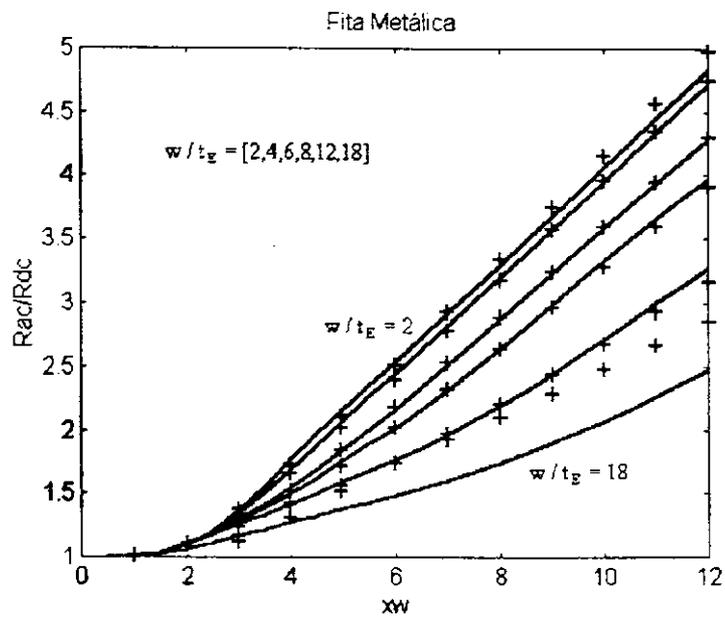


Fig. 3.15 Resistência ac de uma fita metálica. (-) Medidas, (+) Eq. (3.3).

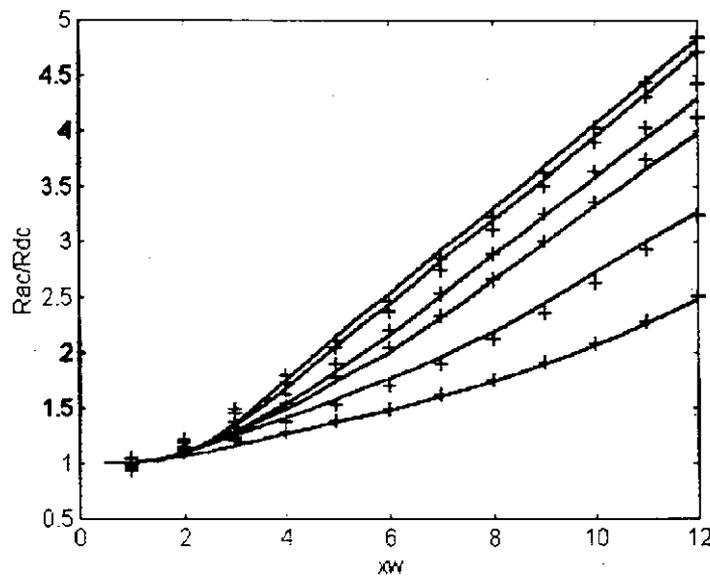


Fig. 3.16 Resistência ac de uma fita metálica. (-) Medidas, (+) Modelo MLP.

3.4.2 Indutores Espirais

Nesta seção, analisa-se o uso de redes neurais para modelar indutâncias. São considerados indutores espirais (retangulares e circulares), Fig. 3.17. Em função dos parâmetros geométricos dos indutores, o modelo ANN deve ser capaz de calcular sua indutância total.

Para o indutor espiral retangular, cada espira corresponde a quatro segmentos retos. Para o indutor circular, cada segmento eqüivale a uma espira. O valor da indutância total leva em conta as contribuições da auto-indutância de cada segmento, e, da indutância mútua entre cada par de segmentos paralelos. Os segmentos paralelos que carregam correntes em sentidos iguais (contrários), originam indutâncias mútuas positivas (negativas). O valor da indutância total é dado por:

$$L_T = \sum_{i=1}^{4\text{esp}} Li + M_+ - M_- \quad (3.5)$$

em que, Li é a auto-indutância de cada segmento, esp é o número de espiras. Para o indutor circular, pares de segmentos paralelos sempre carregam correntes de mesmo sentido. A indutância total é dada por:

$$L_T = \sum_{i=1}^{\text{esp}} Li + \sum_{i=1}^{\text{esp}-1} \sum_{j=i+1}^{\text{esp}} Mij \quad (3.6)$$

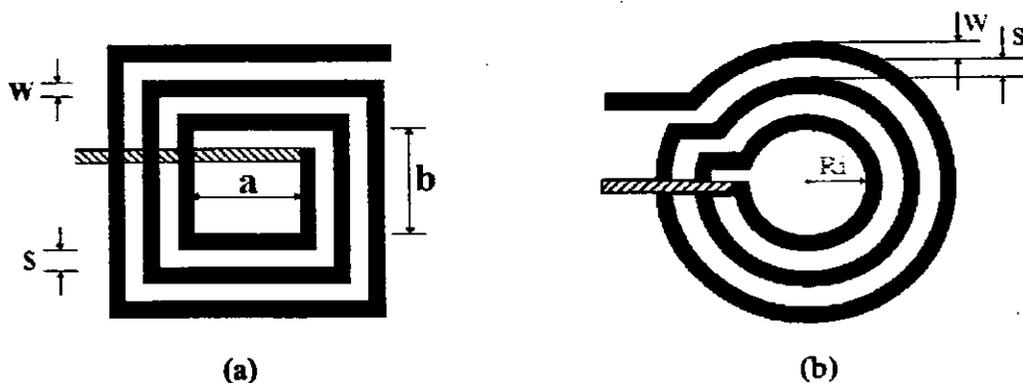


Fig. 3.17 Indutores espirais planares (a) Retangular. (b) Circular.

em que, Mij é a indutância mútua entre os segmentos circulares i e j . A intrincada computação de (3.4) é dada analiticamente, Greenhouse (1974), Chen (1995), enquanto (3.5) requer o cálculo numérico de integrais elípticas, Chen (1995), Pettenpaul et al (1988). Nos dois casos, o esforço computacional é uma função do número de espiras.

Para cada tipo de indutor, uma rede MLP foi treinada para mapear a indutância total, em termos dos parâmetros geométricos. Os seguintes valores foram usados no treinamento: – indutor retangular: $w = [5 : 5 : 20] \mu\text{m}$; $(a, b) = [50 : 10 : 100] \mu\text{m}$;

– Indutor circular: $Ri = [50 : 10 : 100] \mu\text{m}$.

Nos dois casos, os seguintes valores são assumidos: $t_E = 1,2 \mu\text{m}$, $s = 5,0 \mu\text{m}$ e $esp = [1:1:20]$ espiras. Assim, define-se os vetores de entradas dos modelos MLP: $x^T = [\theta, a, b, w, esp]$, indutor retangular; $x^T = [\theta, Ri, w, esp]$, indutor circular.

Por exemplo, para o indutor retangular, a combinação dos quatro parâmetros geométricos de entrada, para a região de interesse, resulta um conjunto de treinamento com 8000 exemplos. Na Fig. 3.18 está representa-se 10% deste conjunto, para $a = 10 \mu\text{m}$.

Durante a fase de treinamento, o número de unidades ocultas foi escolhido por tentativa e erro. Observando-se a convergência deste processo, selecionou-se um MLP com 30 unidades ocultas. A evolução do treinamento foi analisada para diferentes taxas de aprendizado. Alguns resultados são apresentados na Fig. 3.19.

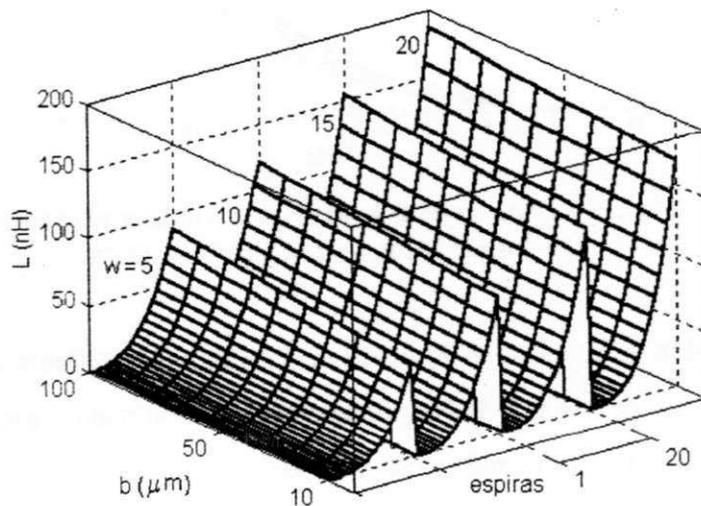


Fig. 3.18 Representação de 10% do conjunto de treinamento empregado no caso do indutor retangular. O gráfico reúne (lado a lado, com $esp=1, 2, \dots, 20$) os valores da indutância total.

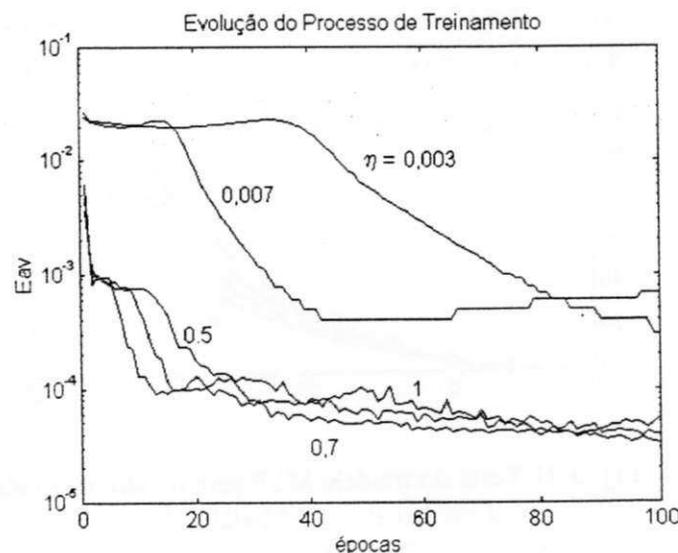


Fig. 3.19 Convergência da rede MLP usada para o indutor espiral retangular.

O procedimento acima, com os mesmos parâmetros para o modelo MLP, foi conduzido para o caso do indutor espiral circular, com resultados semelhantes. Após o treinamento, as redes foram testadas. Alguns resultados típicos de teste são apresentados nas Figs. 3.20, 3.21.

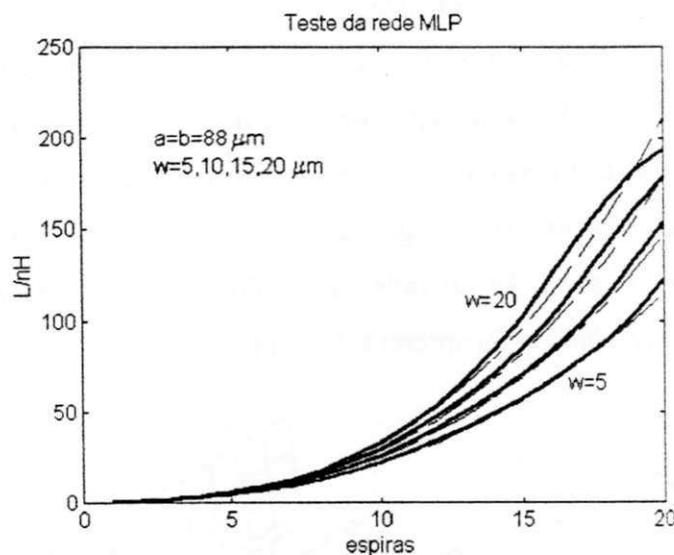


Fig. 3.20 Teste do modelo MLP para o indutor retangular:
(—) Rede MLP, (---) Modelo EM, (3.4).

Os resultados desta seção e da anterior, possibilita um cálculo eficiente do fator de qualidade via o uso de redes MLP. Após testadas e validadas, estas redes podem ser utilizadas no projeto e na otimização destes dispositivos.

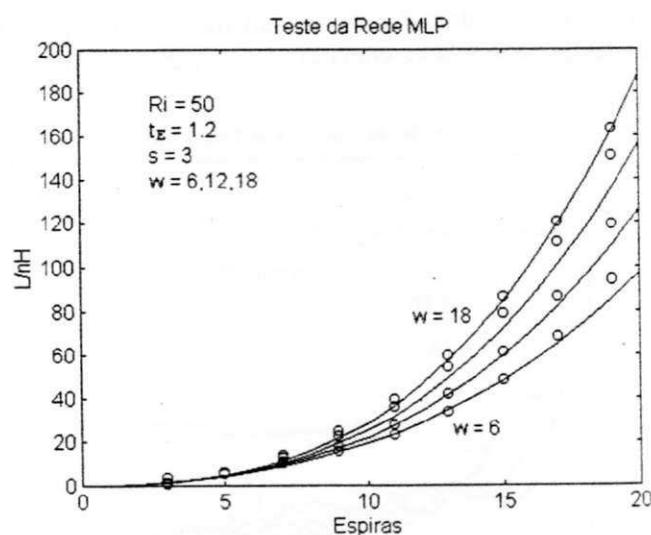


Fig. 3.21 Teste do modelo MLP para o indutor circular:
(o) Rede MLP, (---) Modelo EM, (3.5).

3.4.3 Transistores de Efeito de Campo

Neste exemplo do método EM-ANN, procurou-se modelar a característica de transferência DC da região ativa de FETs. Considera-se a região sob a porta de um MESFET tipo-n, Fig. 3.22, onde supõe-se apenas um tipo de portador (elétrons). O modelo EM neste caso, envolve a solução das equações de transporte de carga nesta região, Reiser (1973), Yamaguchi e Kodera (1979), Khatibzadeh e Trew (1988).

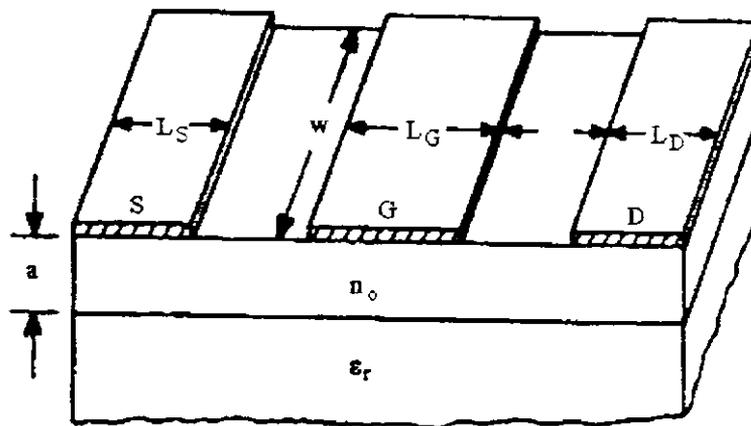


Fig. 3.22 Transistor de efeito de campo metal-semicondutor (MESFET).

$$\nabla^2 \psi = -\frac{q}{\epsilon} [N_D - n(x, y)] \quad (3.7)$$

$$q \frac{\partial n(x, y)}{\partial t} = \nabla \cdot \vec{J} \quad (3.8)$$

$$\vec{J} = -qn(x, y)\vec{v} + qD\nabla n(x, y) \quad (3.9)$$

$$\vec{J}_{tot} = \vec{J} + \epsilon \frac{\partial \vec{E}}{\partial t} \quad (3.10)$$

$$\vec{E} = -\nabla \psi \quad (3.11)$$

em que, ψ é o potencial elétrico; $n(x, y)$, a densidade de elétrons; \vec{E} , a intensidade de campo elétrico; ϵ , a permissividade elétrica; q , a carga elétrica; N_D , a concentração de impurezas doadoras; \vec{v} , a velocidade de deriva dos portadores; D , o coeficiente de difusão;

\bar{J} , a densidade de corrente de condução; \bar{J}_{tot} , a densidade de corrente total; t , o tempo. O modelo EM empregado neste estudo, segue o método de Yamaguchi e Kodera (1979), no qual supõe-se que a distribuição $n(x, y)$ seja conhecida.

Uma rede MLP (com uma camada oculta) e uma rede RBF foram implementadas, com 30 unidades ocultas cada, para modelar a característica $I_D(V_G, V_D)$ em função das faixas de parâmetros dados na Tabela 3.1. Supõe-se a configuração fonte comum, $V_S = 0$. A Fig. 3.23 ilustra a configuração do modelo neural, indicando-se suas entradas e saída. A validação das aproximações através dos modelos MLP e RBF pode ser verificada através das curvas de teste, sobrepostas à resposta do modelo EM, Fig. 3.24.

Tabela 3.1 Parâmetros de entrada das redes neurais.

PARÂMETROS	SÍMBOLO	FAIXA DE VALORES
Comprimento da Porta	L_G	0,5 – 1,5 μm
Largura da Porta	W	250 – 350 μm
Altura da Região Ativa	A	0,4 – 0,5 μm
Tensão de Porta	V_G	-1,0 – 0,0 V
Tensão de Dreno	V_D	0,0 – 4,0 V
Concentração de Impurezas	N_D	0,4 – 1,0 $\times 10^{22} \text{ m}^{-3}$

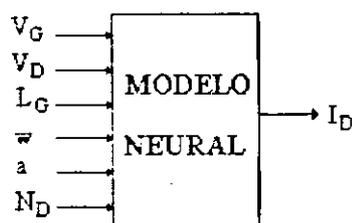


Fig. 3.23 Configuração do modelo neural para o FET.

As redes MLP e RBF foram implementadas em Matlab™ (1998) e C++. Neste exemplo, o programa escrito em linguagem C++ foi utilizado. A evolução do processo de treinamento das redes MLP e RBF, utilizadas para a modelagem da corrente de dreno do MESFET, é apresentada na Fig. 3.25. Em termos do erro médio quadrático, entre as redes MLP e RBF, esta última apresenta uma convergência melhor.

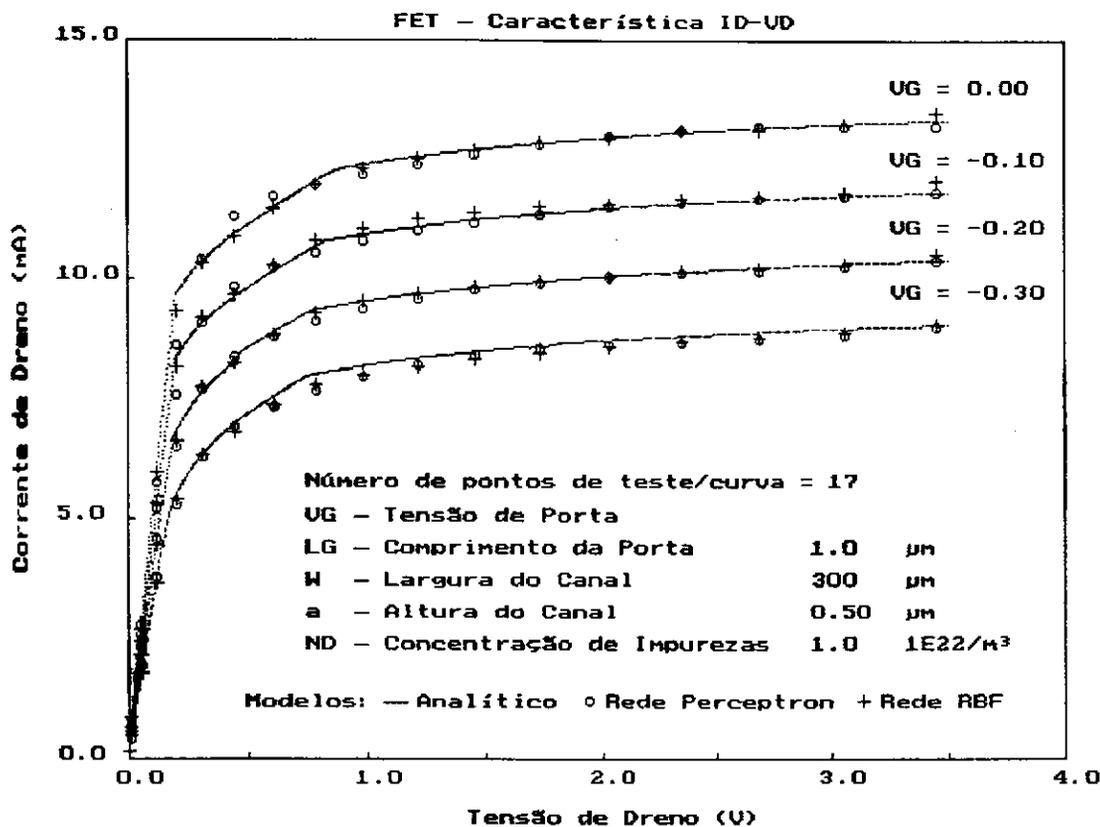


Fig. 3.24 Teste das redes MLP e RBF.

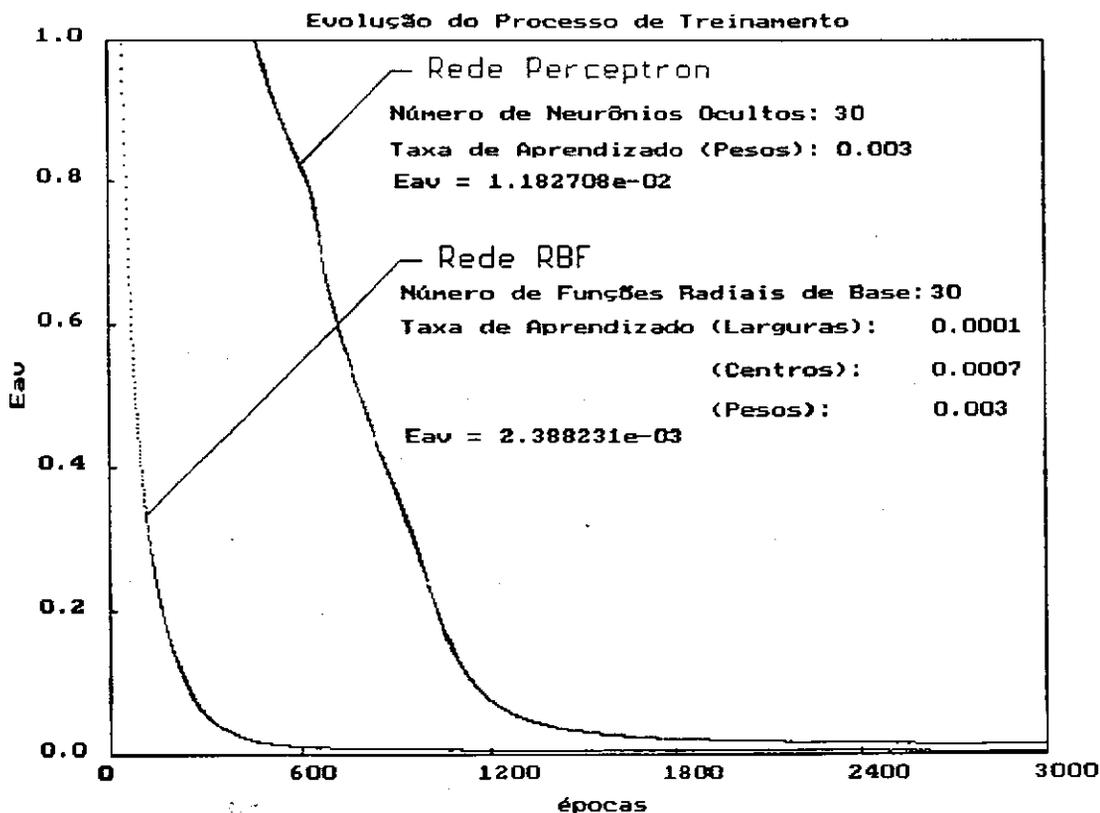


Fig. 3.25 Convergência das redes MLP e RBF.

3.5 Linhas de Microfita Suspensa e Invertida

Uma rede MLP treinada com o algoritmo BP para o cálculo da permissividade efetiva, ϵ_{eff} , de uma linha de microfita, foi proposta por Patnaik et al (1997). Aqui, considera-se os casos da linha de microfita suspensa e invertida.

Por exemplo, um MLP com quatro entradas $x' = [\theta \ w/h \ h_2/h \ \epsilon_{r2}]$, quinze neurônios ocultos e duas saídas $y' = [\epsilon_{eff} \ Z_o]$, em que, Z_o é a impedância característica da microfita invertida; $\theta = -1$. Para o treinamento desta rede MLP, define-se a região de interesse: $0.1 \leq w/h \leq 10$, em 14 pontos; $1 \leq h_2/h \leq 5$, em 5 pontos; $2 \leq \epsilon_r \leq 14$, em 7 pontos; totalizando 490 exemplos de treinamento.

Após a etapa de treinamento, testou-se a resposta do modelo MLP para microfitas com tipos diferentes de substratos, ϵ_{r2} , fora do conjunto de treinamento. Sua capacidade de generalização para exemplos novos pôde ser verificada, por exemplo, para substratos de GaAs e SiO_2 , Fig. 3.26.

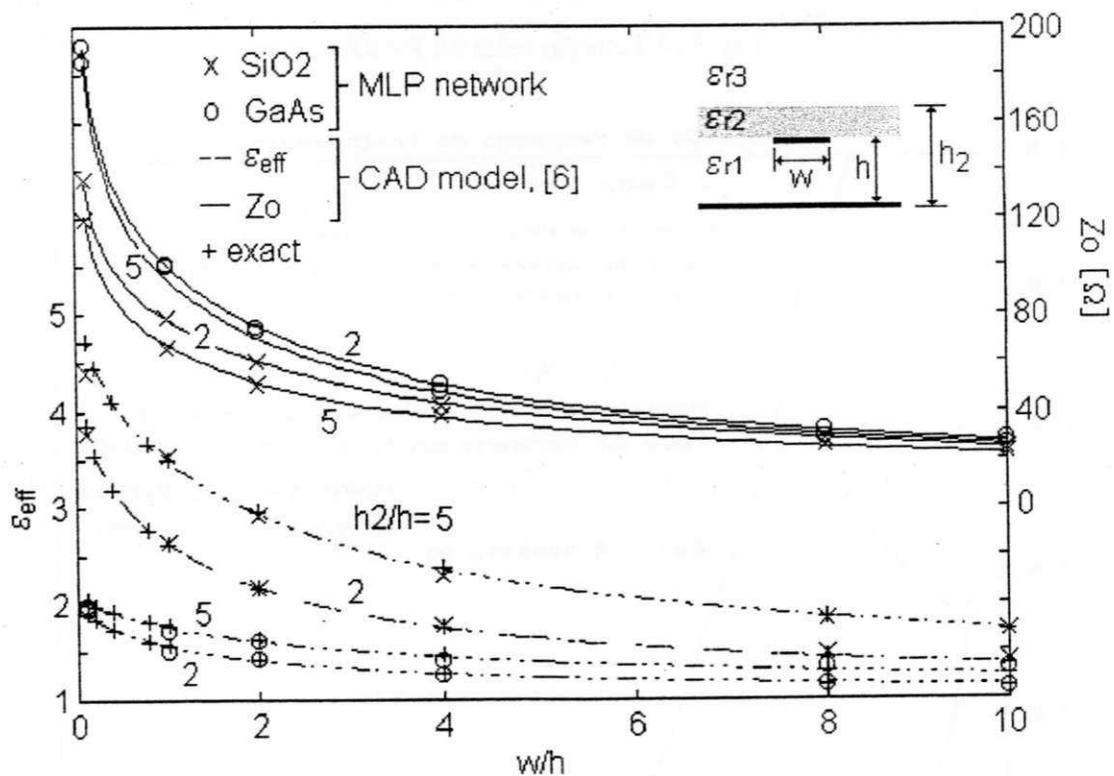


Fig. 3.26 Resultados para a microfita invertida $\epsilon_{r1} = \epsilon_{r3} = 1$,
(SiO_2 , $\epsilon_{r2} = 3.78$) e (GaAs, $\epsilon_{r2} = 12.9$).

3.6 SUBSTRATO PBG COMO FILTRO EM LINHAS DE MICROFITA

Na tarefa de modelar dispositivos e circuitos de RF/microondas, depara-se com determinados comportamentos não lineares, para os quais, as redes neurais MLPs fornecem uma aproximação apenas razoável. As limitações dos MLPs para aproximar as respostas de dispositivos de RF/microondas, por exemplo, através de parâmetros de espalhamento medidos, motivou em parte o desenvolvimento da rede de funções *sample*.

Nesta seção, aborda-se a modelagem através do método RFS de uma estrutura PBG como filtro em linha de microfita. Uma estrutura PBG é composta de redes periódicas infinitas, com diferentes dielétricos, que previnem a propagação em qualquer direção (duas para o caso da rede bidimensional). No entanto, o termo PBG é usado aqui para designar uma rede periódica finita, que introduz uma atenuação sobre a banda de rejeição.

Usando-se uma estrutura PBG como substrato (PBGS), os componentes do circuito são espalhados por várias célula unitárias sobre o PBGS, integrando o filtro com o circuito impresso. Qian et al (1997) e Radisic et al (1998) têm apresentado as características, através de medidas e simulação do método FDTD (*finite difference time domain*), de filtros rejeita banda em linhas de microfita, construídos sobre diferentes substratos PBG.

A rejeição de frequências em projetos em linhas de microfita pode ser conseguido através do emprego de tocos sintonizados em curto. Esta técnica geralmente é estreita em banda, e, ocupa uma área valiosa no projeto de um circuito em microfita. Uma alternativa para rejeitar frequências é o emprego de estruturas PBG, Fig. 3.27.

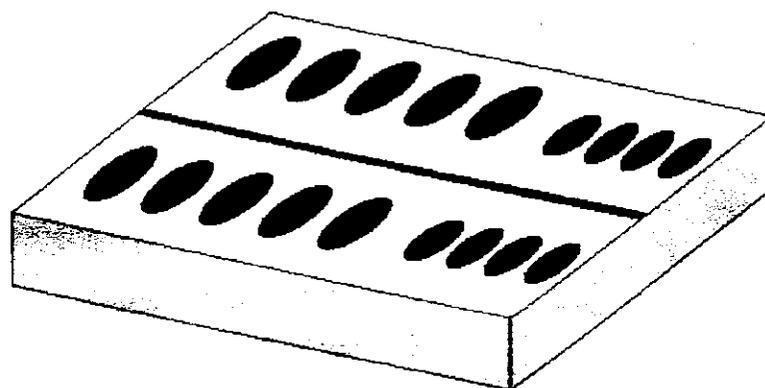


Fig. 3.27 Substrato PBG como filtro em linha de microfita.

No desenvolvimento do modelo RFS, considerou-se os resultados medidos para um filtro em linha de microfita construída sobre um PBGS, Rumsey et al (1998), com os seguintes parâmetros: permissividade relativa do substrato PBG, $\epsilon_{PBGS} = 10,0$, com espessura de 1,6 mm; espaçamento da rede de 18 mm, uma fração de volume de buracos de 0,35. Maiores detalhes são obtidos em Yang et al (1997), que utiliza a mesma estrutura, mas com buracos quadrados.

Comparações entre os resultados dos modelos RFS (desenvolvidos com diferentes resoluções), os resultados medidos e os simulados através do método FDTD, são apresentados na Fig. 3.28. Observa-se a excelente aproximação dos modelos RFS.

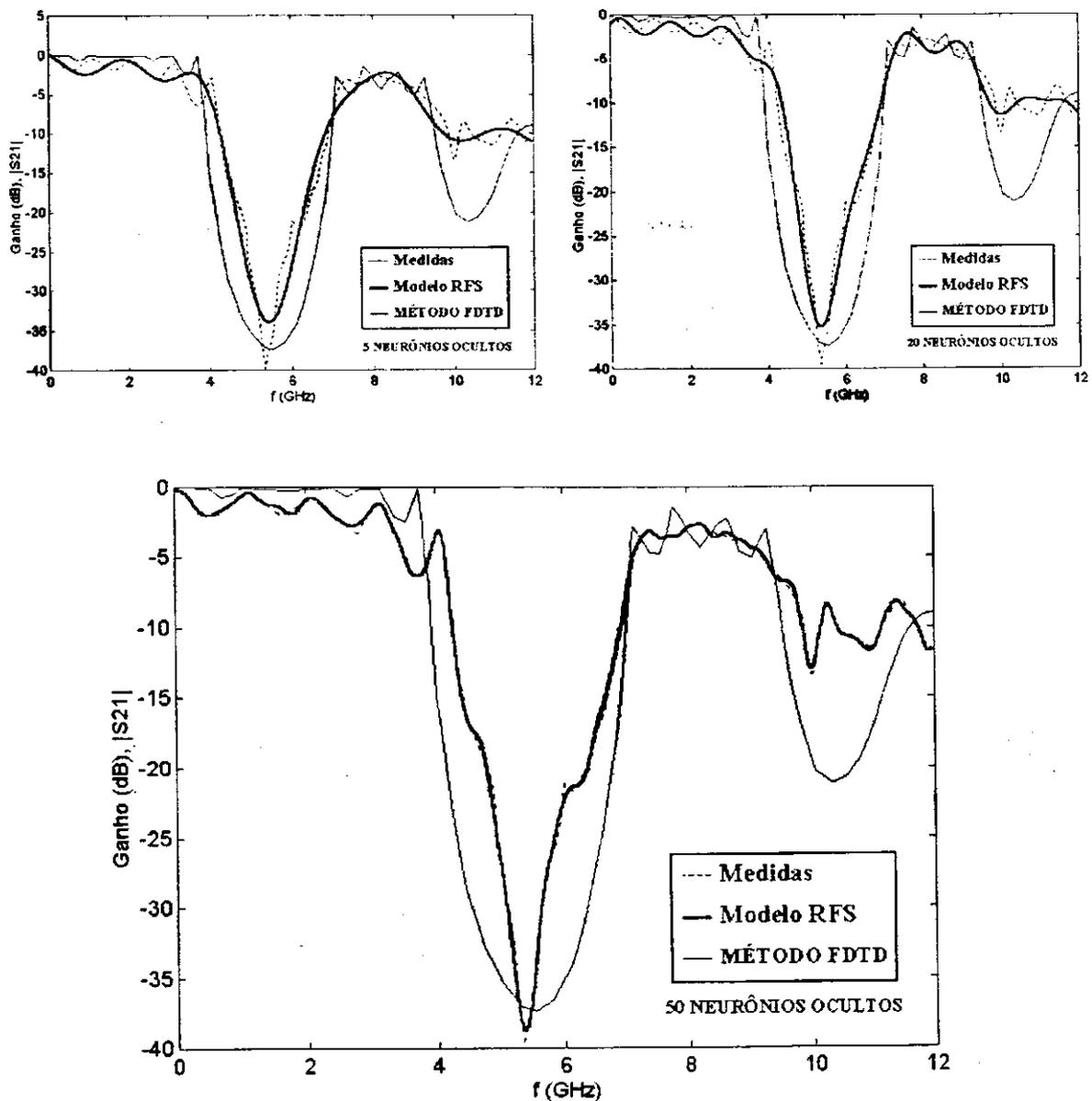


Fig. 3.28 Aproximação da resposta de uma estrutura PBG através de modelos RFS.

3.7 TERMINAÇÕES PARA CALIBRAÇÃO DE ANALISADORES DE REDES

A técnica TRL (thru-reflect-line) é bastante precisa para a calibração de analisadores vetoriais de redes (VNAs), em particular, sendo útil em ambientes sobre pastilha semicondutora. Através de medidas das dimensões das terminações (cargas) padrões (que simplesmente consistem de um número de linhas de transmissão de comprimentos variáveis), suas impedâncias características são determinadas. Para uma calibração precisa, torna-se necessário o uso de linhas longas, o que requer uma grande área sobre a pastilha semicondutora. Esta é uma desvantagem da técnica de calibração TRL. Além disso, para se obter comprimentos de linha diferentes, há a necessidade de posicionamento de pontas de prova durante o processo de calibração.

As técnicas OSLT (open-short-load-thru) e LRM (line-reflect-match), Rytting (1998), contornam estas limitações. Estes são módulos compactos para calibração, pois empregam linhas de transmissão curtas, preferíveis em aplicações sobre pastilha semicondutora. Estes módulos com elementos discretos reduzidos tendem a ser menos precisos, desde que, torna-se mais difícil o cálculo dos coeficientes de reflexão das cargas padrões. Contudo, se os módulos de calibração compactos podem ser caracterizados através de uma calibração de referência, tal como a técnica TRL, é possível realizar calibrações OSLT e LRM precisas de VNAs.

Uma vez que uma terminação padrão imperfeita é caracterizada, deve-se decidir entre usar os dados experimentais diretamente ou desenvolver um modelo para a terminação. Tem sido demonstrado que, a calibração de VNAs (através das técnicas OSLT e LRM) pode ser aperfeiçoada com a aplicação de redes MLPs, Jargon et al (2001), Jargon e Gupta (2001). A estes modelos foram atribuídas as seguintes vantagens: (1) não requerem modelos físicos detalhados; (2) apenas poucos exemplos de treinamento são necessários para modelar precisamente as terminações padrões, logo, o tempo de calibração é reduzido; (3) há uma redução da memória ocupada com relação a tabelas de consulta; (4) são menos susceptíveis a ruídos inerentes nos dados medidos.

Como um primeiro exemplo da aplicação do modelo através da RFS, considera-se as terminação padrões construídas de linhas de transmissão coplanares em guias de onda (CPW), Jargon e Gupta (2001). Estas foram fabricadas com condutores de ouro de $1,5 \mu\text{m}$ de espessura, sobre uma camada de $500 \mu\text{m}$ de GaAs, Williams et al (1990); o condutor

central tem $73 \mu\text{m}$ de largura, separado $49 \mu\text{m}$ do plano terra. A Fig. 3.29 apresenta as magnitudes do coeficiente de reflexão, medidas (em 192 frequências na faixa $0,5\text{--}40 \text{ GHz}$) apenas para uma carga e uma linha de $2,135 \text{ mm}$ de comprimento. Com o uso de apenas uma linha, explica-se a imprecisão em múltiplos de $26,65 \text{ GHz}$, onde a diferença entre os comprimentos das linhas correspondem a múltiplos de meio comprimento de onda. Jargon e Gupta (2001). Neste caso, a carga apresenta um desvio considerável de 50Ω , Fig. 3.29.

O modelo através da rede MLP (com uma camada oculta de cinco neurônios) proposto por Jargon e Gupta (2001) foi treinado com o programa *NeuroModeler* desenvolvido por Zhang et al (1999). A aproximação através deste modelo é apresentada na Fig. 2.29. Nota-se que, apesar de seguir a tendência dos dados medidos, o modelo MLP é impreciso em torno de $26,65 \text{ GHz}$, onde ocorre um pico na curva $|S_{11}|$.

A rede RFS foi empregada para modelar a carga descrita acima. A propriedade de escala deste modelo permite uma aproximação mais precisa desta carga padrão. Conforme ilustrado na Fig. 3.29, variando-se o número de neurônios ocultos e o parâmetro σ , obtêm-se aproximações em diferentes níveis de escala, tornando o modelo mais flexível para escolha de uma aproximação mais apropriada para a técnica de calibração LRM.

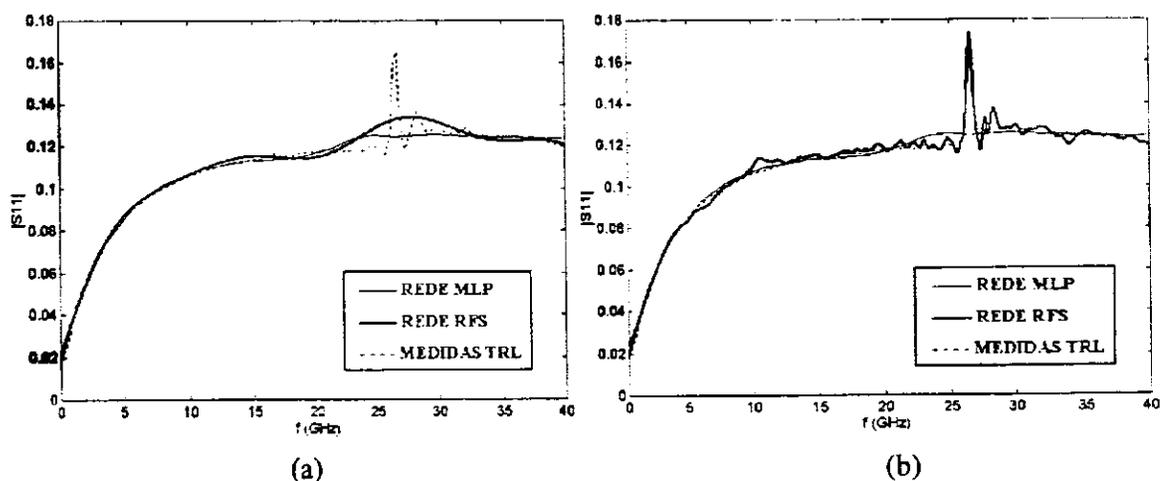


Fig. 3.29 Coeficiente de reflexão de uma carga padrão medida através da técnica TRL.

Comparação entre os modelos: MLP (Jargon e Gupta, 2001) e RFS.

(a) Modelo RFS com resolução baixa. (b) Modelo RFS com resolução alta.

A relação medida do coeficiente de reflexão com relação à frequência para a terminação padrão descrita, Fig. 3.29, é um mapeamento de difícil aproximação. Isto deve-se à presença nesta curva de regiões com não linearidades suaves e abruptas. A rede MLP

apenas segue a tendência dos resultados medidos, evitando o pico em 26,65 GHz. O mesmo ocorre para o modelo RFS com resolução baixa, Fig. 3.29(a).

O modelo RFS com resolução alta aproxima bem a oscilação abrupta na curva do coeficiente de reflexão. Contudo, a sobreposição das ondulações laterais apresentadas pelas funções *sample* com larguras reduzidas (usadas para resolução alta), provocam oscilações indesejáveis na região suave da resposta, Fig. 3.29(b).

Para contornar este problema, propomos restrições para os neurônios ocultos da RFS. Especificamente, os neurônios da RFS são classificados conforme sua resolução: alta ou baixa. Os neurônios destinados para resolução alta têm centros fixos cuidadosamente localizados, e, suas funções de ativação, (2.72), são modificadas para a seguinte forma:

$$y_h = \varphi(\mathbf{x}, \mathbf{t}_h) = \begin{cases} \text{sample}(\theta_h), & \text{se } \|\mathbf{x} - \mathbf{t}\| < \kappa \\ \frac{\text{sen}(\theta_h)}{\theta_h^\xi}, & \text{se } \|\mathbf{x} - \mathbf{t}\| \geq \kappa \end{cases} \quad (3.12)$$

em que, κ e ξ são constantes. Esta modificação é necessária para um neurônio destinado à resolução alta, pois, localiza sua influência na resposta da RFS apenas nas vizinhanças de seu centro, que é fixo. Portanto, a constante κ define a região de influência dos neurônios de resolução alta. Para as entradas, \mathbf{x} , distantes do centro de um destes neurônios, a inclusão da constante ξ , em (3.12), atenua as ondulações laterais das funções *sample* com larguras reduzidas.

Por exemplo, para aproximar as medidas apresentadas na Fig. 3.29, empregou-se uma RFS com 23 neurônios ocultos. Três neurônios são selecionados para resolução alta, com larguras iniciais $\sigma_0 = 200$ e centros fixos, dados por $\mathbf{t} = [24,80 \ 26,65 \ 28,40]$ GHz. Para os demais neurônios, utilizou-se $\sigma_0 = 25$ e os centros são selecionadas aleatoriamente das entradas apresentadas à RFS.

No desenvolvimento do modelo RFS para a terminação padrão considerada, empregou-se (3.12) com $\kappa = 0,05$ e $\xi = 1,432$. O resultado da aproximação é apresentado na Fig. 3.30. Verifica-se uma pior aproximação do pico em 26,65 GHz na curva $|S_{11}|$ com relação ao resultado da Fig. 3.29(b), contudo, as oscilações indesejáveis na aproximação são eliminadas. Além disso, quando emprega-se (3.12) e o conceito de resolução para um neurônio, variações pequenas também são aproximadas, Fig. 3.30.

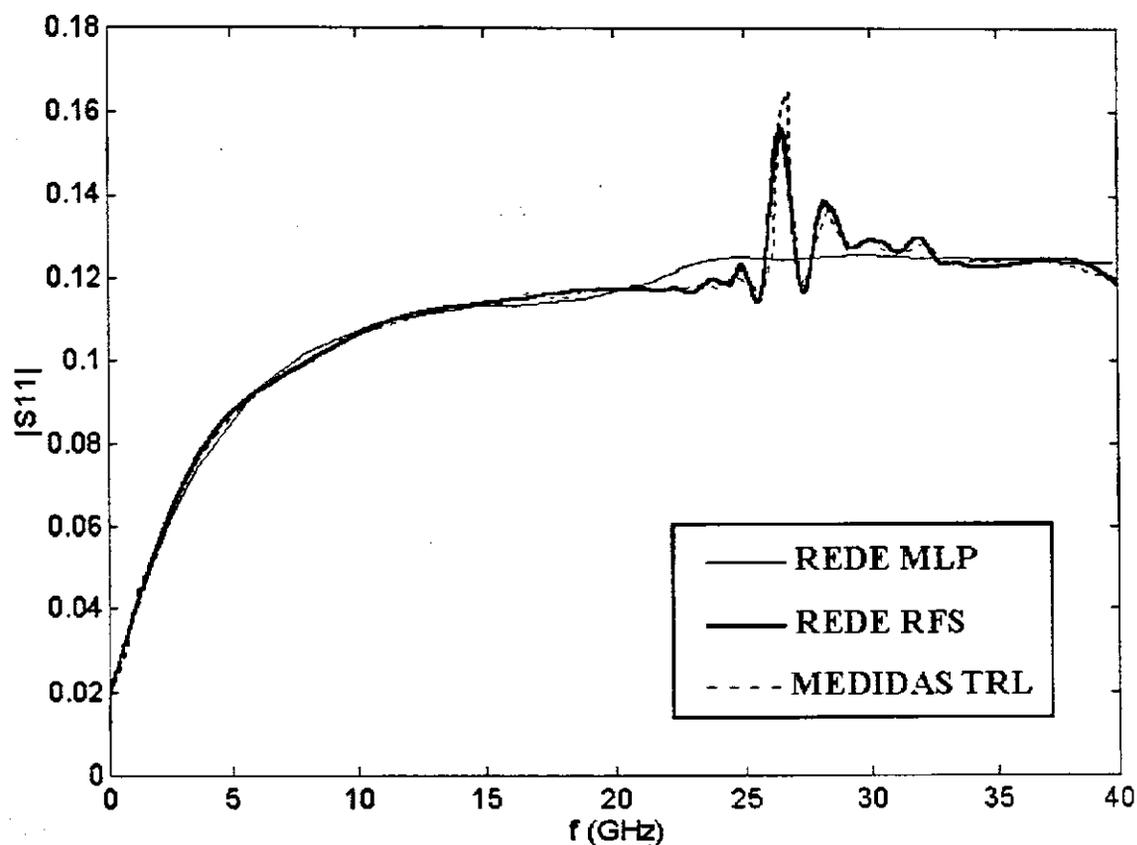


Fig. 3.30 Resultados dos modelos MLP e RFS (com três neurônios de resolução alta).

Como um segundo exemplo da aplicação do método RFS, considerou-se terminações padrões construídas de linhas de transmissão CPW, fabricadas com condutores de ouro de $4,5 \mu\text{m}$ de espessura, sobre uma camada de $625 \mu\text{m}$ de GaAs. As cargas usadas como terminações foram compostas de filme fino de material resistivo (TiWN), Williams et al (1990).

A Fig. 3.31 apresenta as magnitudes do coeficiente de reflexão, medidas (em 165 frequências na faixa 1,0–67 GHz) para uma terminação aberta. Por inspeção, pode-se comparar o modelo MLP, Jargon e Gupta (2001), e o modelo RFS com diferentes resoluções. É notório que o modelo RFS é bem mais preciso que o modelo MLP.

A Fig. 3.32 apresenta as magnitudes do coeficiente de reflexão, medidas (em 165 frequências na faixa 1,0–67 GHz) para uma terminação em curto. Por inspeção, pode-se comparar o modelo MLP, Jargon e Gupta (2001), e o modelo RFS com diferentes resoluções, constatando a maior precisão deste último.

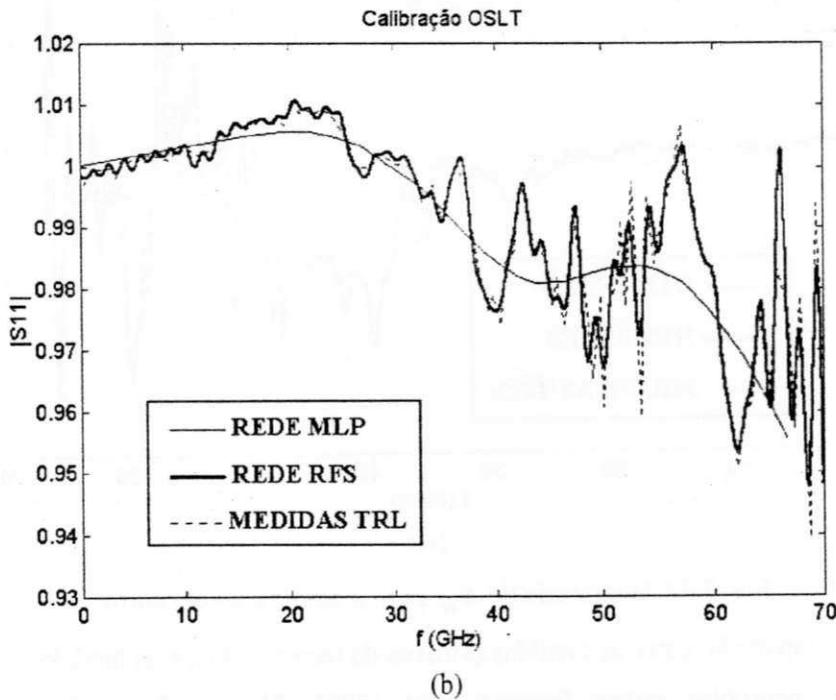
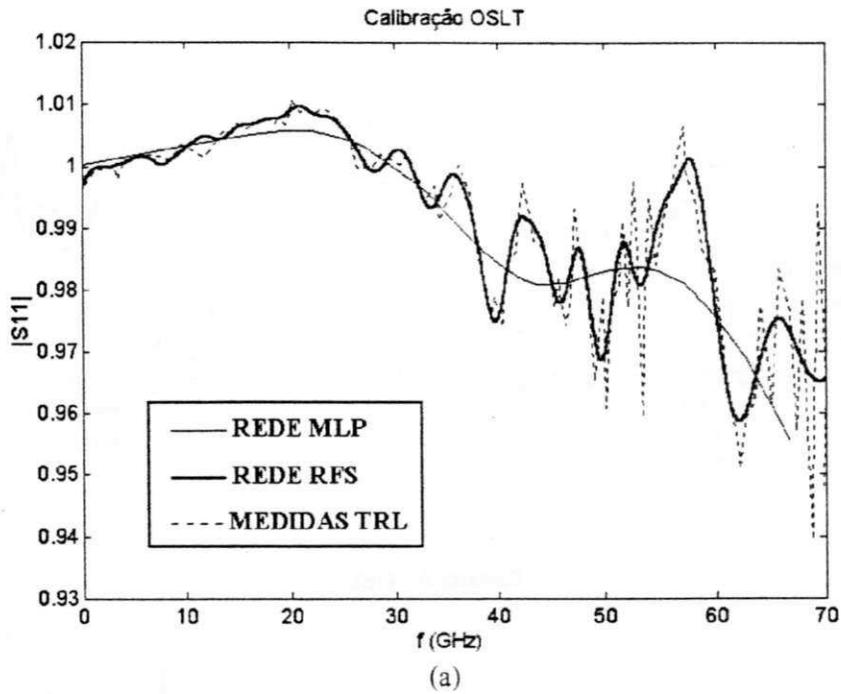


Fig. 3.31 Magnitude de S_{11} para a terminação em curto.

Comparação entre as medidas através da técnica TRL e os modelos:

MLP com 5 neurônios ocultos, Jargon e Gupta (2001); RFS com 50 neurônios ocultos.

(a) RFS com resolução baixa, $\sigma_0 = 10$. (b) RFS com resolução alta, $\sigma_0 = 200$.

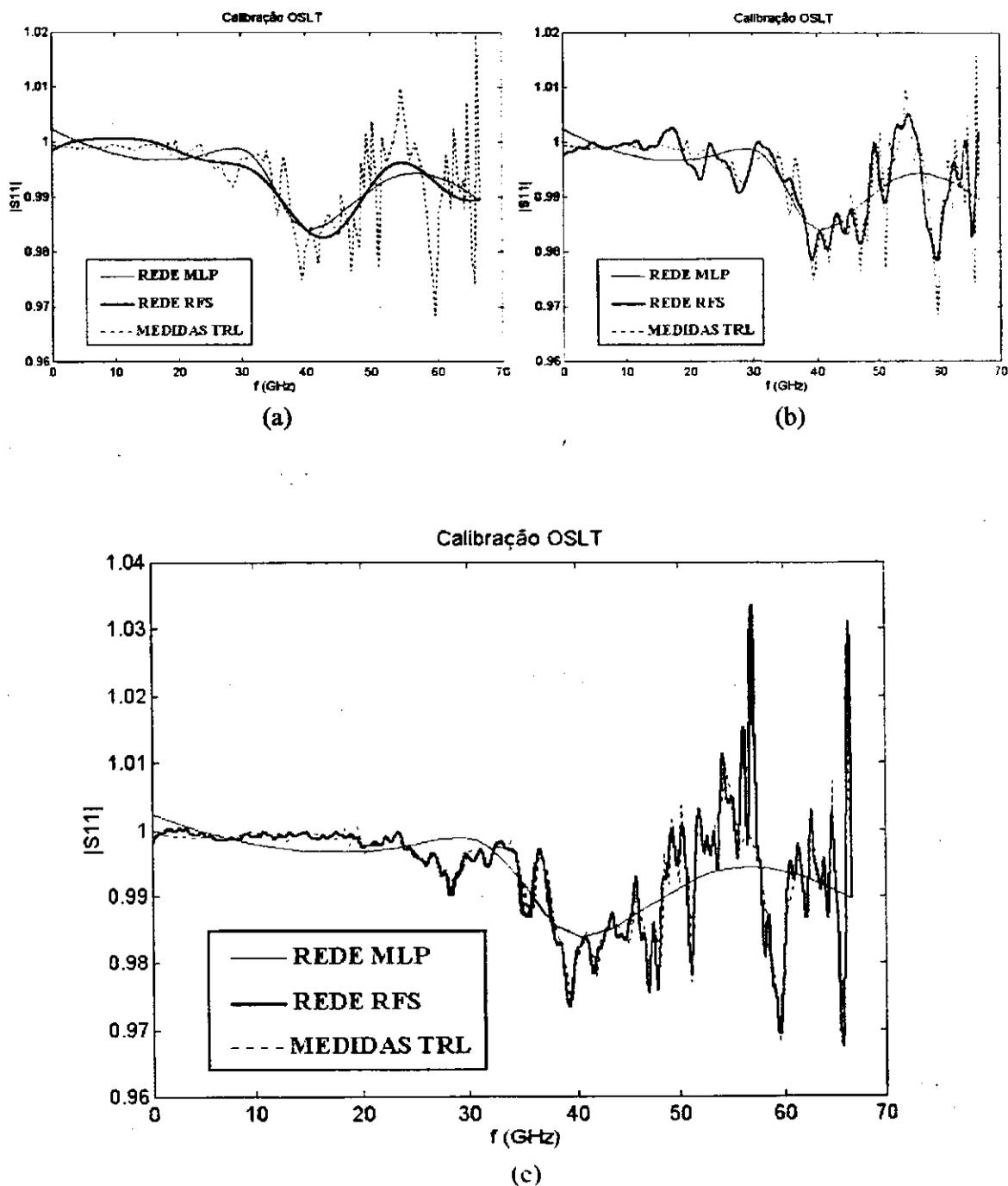


Fig. 3.32 Magnitude de S_{11} para a terminação em curto.

Comparação entre as medidas (através da técnica TRL) e os modelos:
MLP com 5 neurônios ocultos, Jargon e Gupta (2001); RFS com 50 neurônios ocultos.

(a) RFS com resolução baixa, 5 neurônios ocultos e $\sigma_0 = 2$.

(b) RFS com resolução média, $\sigma_0 = 20$.

(c) RFS com resolução alta, 50 neurônios ocultos e $\sigma_0 = 200$.

Capítulo 4

MODELOS DE GaAs MESFETs ATRAVÉS DE REDES MLPs PARA ANÁLISE TRANSITÓRIA DE CIRCUITOS NÃO LINEARES

4.1 INTRODUÇÃO

Nas aplicações reportadas na literatura, Zaabab et al (1994), Santos et al (1997), as ANNs têm sido utilizadas como modelos de dispositivos ativos não lineares, para a análise em regime de circuitos de microondas. Neste caso, a divisão do circuito não linear em sub-redes lineares e não lineares através do MBH, facilita a inclusão de modelos ANNs, para dispositivos ativos, como sub-redes não lineares do circuito. Neste capítulo, descreve-se uma proposta inédita desta técnica EM-ANN, voltada para a análise transitória não linear.

Simuladores de circuito como o SPICE (*simulation program with integrated circuit emphasis*), são construídos em torno de modelos para dispositivos semicondutores, tais como, diodos, BJTs, MOSFETs, e MESFETs, Raghuram (1989) e Pillage et al (1995).

Em geral, a precisão dos modelos empregados determina a qualidade da simulação. Por outro lado, modelos detalhados tornam o simulador mais lento. Portanto, deve haver um compromisso entre a precisão e a velocidade de simulação. Por exemplo, simuladores como o MOTIS (*MOS timing simulator*), Chawla (1995), usam modelos internos simplificados. Com o uso das ANNs é possível transpor grande parte deste problema.

No uso de um programa comercial para simulação de circuitos, geralmente o projetista encontra dificuldades para implementar seu próprio modelo não linear para um dispositivo novo, Pillage et al (1995). Com o uso de modelos através de ANNs, esta tarefa é bastante facilitada. A rede neural é vista como um modelo tipo caixa-preta. Não importa o modelo do dispositivo físico em questão, a capacidade de generalização, para regiões

onde pouco ou nenhum dado de treinamento existe, torna interessante o uso das redes neurais para aproximar mapeamentos complexos de entrada-saída de dispositivos novos, pouco ou não conhecidos.

Neste capítulo, as vantagens dos modelos através de ANNs, são verificadas. Na seção 4.2, os principais tópicos do método utilizado para a análise transitória não linear são detalhados. Na seção 4.3, descreve-se as etapas de desenvolvimento de modelos através de MLPs, numericamente eficientes e precisos, para os dispositivos GaAs MESFETs. Na seção 4.4, demonstra-se a validação do modelo MLP desenvolvido, através da simulação da resposta transitória de um circuito porta lógica, operando na faixa de Gigabit/s. O sucesso obtido nas simulações do modelo MLP desenvolvido, motivaram as aplicações de análise de sensibilidade do circuito porta lógica, seção 4.5.

4.2 ANÁLISE TRANSITÓRIA NÃO LINEAR

O método desenvolvido para o cálculo da resposta transitória de um circuito não linear (para grandes sinais) é o mesmo empregado em simuladores comerciais como o SPICE. A seguir, serão abordados os principais tópicos relacionados a este tipo de solução no domínio do tempo.

4.2.1 Análise DC Não Linear

A análise DC de qualquer circuito não linear leva à um sistema de equações algébricas não lineares. Para obter-se uma solução iterativa deste sistema, após uma escolha inicial, a técnica numérica utilizada é o método de Newton-Raphson (NR) (APÊNDICE). Cada elemento não linear é linearizado, resultando um circuito linear equivalente. Logo, a cada iteração NR o problema fica reduzido à solução de um sistema linear, Raghuram (1989).

Por exemplo, para a polarização de um diodo exponencial, Fig. 4.1, o ponto de operação DC é obtido a partir das equações de circuito (4.1) e (4.3), e, da característica $I-V$ do diodo definida, (4.2):

$$i_d = \frac{V_s - v_d}{R} \quad (\text{Thevenin}) \quad \text{ou} \quad \frac{v_d}{R} + i_d = \frac{V_s}{R} \quad (\text{Norton}) \quad (4.1)$$

$$i_d = I_{SAT} \left[\exp\left(\frac{qv_d}{\eta_d k_B T_J}\right) - 1 \right] \quad (4.2)$$

$$\therefore f(v_d) = \frac{V_s - v_d}{R} - I_{SAT} \left[\exp\left(\frac{qv_d}{\eta_d k_B T_J}\right) - 1 \right] = 0 \quad (4.3)$$

A solução iterativa de (4.3) com o uso do método de NR é obtida por:

$$v_d(n+1) = v_d(n) - \frac{\frac{V_s - v_d(n)}{R} - I_{SAT} \left[\exp\left(\frac{v_d(n)}{\eta_d V_T}\right) - 1 \right]}{\frac{-1}{R} - I_{SAT} \exp\left(\frac{v_d(n)}{\eta_d V_T}\right)} \quad (4.4)$$

Para uma boa escolha da solução inicial $v_d(0)$, (4.4) converge para a solução de (4.3). Contudo, a convergência não é garantida em todos os casos. Quando a escolha inicial estiver distante da solução DC, o método diverge. Na Fig. 4.1, apresenta-se uma representação gráfica da convergência do método NR para o circuito a diodo indicado. Usando-se (4.4) a partir da solução inicial $v_d(0) = 0,25$ V, calcula-se a seqüência de soluções. Graficamente, cada solução é localizada pela interseção entre a reta de carga e a reta tangente à curva característica do diodo (no ponto onde a solução é conhecida). Após nove iterações NR a solução DC foi encontrada: $v_d = 0,416$ V e $i_d = 0,839$ A.

4.2.2 Análise Transitória Linear

Através da análise transitória linear é possível precisar as formas de onda de um circuito sob a excitação de fontes, que são funções arbitrárias do tempo. Aqui, restringe-se a análise para circuitos lineares e invariantes no tempo, contendo elementos que armazenam energia

(capacitores e indutores). Justamente, são estes elementos que introduzem os termos da forma dv/dt e di/dt nas equações de circuito. No caso mais geral, aplicando-se as leis de Kirchhoff, obtém-se a formulação transitória na forma de um sistema de equações diferenciais ordinárias (EDOs).

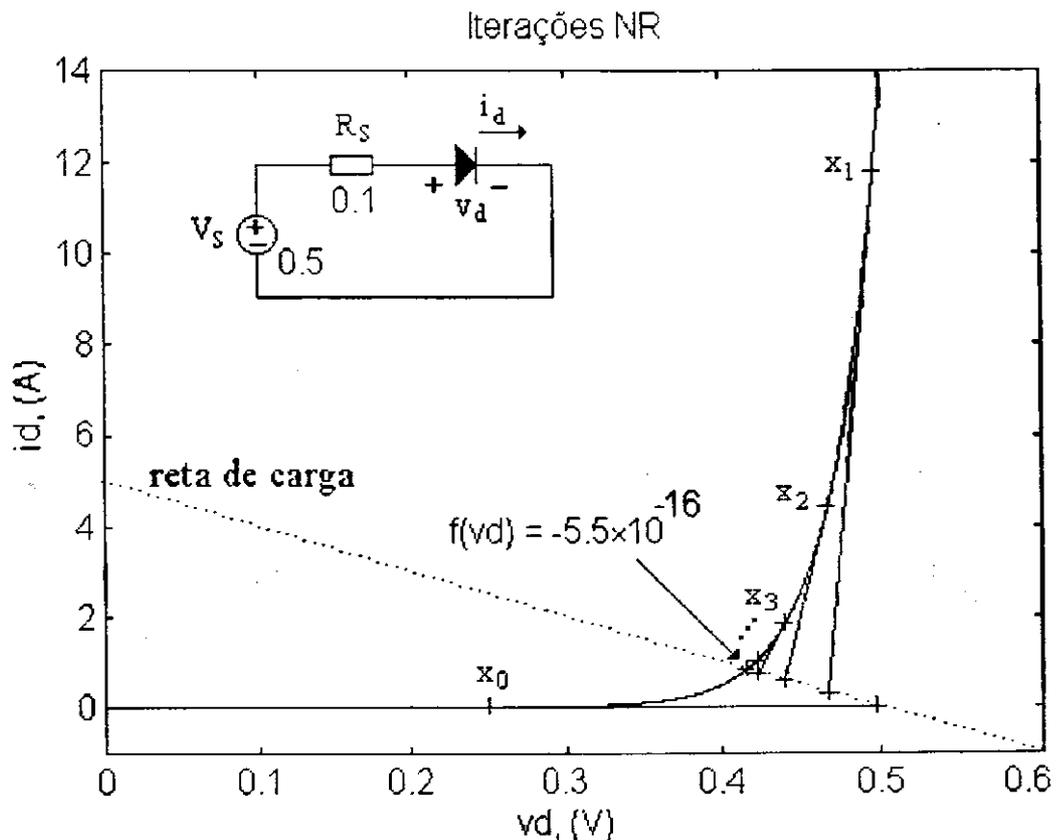


Fig. 4.1 Ponto de operação DC do circuito para polarização de um diodo exponencial ($\eta_d = 1,2$; $I_{SAT} = 1\mu A$; $T_J = 22^\circ C$) através do método de Newton-Raphson.

No método da integração direta, as EDOs que descrevem capacitores e indutores são escritas na forma integral. Por sua vez, as integrais são aproximadas numericamente através da regra trapezoidal. Desta aproximação, resultam circuitos equivalentes Norton ou Thevenin, para os quais, apenas um sistema de equações algébricas precisa ser resolvido. Este procedimento é descrito em seguida. A EDO que descreve um capacitor de capacitância C é dada por,

$$i = C \frac{dv}{dt} \quad (4.5)$$

Integrando-se (4.5) num intervalo de tempo de interesse, obtém-se,

$$v(t + \Delta t) = v(t) + \frac{1}{C} \int_t^{t+\Delta t} i(\tau) d\tau \quad (4.6)$$

Do cálculo numérico de (4.6), via a regra trapezoidal, Gerald (1970), resulta

$$v(t + \Delta t) = v(t) + \frac{\Delta t}{2C} [i(t) + i(t + \Delta t)] \quad (4.7)$$

Do mesmo modo, a partir da EDO para um indutor de indutância L, resulta:

$$v = L \frac{di}{dt} \quad (4.8)$$

$$i(t + \Delta t) = i(t) + \frac{1}{L} \int_t^{t+\Delta t} v(\tau) d\tau \quad (4.9)$$

$$i(t + \Delta t) = i(t) + \frac{\Delta t}{2L} [v(t) + v(t + \Delta t)] \quad (4.10)$$

Por intermédio de (4.7) e (4.10), construímos os circuitos equivalentes Thevenin e Norton para o capacitor e para o indutor, respectivamente, Fig. 4.2.

Na implementação computacional do método da integração direta, as fontes analógicas independentes precisam ser discretizadas. Aqui, o intervalo de tempo (passo) entre dois valores discretos destas fontes é considerado constante. Seus valores são representados por rampas ou degraus no tempo, Pillage et al (1995). Quanto menor o passo considerado, maior a precisão do método. Contudo, será maior o esforço computacional requisitado. Em seguida, dois exemplos simples servem como ilustração deste método.

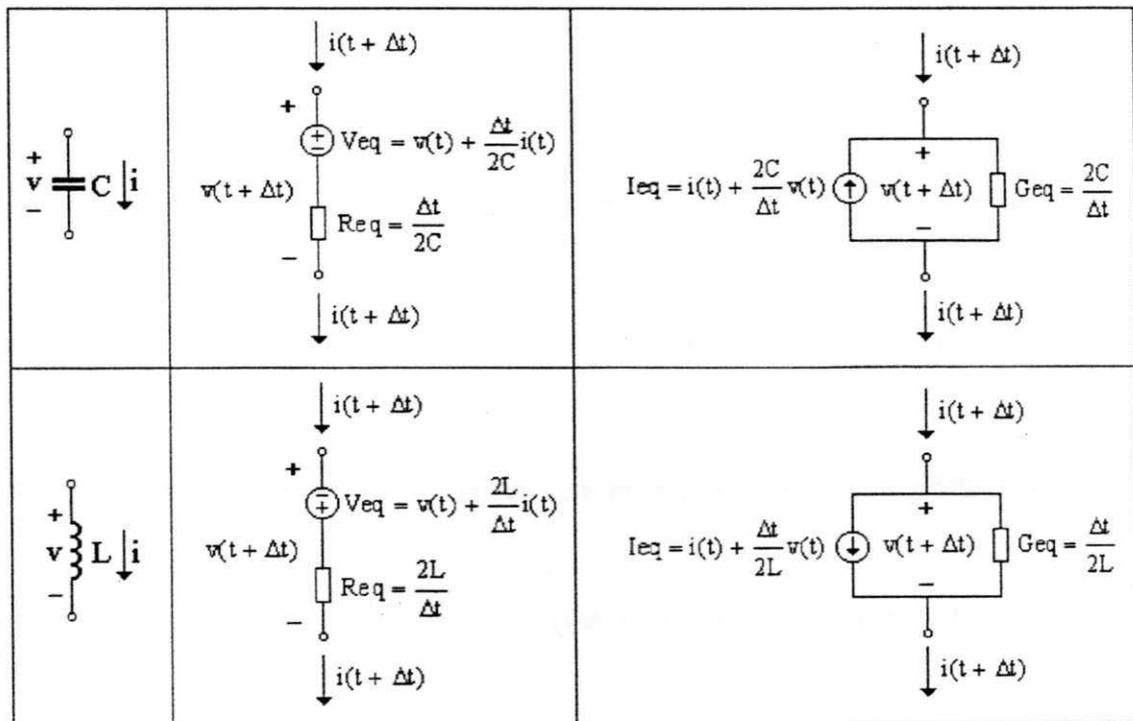


Fig. 4.2 Circuitos equivalentes Thevenin e Norton para capacitores e indutores resultantes da integração numérica via a regra dos trapézios.

Num primeiro exemplo, formulou-se o projeto de um sistema de segunda ordem, sub-amortecido, com primeira ultrapassagem em 15%, tempo de estabilização em 5% de 0,8 ms e ganho estático unitário. Neste caso, utilizou-se o análogo elétrico – o circuito RLC série, Fig. 4.3(a), com $L = 160$ mH. O projeto deste circuito é baseado no cálculo dos parâmetros da equação de segunda ordem:

$$\frac{d^2 v_C(t)}{dt^2} + 2\xi_n \omega_n \frac{dv_C(t)}{dt} + \omega_n^2 v_C(t) = K \omega_n^2 v(t) \quad (4.11)$$

em que,

$$m_{15\%} = K \exp\left(-\xi_n \pi / \sqrt{1 - \xi_n^2}\right) \text{ e } K = 1 \Rightarrow \xi_n \approx 0,5 \quad (4.12)$$

$$\xi_n = \frac{R}{2} \sqrt{\frac{C}{L}} \quad (4.13)$$

$$\omega_n = \frac{1}{\sqrt{LC}} \approx \frac{2\pi \times 85\%}{t_{s5\%}} = 6283,19 \text{ rad/s} \quad (4.14)$$

Com os dados anteriores, foram obtidos os seguintes valores: $R = 1020 \text{ } \Omega$ e $C = 160 \text{ nF}$.

Na análise deste circuito RLC através do método da integração direta, o primeiro passo é obter o respectivo circuito equivalente. Para isto, recorre-se aos modelos Thevenin mostrados na Fig. 4.2, resultando o circuito da Fig. 4.3(b). Resolvendo a equação de malha para este circuito,

$$V_S(t + \Delta t) - Ri_L(t + \Delta t) + [V_{eq_L} - Re_{q_L}i_L(t + \Delta t)] - [V_{eq_C} + Re_{q_C}i_L(t + \Delta t)] = 0 \quad (4.15)$$

obtém-se a resposta transitória ao degrau unitário do circuito RLC projetado, Fig. 4.3(c), inicialmente em repouso.

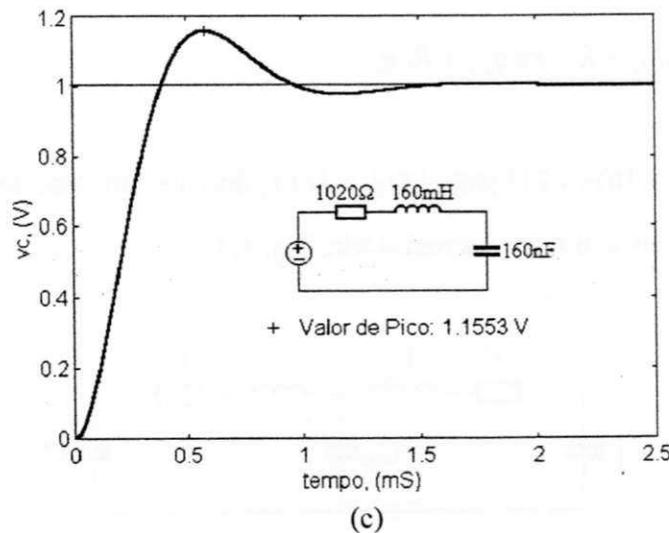
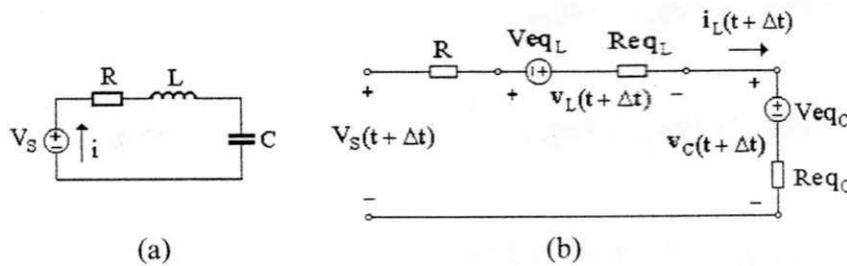


Fig. 4.3 (a) Circuito RLC série. (b) Circuito equivalente Thevenin para análise transitória.
(c) Resposta ao degrau do circuito RLC com condições iniciais nulas, $\Delta t = 1,0 \text{ } \mu\text{s}$.

Um exemplo interessante é o circuito de *beat-generation*, Fig. 4.4(a). O capacitor central funciona como uma ‘ponte de energia’ entre os capacitores extremos. Este fenômeno ocorre em muitos sistemas físicos quase idênticos e fracamente acoplados. O circuito equivalente Thevenin para a análise transitória é mostrado na Fig. 4.4(b). A partir da condição inicial, $x_4(0) = 1,0$ V, a resposta transitória foi calculada com o auxílio das equações de malha:

$$Veq_1 - Req_1 i_0(t + \Delta t) + Req_{C1} i_1(t + \Delta t) = 0 \quad (4.16)$$

$$Veq_2 - Req_2 i_1(t + \Delta t) + Req_{C1} i_0(t + \Delta t) = 0 \quad (4.17)$$

em que,

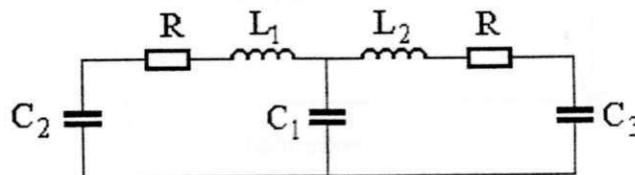
$$Veq_1 = Veq_{C2} - Veq_{L1} - Veq_{C1} \quad (4.18)$$

$$Veq_2 = Veq_{C1} + Veq_{L2} - Veq_{C3} \quad (4.19)$$

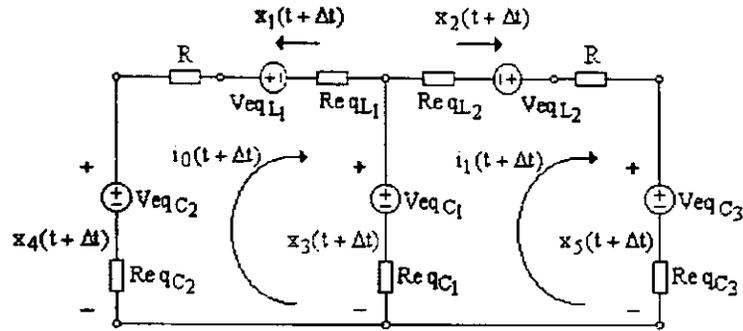
$$Req_1 = Req_{C2} + R + Req_{L1} + Req_{C1} \quad (4.20)$$

$$Req_2 = Req_{C1} + R + Req_{L2} + Req_{C3} \quad (4.21)$$

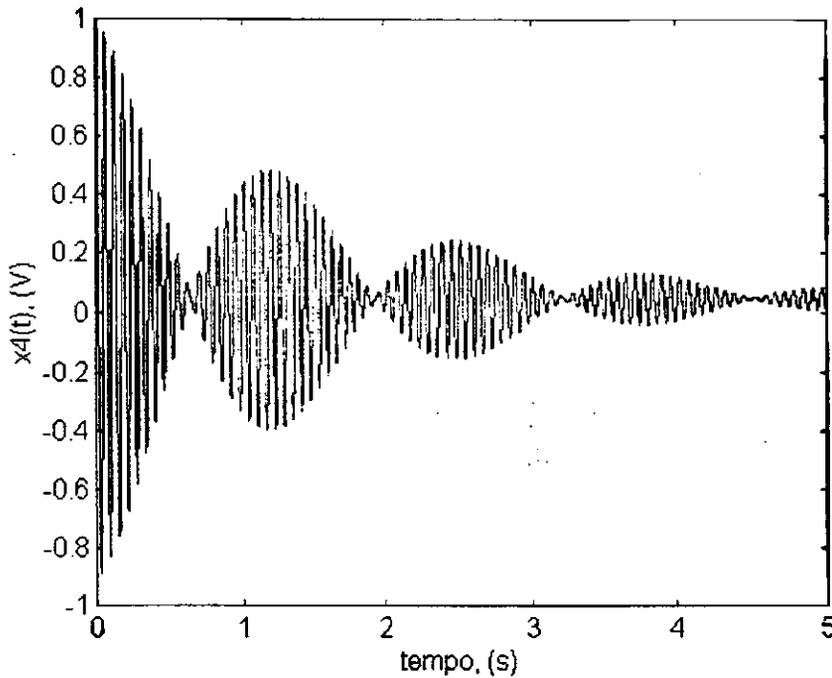
Com a solução de (4.16)-(4.21) para $i_0(t)$ e $i_1(t)$, durante um dado tempo de simulação, recursivamente obtém-se a resposta transitória, Fig. 4.4(c).



(a)



(b)



(c)

Fig. 4.4 (a) Circuito *beat generation*. (b) Equivalente Thevenin para análise transitória.

(c) Resposta transitória para $x_4(t)$. dados: condições iniciais nulas, exceto $x_4(0) = 1,0$ V,

$$L_1 = L_2 = 0,01 \text{ H}, C_1 = 0,2 \text{ F}, C_2 = C_3 = 0,01 \text{ F}, R = 0,025 \text{ } \Omega \text{ e } \Delta t = 0,5 \text{ } \mu\text{s}.$$

4.2.3 Análise Transitória Não Linear

Este tipo de análise é útil na determinação do comportamento para grandes sinais (no domínio do tempo) de circuitos contendo dispositivos não lineares. Considera-se a solução de circuitos contendo fontes independentes, que são funções arbitrárias do tempo. Em

última análise, trata-se do cálculo da solução de equações diferenciais não lineares simultâneas. Isto implica num custo computacional elevado.

Basicamente, através da interpretação do circuito em questão, em cada passo do tempo, os modelos associados para capacitores e indutores são definidos. Estes dependem do esquema de integração utilizado. Em geral, estes modelos contêm resistores não lineares. Com isto, o problema da análise transitória fica reduzido à solução DC de um circuito não linear em cada passo do tempo. Este problema foi abordado na seção 4.1.1.

Neste estudo, o procedimento numérico é semelhante àquele empregado em simuladores de circuitos comerciais como o SPICE: combinação das equações nodais modificadas, iteração NR e integração via a regra trapezoidal. Quando apropriadamente implementado, este método pode levar a análise transitória de circuitos não lineares com precisão elevada (bem abaixo da precisão com que circuitos integrados podem ser construídos), Pillage et al (1995).

Contudo, o custo em suporte e manutenção do programa, desenvolvimento de modelos de dispositivos e utilização de recursos de CPU, são elevados. Especificamente, o esforço computacional tem sido um obstáculo para a aplicação do método a circuitos maiores. O objetivo aqui não é implementar um simulador de circuitos de uso geral. Antes, do ponto de vista computacional, o interesse é na aplicação da técnica através das ANNs para tornar o método no domínio do tempo mais eficiente.

A seguir, demonstra-se a aplicação do método para alguns circuitos elementares. Como um primeiro exemplo, aplicou-se o procedimento descrito acima para um circuito limitador construído com dois diodos exponenciais, Fig. 4.5(a). Dada a relação exponencial para os diodos, (4.2), com $\alpha_D = q/(\eta_a kT_j)$, resulta para este circuito:

$$f_1(v_1, v_2) = (V_S - v_1) / R_S - I_{SAT} \{ \exp[\alpha_D (v_1 - v_2)] - 1 \} = 0 \quad (4.22)$$

$$f_2(v_1, v_2) = I_{SAT} \{ \exp[\alpha_D (v_1 - v_2)] - 1 \} - I_{SAT} \{ \exp[\alpha_D (v_2 - V_{Lm})] - 1 \} - v_2 / R_L = 0 \quad (4.23)$$

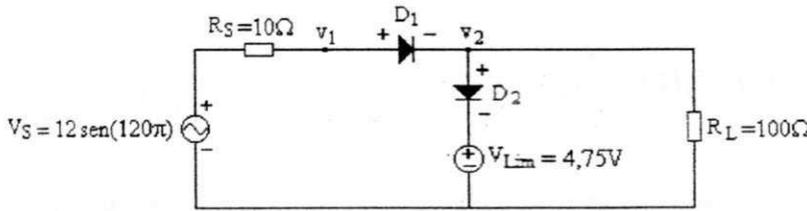
Desde que o circuito limitador, Fig. 4.5(a), não contém elementos que armazenem energia, o cálculo de sua resposta no tempo é feito diretamente a partir das equações de NR:

$$\mathbf{J}^n \Delta \mathbf{v}^n = -\mathbf{F}(\mathbf{v}^n), \quad \Delta \mathbf{v}^n = \mathbf{v}[t+n\Delta t] - \mathbf{v}[t+(n-1)\Delta t] \quad (4.24)$$

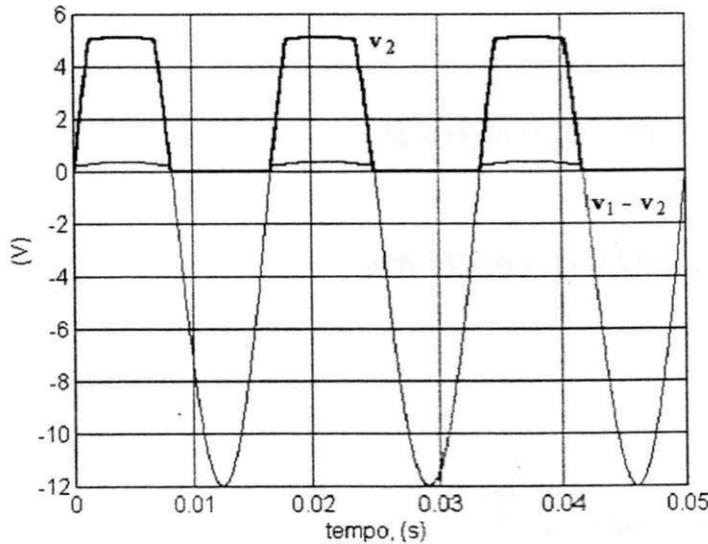
no n -ésimo passo de tempo, em que, \mathbf{J} é a matriz Jacobiana. Através de (4.22)-(4.23), resulta:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_1}{\partial v_2} \\ \frac{\partial f_2}{\partial v_1} & \frac{\partial f_2}{\partial v_2} \end{bmatrix} = \begin{bmatrix} -1/R_S - \alpha_D I_{SAT} \exp[\alpha_D (v_1 - v_2)] \\ \alpha_D I_{SAT} \exp[\alpha_D (v_1 - v_2)] \\ \alpha_D I_{SAT} \exp[\alpha_D (v_1 - v_2)] \\ -\alpha_D I_{SAT} \exp[\alpha_D (v_1 - v_2)] - \alpha_D I_{SAT} \exp[\alpha_D (v_1 - V_{Lim})] - 1/R_L \end{bmatrix} \quad (4.25)$$

Finalmente, concluí-se a análise transitória resolvendo numericamente o sistema (4.24) através do método da fatoração LU, Gerald (1970). Um resultado de simulação para o circuito limitador a diodo é apresentado na Fig. 4.5(b).



(a)



(b)

Fig. 4.5 (a) Circuito limitador a diodo. (b) Resposta transitória.

Num segundo exemplo, simulou-se um transistor de junção bipolar (BJT) funcionando como uma chave eletrônica, Fig. 4.6(a). Com o emprego do modelo de Ebers-Moll (1954), que modela a característica não linear do BJT, resulta o circuito equivalente da Fig. 4.6(b). Por sua vez, através da linearização NR, os diodos exponenciais deste modelo são substituídos por circuitos equivalentes Norton. Com isto, obtém-se o circuito linearizado da Fig. 4.6(c). Na computação da resposta transitória (em cada passo de tempo) apenas uma análise 'DC' não linear é necessária, com a iteração NR baseada no circuito da Fig. 4.6(c).

O modelo não linear do BJT relaciona as correntes no transistor através das expressões, Ebers e Moll (1954).

$$i_b(t) = -i_e(t) - i_c(t) \quad (4.26)$$

$$i_e(t) = -i_F(t) + \alpha_R i_R(t) \quad (4.27)$$

$$i_c(t) = \alpha_F i_F(t) - i_R(t) \quad (4.28)$$

em que, $i_F(t)$ e $i_R(t)$ são dadas por (4.2), $\alpha_F = 0,99$ e $\alpha_R = 0,5$. Por outro lado, na iteração NR utilizou-se o circuito equivalente Norton linearizado para os diodos, resultando:

$$i_F(t + \Delta t) = I_{eq}^e(t) + G_{eq}^e(t) v_{be}(t + \Delta t) \quad (4.29)$$

$$i_R(t + \Delta t) = I_{eq}^c(t) + G_{eq}^c(t) v_{bc}(t + \Delta t) \quad (4.30)$$

em que,

$$I_{eq}^e(t) = i_F(t) - v_{be}(t) \frac{di_F(t)}{dv_{be}(t)} \quad (4.31)$$

$$G_{eq}^e(t) = \frac{di_F(t)}{dv_{be}(t)} \quad (4.32)$$

$$I_{eq}^c(t) = i_R(t) - v_{bc}(t) \frac{di_R(t)}{dv_{bc}(t)} \quad (4.33)$$

$$G_{eq}^c(t) = \frac{di_R(t)}{dv_{bc}(t)} \quad (4.34)$$

As seguintes equações de malha descrevem o circuito,

$$f_1(v_{be}, v_{bc}) = i_b(t) - [V_S(t) - v_{be}(t)] / R_B = 0 \quad (4.35)$$

$$f_2(v_{be}, v_{bc}) = i_c(t) - [V_{cc} + v_{bc}(t) - v_{be}(t)] / R_C = 0 \quad (4.36)$$

além disso,

$$i_b(t) = -i_c(t) - i_e(t). \quad (4.37)$$

No cálculo da resposta transitória, resolve-se (4.24) por fatoração LU. Usando-se (4.35)-(4.36) para o cálculo da matriz Jacobiana, determina-se:

$$\mathbf{J} = \begin{bmatrix} (1 - \alpha_F)G_{eq}^e + 1/R_B & (1 - \alpha_R)G_{eq}^c \\ \alpha_F G_{eq}^e + 1/R_C & -G_{eq}^c - 1/R_C \end{bmatrix} \quad (4.38)$$

A resposta obtida para $\Delta t = 1,0 \mu s$ é ilustrada na Fig. 4.7.

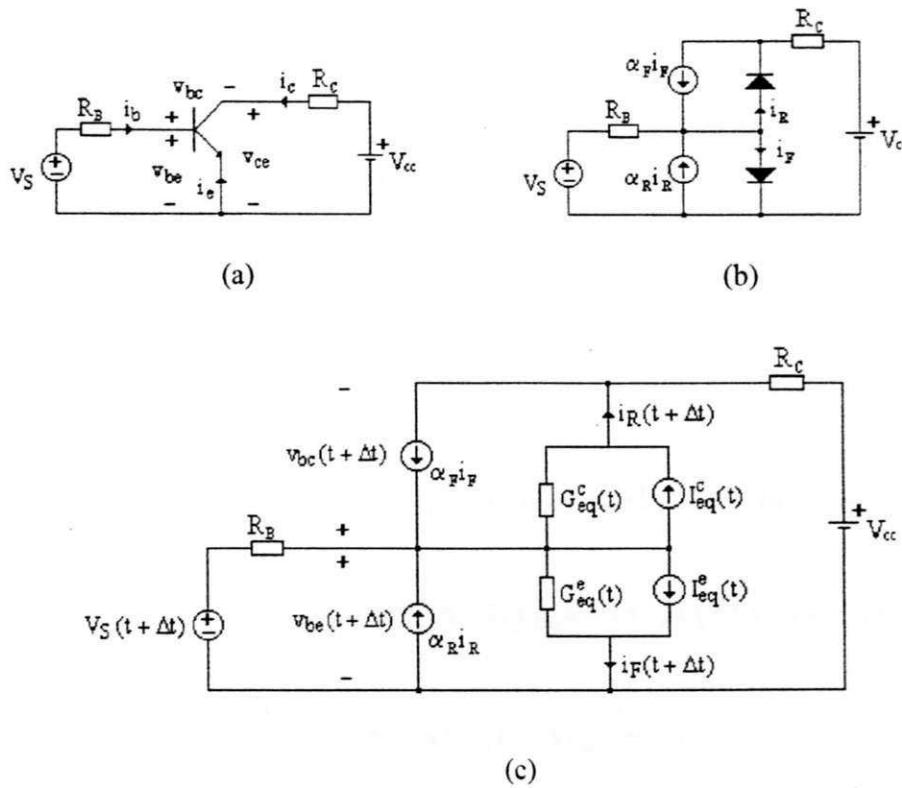


Fig. 4.6 (a) Circuito chave eletrônica com um BJT. (b) Inclusão do modelo de Ebers-Moll. (c) Circuito equivalente linearizado usado na iteração NR.

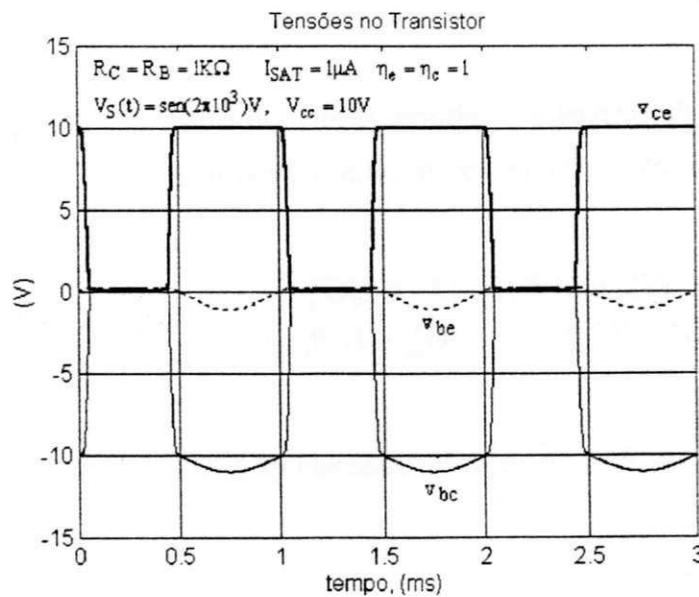


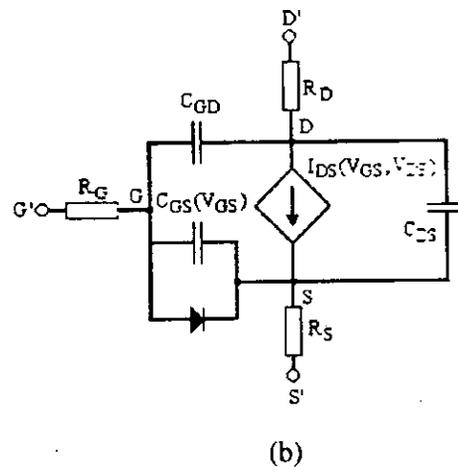
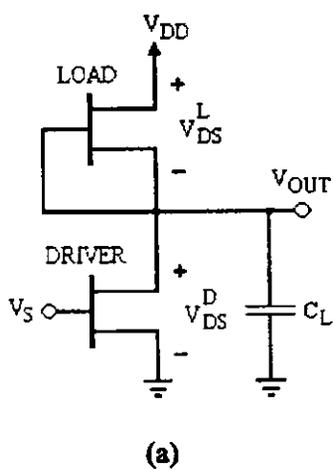
Fig. 4.7 Resposta no tempo para o circuito chave eletrônica com um BJT.

Num último exemplo de análise transitória não linear, considera-se um GaAs MESFET de canal tipo-n como amplificador com uma carga não linear, Fig. 4.8(a). Este circuito é uma porta lógica sem o circuito de mudança de nível, Curtice (1980). Para o GaAs MESFET, supõe-se o circuito equivalente para grandes sinais proposto por Curtice (1980), Fig. 4.8(b). Ele consiste basicamente de uma fonte de corrente controlada por tensão, três capacitâncias entre eletrodos e um diodo ideal com o propósito de 'grampeamento', quando $V_{GS} > 0$. Os resistores representam as resistências na região de contato. Este modelo contém apenas dois elementos não lineares, que são descritos por:

$$I_{DS}(V_{GS}, V_{DS}) = \beta_{FET} (V_{GS} + V_{Th})^2 (1 + \lambda_{FET} V_{DS}) \tanh(\alpha_{FET} V_{DS}) \quad (4.39)$$

$$C_{GS}(V_{GS}) = C_{GS}(0) / \sqrt{(1 - V_{GS} / V_{BI})} \quad (4.40)$$

Admitindo-se que $V_{GS} \leq 0$, pode-se ignorar o diodo no modelo do MESFET. Por outro lado, considerando-se: resistências de contato nulas; o curto-circuito entre porta e fonte no MESFET funcionando como uma carga não linear, Fig. 4.8(a); $C_{DS} = 0$: obteve-se o circuito resultante, Fig. 4.8(c), Silva et al (2001).



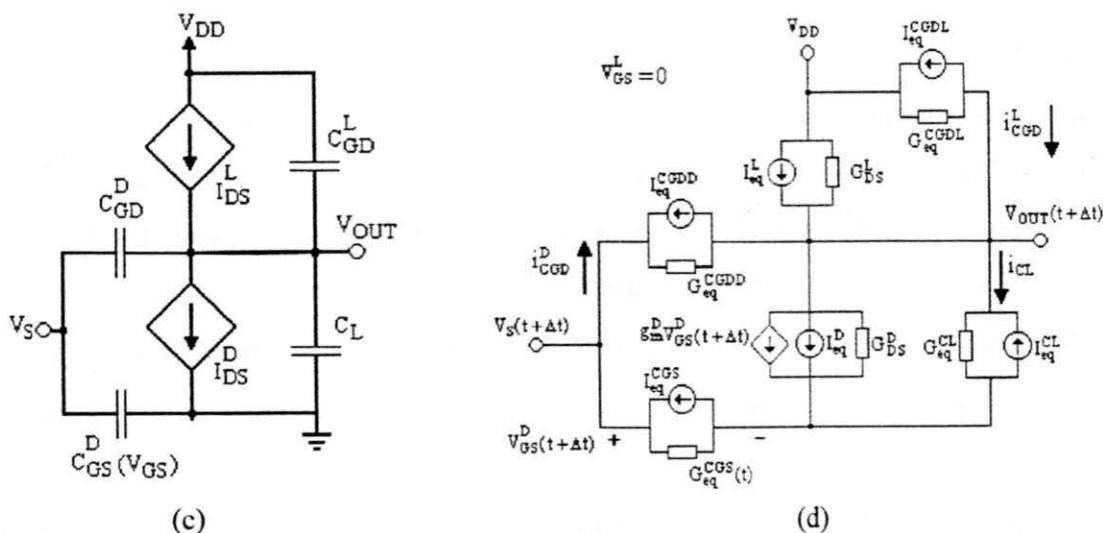


Fig. 4.8 (a) Porta lógica com GaAs MESFET sem o circuito de mudança de nível. (b) Circuito equivalente para o GaAs MESFET. (c) Circuito não linear resultante. (d) Circuito equivalente linearizado para análise transitória.

Dois equações, uma nodal e outra de malha, foram empregadas na análise do circuito:

$$f_1(V_{DS}^L, V_{DS}^D) = i_{CGD}^D + I_{DS}^L - I_{DS}^D - i_{CL} + i_{CGD}^L = 0 \quad (4.41)$$

$$f_2(V_{DS}^L, V_{DS}^D) = V_{DD} - V_{DS}^L - V_{DS}^D = 0 \quad (4.42)$$

No cálculo da resposta transitória, todos os elementos, Fig. 4.8(c), precisam ser linearizados. Em particular, o circuito Norton empregado como modelo linearizado do MESFET é definido por intermédio da relação recursiva:

$$I_{DS}(t + \Delta t) = I_{eq}(t) + g_m(t)V_{GS}(t + \Delta t) + G_{DS}(t)V_{DS}(t + \Delta t) \quad (4.43)$$

em que,

$$I_{eq}(t) = I_{DS}(t) + g_m(t)[-V_{GS}(t)] + G_{DS}(t)[-V_{DS}(t)] \quad (4.44)$$

$$g_m(t) = \frac{\partial I_{DS}(t)}{\partial V_{GS}(t)} \quad (4.45)$$

$$G_{DS}(t) = \frac{\partial I_{DS}(t)}{\partial V_{DS}(t)} \quad (4.46)$$

Por outro lado, define-se a seguinte relação para o capacitor não linear $C_{GS}(V_{GS})$

$$i_{CGS}(t) = C_{GS}(V_{GS}) \frac{dV_{GS}}{dt} \quad (4.47)$$

em que, $C_{GS}(V_{GS})$ é dado por (4.40). Da integração numérica de (4.47) via a regra trapezoidal, resulta o modelo linearizado do capacitor não linear, Pillage et al (1995):

$$i_{CGS}(t + \Delta t) = G_{eq}^{CGS}(t) V_{GS}(t + \Delta t) - I_{eq}^{CGS}(t) \quad (4.48)$$

em que,

$$G_{eq}^{CGS}(t) = \frac{2C_{GS}(t + \Delta t)}{\Delta t} \quad (4.49)$$

$$I_{eq}^{CGS}(t) = \frac{C_{GS}(t + \Delta t)}{C_{GS}(t)} i_{CGS}(t) + \frac{2C_{GS}(t + \Delta t)}{\Delta t} V_{GS}(t) \quad (4.50)$$

Os demais capacitores da Fig. 4.8(c) são lineares e seus equivalentes Norton são dados na Fig. 4.2. Na Fig. 4.8(d) representa-se o circuito resultante linearizado para a análise transitória. Com este circuito, efetua-se (4.41)-(4.42) no tempo $(t + \Delta t)$, em que:

$$i_{CGD}^D(t + \Delta t) = [V_S(t + \Delta t) - V_{OUT}(t + \Delta t)] G_{eq}^{CGDL}(t) - I_{eq}^{CGDL}(t) \quad (4.51)$$

$$i_{CGD}^L(t + \Delta t) = [V_{DD} - V_{OLT}(t + \Delta t)]G_{eq}^{CGDD}(t) - I_{eq}^{CGDD}(t) \quad (4.52)$$

$$i_{CL}(t + \Delta t) = V_{OUT}(t + \Delta t)G_{eq}^{CL}(t) - I_{eq}^{CL}(t) \quad (4.53)$$

Agora, a resposta transitória pode ser calculada. Para isto, resolve-se (4.24) por fatoração LU, calculando-se a matriz Jacobiana através de (4.41)-(4.42):

$$\mathbf{J} = \begin{bmatrix} G_{DS}^L + G_{eq}^{CGDL} & -G_{eq}^{CGDL} & -G_{DS}^D & -G_{eq}^{CL} \\ -1 & & -1 & \end{bmatrix} \quad (4.54)$$

Supondo-se os dois MESFETs idênticos, mas o amplificador com o dobro da largura daquele funcionando como carga não linear, simulou-se a resposta transitória da porta lógica. Com as condições iniciais nulas, exceto que: $V_{DS}^D(0) = 0,3661$ V e $V_{DS}^L(0) = 3,1339$ V, na Tabela 4.1 estão indicados os valores empregados na simulação. Para evitar a duplicação destes resultados, posterga-se sua apresentação para a Fig. 4.13.

Tabela 4.1 Valores utilizados no cálculo da resposta transitória.

$\alpha_{FET} = 1,5 \text{ V}^{-1}$	$C_{GD} = 0,3 \text{ fF}$
$\beta_{FET} = 65 \text{ } \mu\text{A}/\text{V}^2$	$C_{DS} = 0$
$V_{Th} = 2,5 \text{ V}$	$C_L = 6 \text{ fF}$
$V_{BI} = 0,5 \text{ V}$	$R_S = R_G = R_D = 0$
$\tau_{FET} = 10 \text{ ps,}$	$\lambda_{FET} = 0$
$C_{GS}(0) = 6 \text{ fF}$	$V_{DD} = 3,5 \text{ V}$

4.3 MODELOS NEURAIIS PARA GaAs MESFETs

Desde o desenvolvimento comercial da tecnologia para circuitos integrados construídos com transistores MESFETs de GaAs, tem-se presenciado o crescimento de modelos voltados para simuladores de circuitos como o SPICE. Por exemplo, nas expressões do modelo proposto por Schichman e Hodges (1968), usadas no programa SPICE2, a corrente de dreno é definida por (for $V_{DS} > 0$):

$$I_{DS}(V_{GS}, V_{DS}) = \begin{cases} 0, & V_{GS} + V_{Th} < 0 \\ \beta_{FET} (V_{GS} + V_{Th})^2 (1 + \lambda_{FET} V_{DS}), & 0 \leq V_{GS} + V_T \leq V_{DS} \\ \beta_{FET} V_{DS} [2(V_{GS} + V_T) - V_{DS}] (1 + \lambda_{FET} V_{DS}), & 0 \leq V_{DS} \leq V_{GS} + V_T \end{cases} \quad (4.55)$$

em que, V_{Th} é uma tensão de limiar, β_{FET} e λ_{FET} são constantes. Esta expressão analítica provê uma boa aproximação na região de corrente de saturação. Contudo, a interpolação torna-se imprecisa fora desta região.

O modelo de Curtice. (4.39)-(4.40), que usa a função tangente hiperbólica, é derivado do modelo de Schichman-Hodges, modificado para provê um apropriado 'joelho de potencial' na característica $I-V$. Neste modelo, o tempo de trânsito sob a porta, τ , também é um parâmetro do modelo, apesar da dependência sob τ não ser explicitada em (4.39). Este modelo têm sido implementado no código fonte do SPICE, Sussman-Fort et al (1984), com treze parâmetros.

Um modelo aperfeiçoado foi descrito por Statz et al (1984). Por sua vez, McCamant et al (1987), procuraram modelar as características específicas do comportamento do MESFET, para as quais, os modelos de Curtice, e de Statz, falham. Um modelo realístico para o SPICE é reportado por Parker e Skellern (1997), com 32 parâmetros.

Os transistores tipo MESFET são complexos internamente e modelos externos simples não conseguem descrever precisamente seu comportamento sob todas as condições. Apenas com um modelo interno bidimensional detalhado, Reiser (1973).

Barnes et al (1976), e, Yamagushi e Kodera (1977), que resultados mais precisos podem ser obtidos. Uma comparação entre as características $I-V$ descritas acima, Fig. 4.9, revela a precisão destes modelos. Os resultados 2-D são tidos como os mais precisos.

A rede MLP treinada, constrói aproximações globais para mapeamentos não lineares, com boa capacidade para extrapolação, mas com um aprendizado relativamente lento. Isto não é um problema, quando o treinamento é feito *off-line*. Aqui, admite-se as redes MLPs com funções logísticas na camada oculta, e, camada de saída é linear.

Para desenvolver o modelo MLP para GaAs MESFETs, os dados obtidos através de análise numérica 2-D apresentados na Fig. 4.9, foram empregados na geração de um conjunto com 99 exemplos de treinamento.

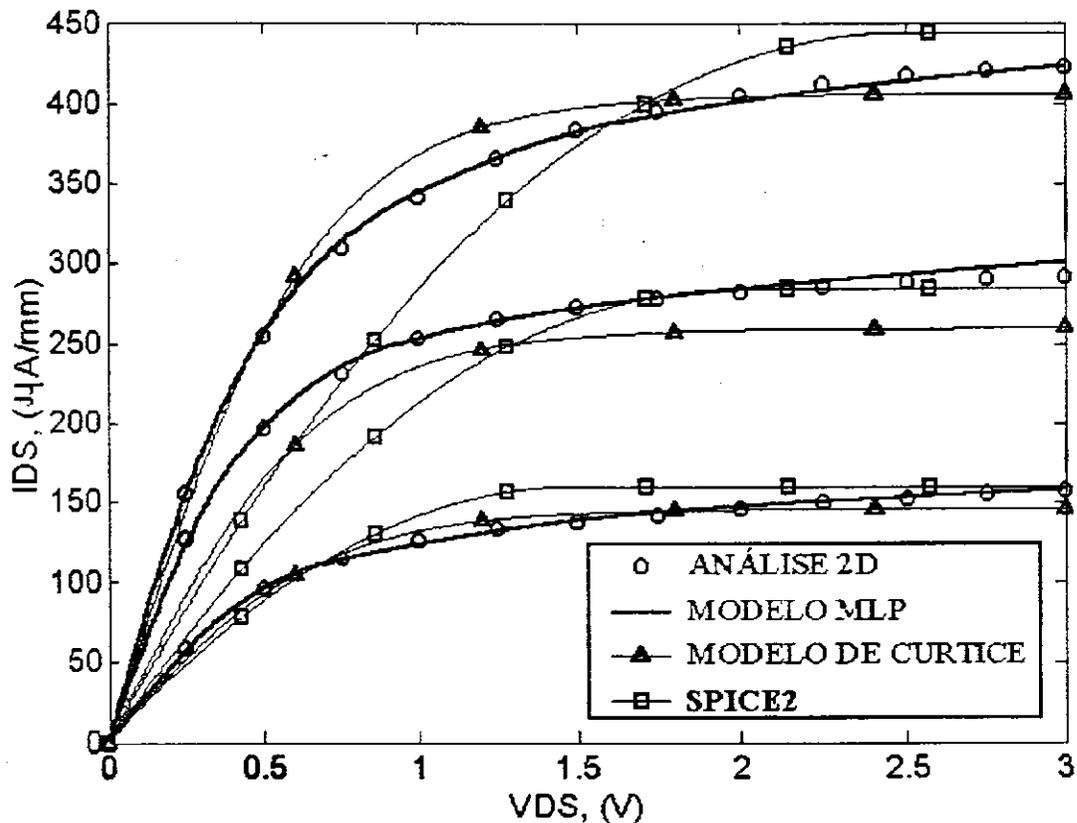


Fig. 4.9 Características $I-V$ para um GaAs MESFET. Modelos - Schichman-Hodges: $V_{Th} = 2,5\text{ V}$, $\beta_{FET} = 71\mu\text{A}/\text{V}^2$, $\lambda_{FET} = 0$; Curtice: $V_{Th} = 2,5\text{ V}$, $\beta_{FET} = 65\mu\text{A}/\text{V}^2$, $\lambda_{FET} = 0$; programa 2-D: comprimento da porta = $1,0\ \mu\text{m}$, concentração de doadores = $3 \times 10^{16}\ \text{cm}^{-3}$, espessura da camada ativa = $0,25\ \mu\text{m}$, $V_{BI} = 0,5\text{ V}$. Generalização da rede MLP (MESFET com largura = $10\ \mu\text{m}$).

No desenvolvimento do modelo MLP para GaAs MESFETs, define-se a seguinte configuração: duas entradas ($V_{GS} = [0,0; -0,5; -1,0]$ V; $V_{DS} \in [0,3]$ V), treze neurônios ocultos, e uma saída, a corrente de dreno, Silva et al (2001), Silva et al (2002). Foram empregados diferentes algoritmos para o treinamento do modelo MLP. Uma comparação (em termos de curvas de erro) entre os resultados dos algoritmos BP, delta-bar-delta, SuperSAB e Rprop, é apresentada na Fig. 4.10. Desde que, com o uso do método MLP/Rprop obtém-se o menor erro na aproximação, este modelo MLP foi o escolhido. A capacidade de generalização desta rede foi testada para exemplos fora do conjunto de treinamento, Fig. 4.9, $V_{GS} = [-0,25; -0,75]$ V.

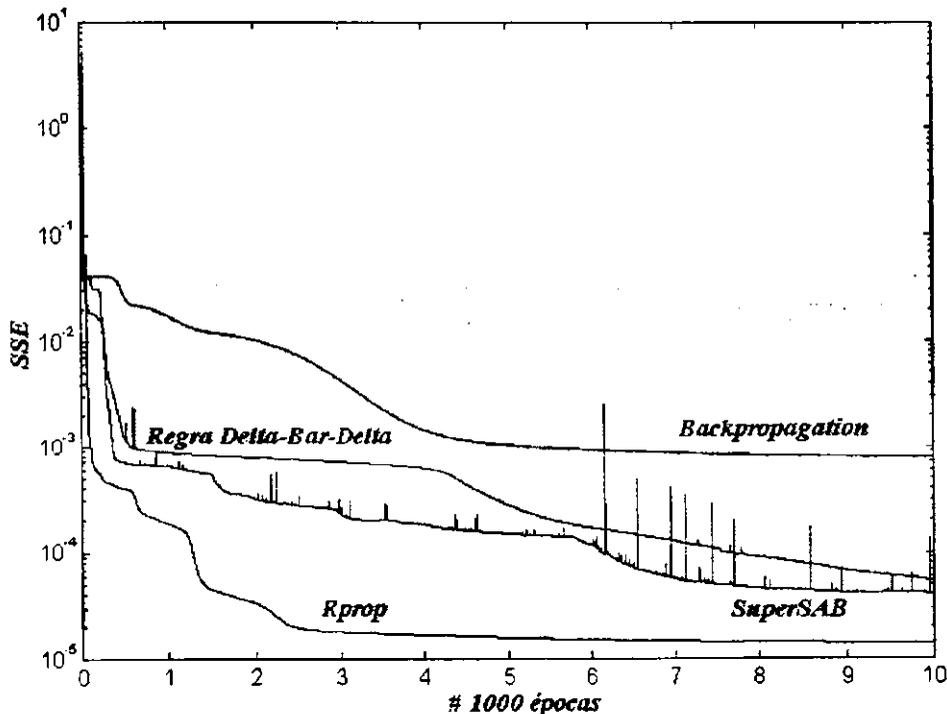


Fig. 4.10 Convergência dos algoritmos de treinamento.

Usando-se o modelo MLP descrito acima para aproximar a corrente de dreno, outros parâmetros dos MESFETs podem ser estimados: a condutância de dreno e a transcondutância. Levando-se em conta os resultados da análise 2D para a corrente de dreno, o cálculo da condutância de dreno pode ser feito por interpolação. Para propósitos de comparação, na Fig. 4.11 são apresentados os resultados obtidos com o uso de *splines* cúbicas e do modelo MLP.

Devido ao número insuficiente de dados precisos (análise 2D) para relacionar a corrente de dreno com a tensão de porta, um cálculo preciso da transcondutância do MESFET não é possível. Contudo, se o modelo MLP generaliza (responde precisamente para exemplos fora do conjunto de treinamento), então, a transcondutância pode ser estimada através deste modelo MLP, Fig 4.12.

Finalmente, concluímos a modelagem dos GaAs MESFETs, empregando os resultados anteriores para treinar um modelo MLP, com a seguinte configuração: duas entradas $x = [V_{GS} \ V_{DS}]$, vinte neurônios ocultos, e três saídas $z = [I_{DS} \ G_{DS} \ g_m]$. Este modelo foi utilizado na análise de circuitos no domínio do tempo. Por exemplo, a Fig. 4.12 ilustra o resultado para a transcondutância do MESFET, aproximada por este modelo MLP.

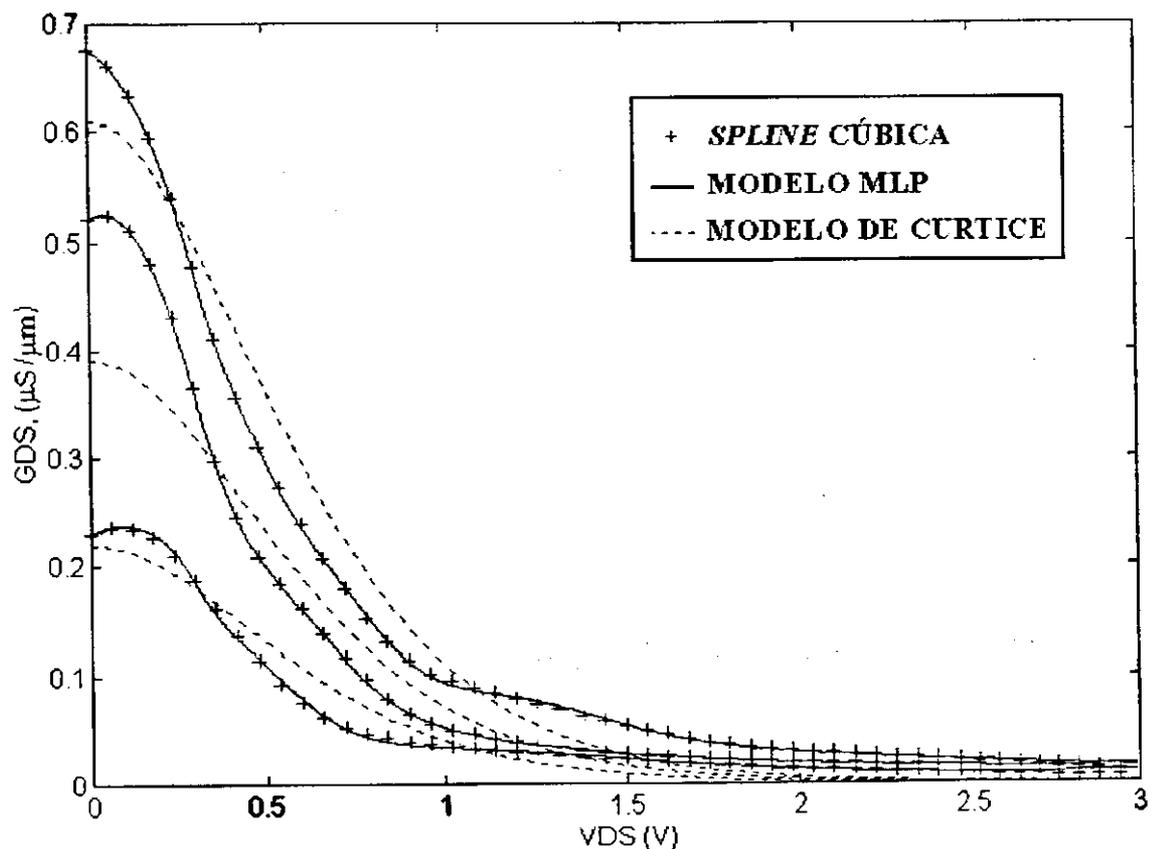


Fig. 4.11 Computações da condutância de dreno.

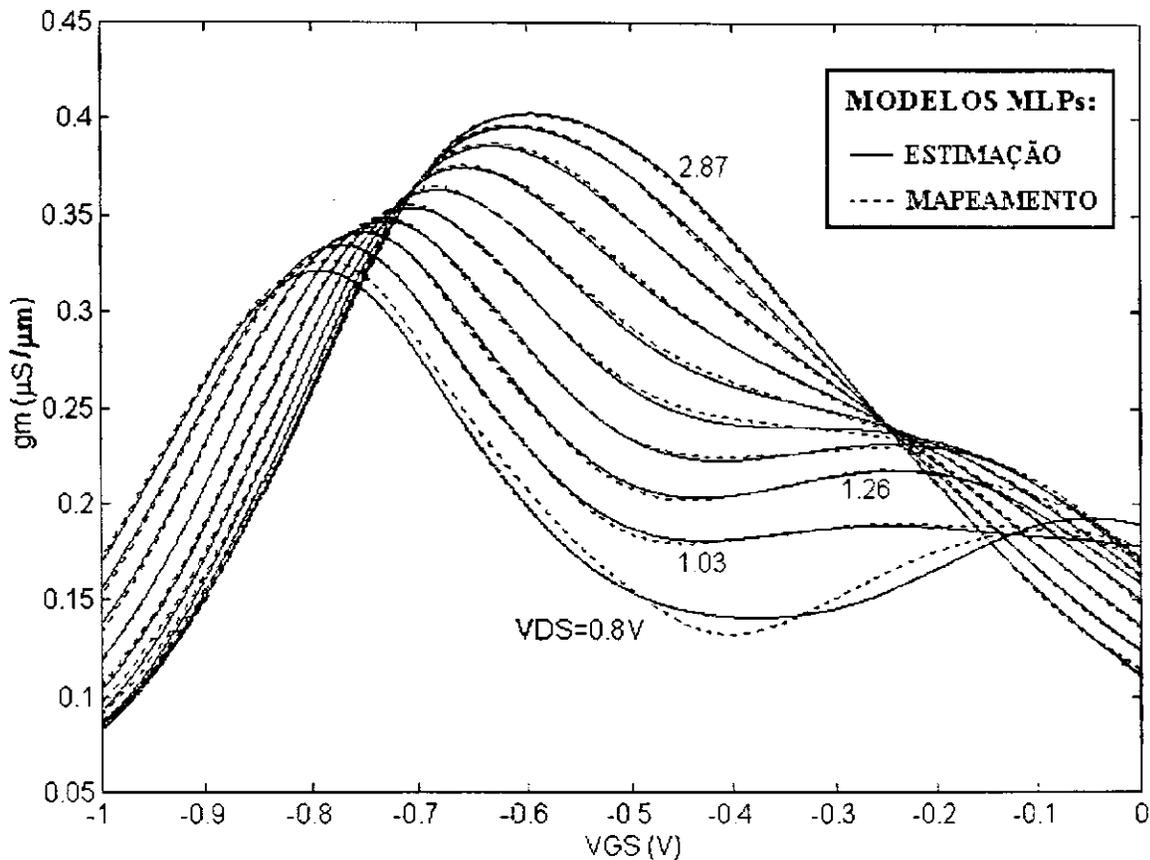


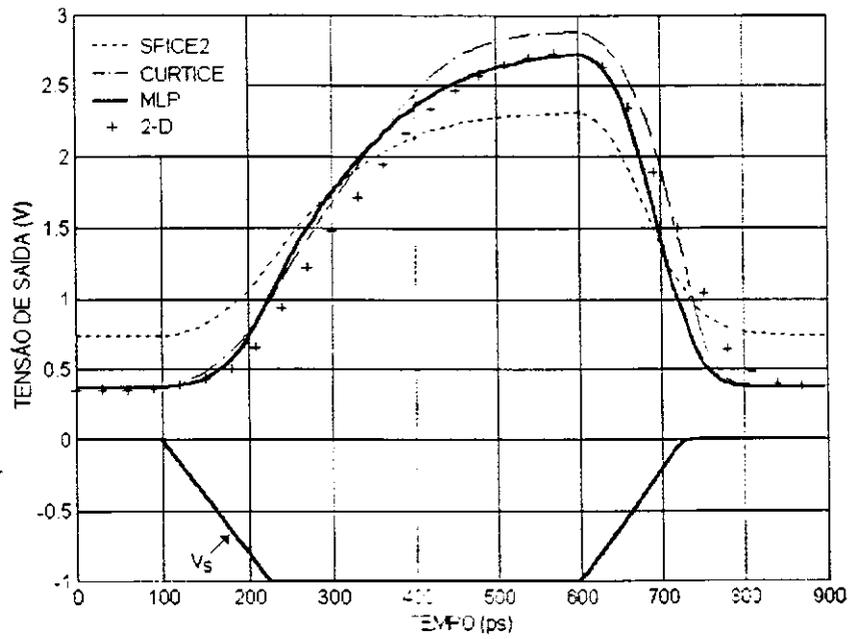
Fig. 4.12 Transcondutância do MESFET: estimativa e aproximação através de dois modelos MLP.

4.4 ANÁLISE DO CIRCUITO PORTA LÓGICA COM GaAs MESFETs

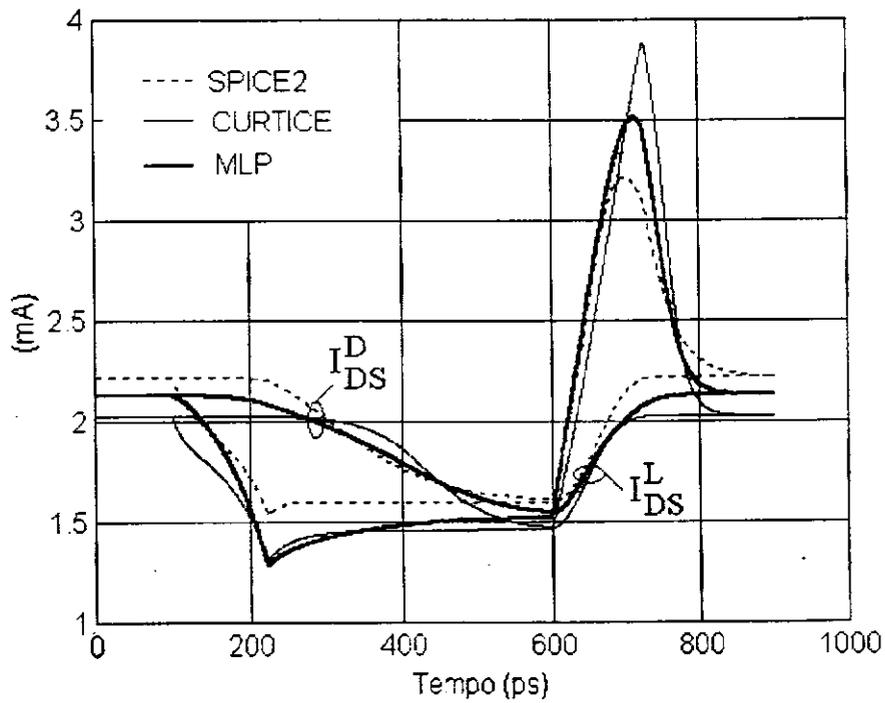
Para demonstrar a precisão e a eficiência computacional no uso do modelo MLP para GaAs MESFETs, considera-se a análise transitória do circuito porta lógica, Fig. 4.13(a), com uma entrada pulsada.

Na Fig. 4.13(a), pode-se observar que, a tensão de saída da porta lógica, obtida a partir do modelo de Schichman-Hodges, apresenta erros no tempo de subida, ganho e atraso de propagação. Os resultados obtidos no uso do modelo de Curtice são relativamente precisos. O modelo MLP é o que melhor aproxima os resultados 2-D. Na Fig. 4.13(b) tem-se as formas de onda obtidas para as correntes de dreno dos MESFETs. A partir das

trajetórias dinâmicas sobre as respectivas curvas $I-V$ de cada modelo, Fig. 4.13(c), o comportamento do MESFET *driver* pôde ser verificado.



(a)



(b)

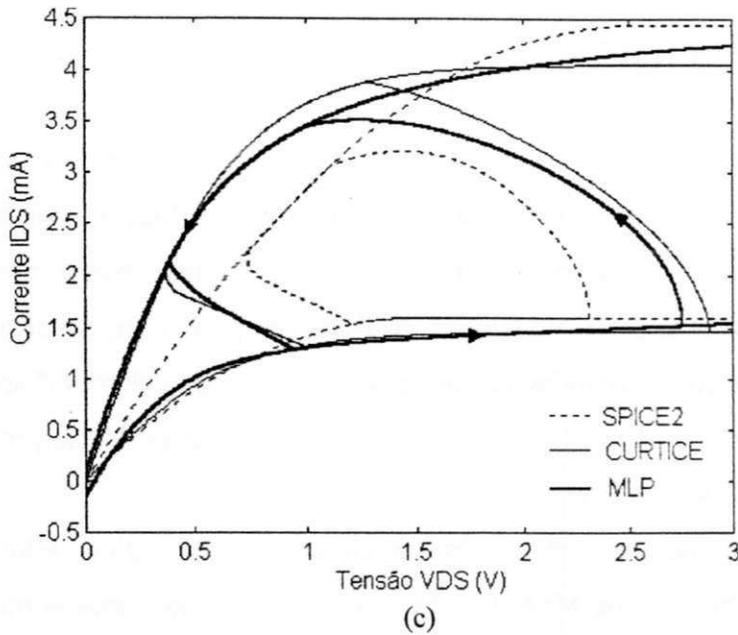


Fig. 4.13 Comparação entre os resultados de simulação do circuito porta lógica com tensão de entrada pulsada para três modelos do GaAs MESFET. Os valores usados são os mesmos dados na Tab. 4.1 e na legenda da Fig. 4.10. (a) Tensões de saída. (b) Correntes de dreno. (c) Características $I-V$ do MESFET driver e trajetórias dinâmicas de tensão-corrente.

O tempo de ocupação da CPU para os modelos simulados, são dados na Fig. 4.14. Por inspeção direta, observa-se que: o esforço computacional do modelo MLP é bem menor que o do modelo de Curtice; um pouco maior que o do modelo de Schichman-Hodges.

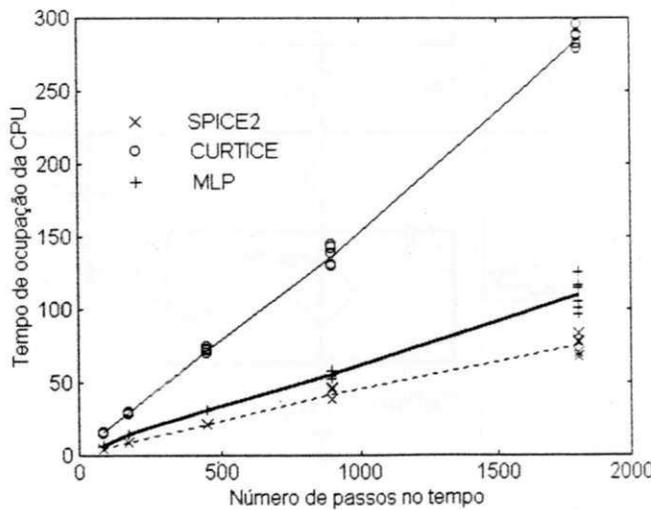


Fig. 4.14 Comparação entre os tempos de ocupação da CPU (simulação em MatlabTM (1998) sob um processador AMD-500).

4.5 ANÁLISE DE SENSIBILIDADE DO CIRCUITO PORTA LÓGICA

O sucesso obtido nas aplicações do modelo MLP para os MESFETs (para simulação da resposta transitória e pulsada do circuito porta lógica), naturalmente aponta para sua aplicação na análise de sensibilidade deste circuito. Neste caso, o circuito é analisado quando seus dispositivos internos mudam. Em particular, no projeto de circuitos integrados, imperfeições atribuídas no processo de fabricação destes dispositivos devem ser levadas em consideração. Além disso, um projetista às vezes deseja proceder com uma análise *what-if*, Pillage et al (1989).

Nesta seção, para as simulações do circuito porta lógica, adota-se o circuito equivalente completo para os MESFETs (sem o diodo de 'grampeamento'), Fig. 4.8(b). Desta forma, o circuito completo para a análise da porta lógica foi obtido, Fig. 4.15. As seguintes constantes foram empregadas na análise de sensibilidade: $R_D = R_S = 0,15\Omega/\mu\text{m}$, $R_G = 0,5\Omega/\mu\text{m}$, $C_{GD} = 0,06\text{ fF}/\mu\text{m}$, $C_{DS} = 0,2\text{ fF}/\mu\text{m}$, Curtice (1980).

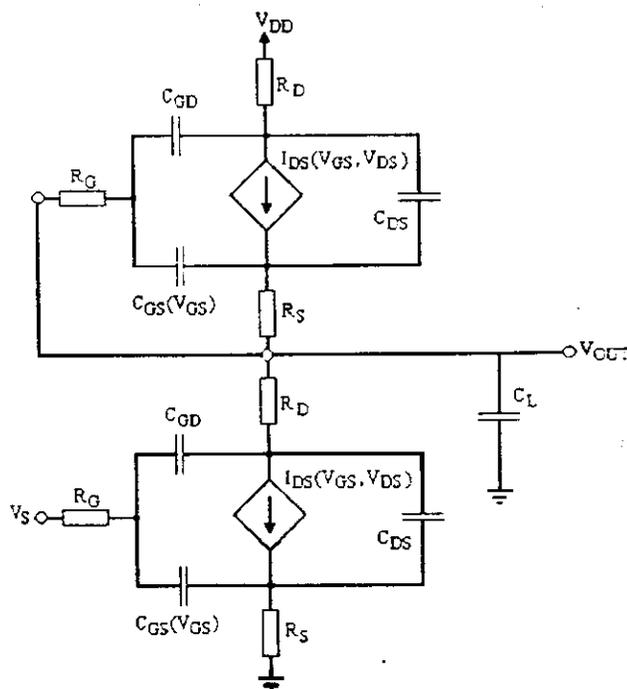


Fig. 4.15 Circuito equivalente para a porta lógica, considerando-se o circuito equivalente completo para os MESFETs.

A análise de sensibilidade pode ser realizada aplicando-se métodos diretos ou adjuntos, Pillage et al (1989). Uma alternativa é efetuar simulações repetidas do circuito, mudando os parâmetros de interesse a cada simulação. Contudo, a repetição da análise transitória é indesejável, devido ao custo computacional envolvido. Neste caso, a eficiência dos modelos MLPs desenvolvidos tornam interessante esta alternativa, Silva et al (2002).

Em particular, a mudança na resposta da porta lógica, quando as larguras dos MESFETs variam, foi investigada. Por exemplo, a mudança na resposta a uma entrada pulsada, de acordo com a largura do MESFET funcionado como carga não linear, Fig. 4.15, são apresentados na Fig. 4.16.

Quando empregam-se os modelos MLPs para computar as respostas da porta lógica, a eficiência da análise de sensibilidade é bastante melhorada, e, uma considerável redução do tempo total de computação pode ser observada.

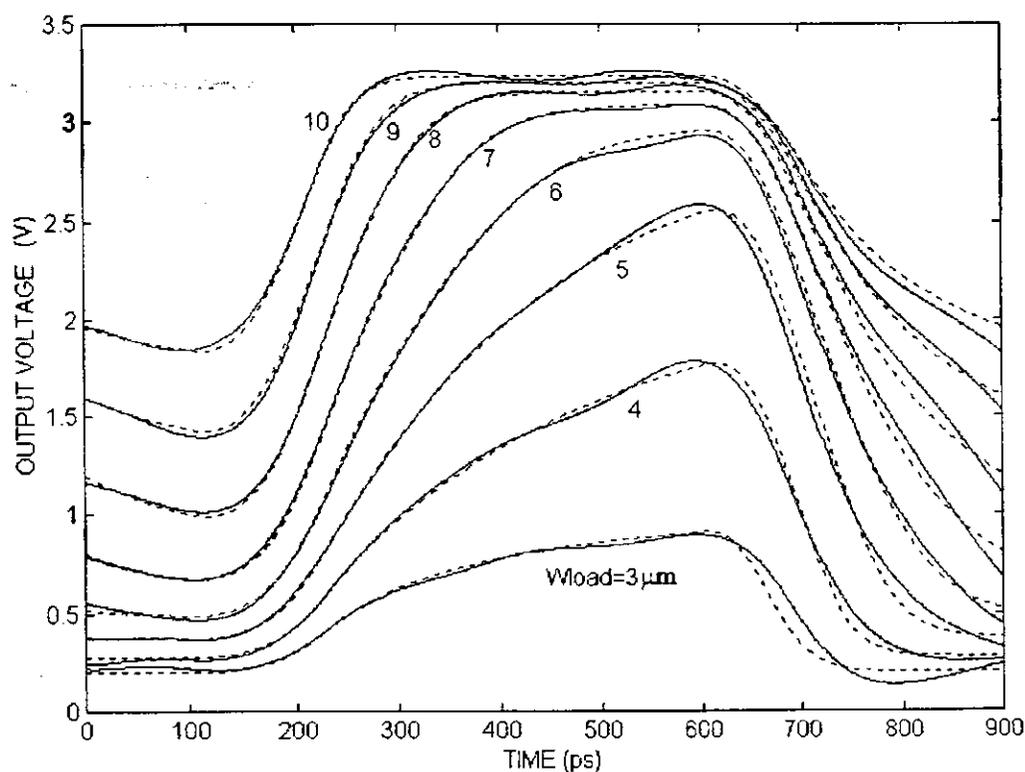


Fig. 4.16 Respostas pulsadas do circuito porta lógica para diferentes larguras do MESFET funcionado como carga não linear. (---) Respostas usando-se o modelo MLP para os transistores GaAs MESFETs. (—) Respostas usando-se um modelo MLP para o circuito porta lógica.

Capítulo 5

MÉTODO DO BALANÇO HARMÔNICO SIMPLIFICADO: REDES DE FOURIER

5.1 INTRODUÇÃO

A análise em regime de circuitos não lineares com respostas periódicas é um problema importante em projetos auxiliados por computador (CAD). Em princípio, a resposta periódica pode ser encontrada por integração das equações diferenciais, que descrevem o circuito, até que a resposta transitória torne-se desprezível. Contudo, esta aproximação por “força-bruta” é dispendiosa, principalmente quando os transitórios são governados por constantes de tempo, maiores que o período das fontes de alimentação.

Alguns métodos têm sido reportados para solucionar este problema. Entre outros, destacam-se: os métodos que usam séries de potência e séries de Volterra, Rhyne et al (1988) e Hu et al (1989), descrevendo o circuito completo no domínio da frequência; Aprille e Trick (1972) lidam com a solução de um problema de valor de contorno em dois pontos. A idéia neste caso é encontrar o vetor solução inicial $\mathbf{x}(0)$, tal que, quando integre-se o sistema de equações $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ sobre um período completo T , resulta o vetor de estados $\mathbf{x}(T) = \mathbf{x}(0)$.

Neste capítulo, propõe-se uma variação do método do balanço harmônico (MBH). Os métodos de balanço harmônico são variações do método de Galerkin, aplicados a circuitos não lineares. No método de Galerkin, admite-se a solução a priori na forma de uma combinação linear de funções de base, cujos coeficientes são desconhecidos. Esta solução é substituída nas equações do circuito e os coeficientes são ajustados, de tal forma que, estas equações sejam satisfeitas, tão precisamente quanto possível.

Quando a forma da solução escolhida for uma série de Fourier (SF), este procedimento tem sido referido como balanço harmônico. A aplicação do MBH para análise de circuitos não lineares deve-se a Baily (1968) e Lindenlaub (1969). Nesta versão,

cada variável de estado do circuito é expandida numa série de Fourier, cujos coeficientes são ajustados através de uma rotina de otimização.

Uma versão do método do balanço harmônico foi apresentada por Nakla e Vlach (1976). Eles reduziram o número de variáveis a serem otimizadas, particionando o circuito (rede) em sub-redes lineares e não lineares. As sub-redes lineares são resolvidas no domínio da frequência, as não lineares, no domínio do tempo. Apenas as variáveis associadas com a conexão destas sub-redes precisam ser otimizadas. Eles chamaram a técnica resultante de balanço harmônico por partes. Neste caso, se uma parte considerável da rede for linear, a dimensão do problema de otimização é bastante reduzida.

Uma versão simplificada deste método é abordada neste capítulo, Silva e Melo (1999). Nesta versão, para análise das sub-redes lineares no domínio da frequência, emprega-se a propriedade: “A resposta de um sistema linear e invariante no tempo (LTI) a uma exponencial complexa é uma exponencial complexa”, Oppenheim e Willsky (1983). Com esta observação, descarta-se o uso da DFT (transformada discreta de Fourier) para análise em regime permanente senoidal das sub-redes lineares.

Considerando a rede de Fourier como uma máquina adaptativa, que calcula séries de Fourier, descreve-se o MBH do ponto de vista destas redes. Neste caso, cada rede de Fourier aproxima a resposta periódica de uma sub-rede não linear do circuito.

5.2 FUNDAMENTOS DO MBH

O método do balanço harmônico pode ser descrito nos três passos seguintes:

- A rede não linear sob consideração precisa ser dividida num número mínimo possível de sub-redes lineares e não lineares.
- Cada sub-rede linear é aumentada por uma fonte periódica na forma de uma SF.
- A solução das sub-redes lineares no domínio da frequência é obtida pelo uso de uma rotina de otimização, através da qual, os coeficientes das séries de Fourier são ajustados. O objetivo é satisfazer as equações de circuito das sub-redes lineares, bem como, das não lineares, tão precisamente quanto possível.

Para uma descrição clara, considere uma rede simples, que consiste de duas sub-redes S_1 e S_2 , Fig. 5.1(a). Sejam $v(t)$ e $i(t)$ as formas de onda de tensão e corrente, respectivamente, nos terminais A-B. Admita que a sub-rede S_1 não contém quaisquer

fontes dependentes de tensão ou de corrente na sub-rede S_2 . Separe as duas sub-redes, aumentando a sub-rede S_1 por uma fonte de corrente com forma de onda $i(t)$, Fig. 5.1(b). Se a rede resultante tem uma solução única para todas as suas tensões e correntes de ramos, estas serão idênticas àquelas da sub-rede S_1 , Fig. 5.1(a). A tensão nos terminais A-B da sub-rede aumentada é idêntica à tensão $v(t)$. Por outro lado, se aumentarmos a sub-rede S_2 por uma fonte de tensão $v(t)$, a forma de onda de corrente resultante nos terminais A'-B' será idêntica a $i(t)$.

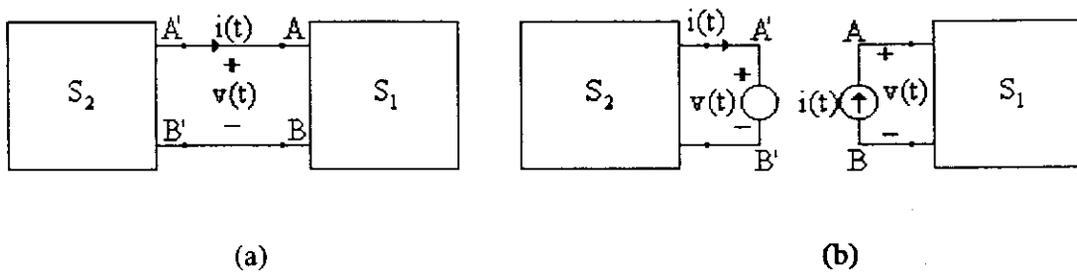


Fig. 5.1(a) Rede arbitrária. (b) A mesma rede separada em duas sub-redes, cada uma aumentada com fontes apropriadas.

O procedimento acima é repetido com algumas modificações. Em primeiro lugar, aumenta-se S_1 por uma fonte de corrente independente $i_1(t) \neq i(t)$. A tensão resultante nos terminais A-B foi denotada por $v_1(t)$. Se a sub-rede S_2 for aumentada por uma fonte de tensão $v_2(t) = v_1(t)$, alguma corrente $i_2(t)$ fluirá nos terminais A'-B'. Obviamente, $i_2(t) \neq i_1(t)$. Aqui, define-se uma função erro $e(t) = i_2(t) - i_1(t)$ e efetua-se um conjunto de alterações com o objetivo de minimizar este erro num período. Neste contexto, a fonte $i_1(t)$ que reduz este erro a zero é idêntica à corrente $i(t)$.

A idéia básica descrita acima pode ser generalizada para redes com qualquer número de portas. Seja uma rede arbitrária S com entradas periódicas de período T . O procedimento é iniciado através da divisão da rede S em um número mínimo possível de sub-redes lineares e não lineares. Em seguida, aumenta-se as sub-redes lineares com fontes de correntes periódicas, como uma expansão em série de Fourier:

$$i_{SF}(t) = \sum_{m=0}^M a_m \cos(m\omega t) + \sum_{m=1}^M b_m \text{sen}(m\omega t) \tag{5.1}$$

em que, a_m e b_m são os coeficientes da série e $\omega = 2\pi/T$. Sob esta forma de excitação, as tensões nos terminais das sub-redes lineares são determinadas no domínio da frequência,

$$v(t) = \sum_{m=0}^M v_m(t) \quad (5.2)$$

em que, $v_m(t)$ é a tensão que resulta da aplicação de fontes com frequência $m\omega$.

Em seguida, aplicam-se às sub-redes não lineares fontes de tensão com formas de onda idênticas àquelas dos terminais das sub-redes lineares. Em geral, as correntes terminais resultantes $i_N(t)$ serão funções não lineares de $v(t)$, ou seja,

$$i_N(t) = f(v) \quad (5.3)$$

Para uma solução consistente,

$$i_N(t) = i_{SF}(t), \quad 0 \leq t \leq T \quad (5.4)$$

O problema é escolher coeficientes em (5.1), tal que, (5.4) seja satisfeita. Com este objetivo, define-se a seguinte função custo, Nakla e Vlach (1976):

$$E(a_m, b_m) = \int_0^T e(t)e(t)dt, \quad e(t) = i_{SF}(t) - i_N(t) \quad (5.5)$$

5.3 MBH SIMPLIFICADO E REDES DE FOURIER

Na literatura, tem sido reportado o uso do MBH em conjunto com a DFT para a análise das sub-redes lineares, Gilmore e Steer (1991) e Rizzoli et al (1992). Neste estudo, contudo, procede-se com uma análise direta. Para isto, considera-se as sub-redes lineares como sistemas LTI, e, supomos fontes de alimentação DC ou puramente senoidais. Com estas restrições, a seguinte propriedade é empregada, Oppenheim e Willsky (1983):

$$x(t) = \exp(st) \quad \rightarrow \quad y(t) = H(s) \exp(st) \quad (5.6)$$

em que, $H(s)$ é a transformada de Laplace da resposta ao impulso do sistema LTI. No cálculo das respostas periódicas das sub-redes lineares ao seno e cosseno em (5.2), usa-se,

$$a_m \cos(m\omega t) \quad \rightarrow \quad a_m |H(m\omega)| \cos[m\omega t + \angle H(m\omega)] \quad (5.7)$$

$$b_m \sin(m\omega t) \quad \rightarrow \quad b_m |H(m\omega)| \sin[m\omega t + \angle H(m\omega)] \quad (5.8)$$

A resposta para uma fonte DC é um caso particular.

Como foi visto, no MBH as sub-redes lineares são aumentadas por fontes de correntes expressas na forma de séries de Fourier. Considerando-se a rede de Fourier como uma máquina adaptativa que calcula séries de Fourier, pode-se associar a cada uma das fontes de corrente uma rede de Fourier, Fig. 2.9. Portanto, um circuito analisado através do MBH simplificado usa uma rede de Fourier para cada componente não linear. A seguir, através de aplicações a circuitos simples, verificar-se-á a validade deste método através das redes de Fourier.

5.4 APLICAÇÕES DO MBH SIMPLIFICADO

Inicialmente, o MBH simplificado foi aplicado a um circuito retificador de $\frac{1}{2}$ onda, com filtro RC, Fig. 5.2(a), para o qual, a fonte de alimentação é senoidal:

$$v_s(t) = V_p \sin(\omega t) \quad (5.9)$$

A Fig. 5.2(b) mostra o modelo utilizado para o diodo, consistindo do capacitor da junção PN e do diodo ideal em paralelo, cuja corrente é uma função não linear da tensão, sendo dada por Mammanna e Zuffo (1973),

$$i_D(t) = f(x_1) = I_{SAT} (\exp(qx_1 / kT) - 1) \quad (5.10)$$

A sub-rede linear é aumentada pela fonte de corrente dada na forma (5.1), Fig. 5.2(c).

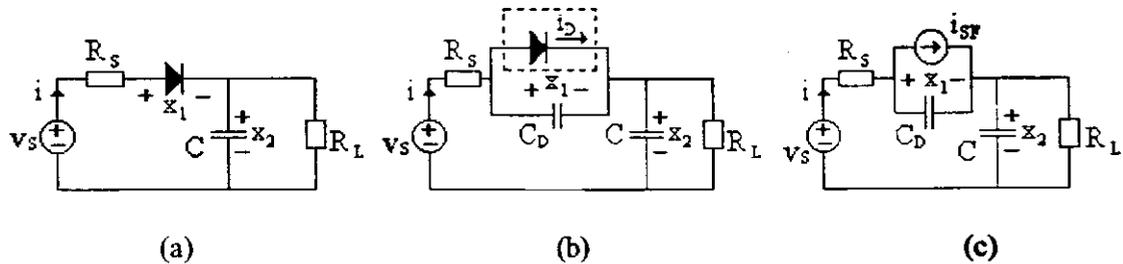


Fig. 5.2 (a) Circuito retificador 1/2 onda com filtro RC. (b) Inclusão do modelo para o diodo e destaque para a sub-rede não linear. (c) Sub-rede linear aumentada com a fonte $i_{SF}(t)$.

Dados: $R_s = 5\Omega$, $R_L = 1K\Omega$, $C = 1000\mu F$, $C_D = 1\mu F$, $I_{SAT} = 1\mu A$, $T = 290K$.

Com o emprego das leis de Kirchoff e da transformada de Laplace, obtém-se o seguinte sistema de equações para o circuito da Fig. 5.2(c), no tempo e na frequência:

$$\left\{ \begin{array}{l} \dot{x}_1(t) = \frac{i(t) - i_{SF}(t)}{C_D} \\ \dot{x}_2(t) = \frac{i(t) - x_2(t)/R_L}{C} \\ v_s(t) - R_s i(t) - x_1(t) - x_2(t) = 0 \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} sC_D X_1(s) = I(s) - I_{SF}(s) \\ sC X_2(s) = I(s) - X_2(s)/R_L \\ V_s(s) - R_s I(s) - X_1(s) - X_2(s) = 0 \end{array} \right. \quad (5.11)$$

Com a solução de (5.11) para $X_1(s)$, obtém-se,

$$X_1(s) = H(s)V_s(s) - Z(s)I_{SF}(s) \quad (5.12)$$

$$H(s) = 1/[k_2(s) + sC_D k_1(s)] \quad (5.13)$$

$$Z(s) = [R_s + k_1(s)]/[k_2(s) + sC_D k_1(s)] \quad (5.14)$$

$$k_1(s) = \frac{R_L}{1 + sR_L C} \quad (5.15)$$

$$k_2(s) = 1 + sR_s C_D \quad (5.16)$$

A tensão sobre o diodo resulta da contribuição de duas fontes, v_s e i_{SF} , (5.12). A partir de (5.7) e (5.8), obtém-se esta tensão diretamente no domínio do tempo,

$$x_1(t) = |H(\omega)|V_p \text{sen}[\omega t + \angle H(\omega)] - \sum_{m=0}^M |Z(m\omega)|a_m \cos[m\omega t + \angle Z(m\omega)] - \sum_{m=1}^M |Z(m\omega)|b_m \text{sen}[m\omega t + \angle Z(m\omega)] \quad (5.17)$$

Terminada a análise da sub-rede linear, tem-se um problema de otimização sem restrições, ou seja: dada uma solução inicial para os coeficientes de (5.1), minimize (5.5). Inicialmente, empregou-se o método do gradiente para este propósito. Contudo, este método mostrou-se ineficiente, apresentando oscilações, divergindo ou levando longos períodos de computação para aproximar o mínimo de (5.5).

Constatada estas limitações na utilização do método do gradiente, o método de Fletcher e Powell (1963) foi utilizado. Este método mostrou-se estável e bastante rápido, com um desempenho superior ao método do gradiente. A cada iteração deste método, no cálculo do mínimo de (5.5), procedimentos numéricos de estimação, extrapolação e interpolação são efetuados na pesquisa da amplitude de passo ótima para o ajuste dos parâmetros da rede Fourier, (APÊNDICE). Isto aumenta o custo computacional a cada iteração, contudo, verifica-se que o procedimento de minimização como um todo é acelerado.

De forma geral, a aplicação de métodos de otimização requer o cálculo do gradiente de (5.5). Em seguida estas expressões são formuladas. Para as derivadas de E em relação aos coeficientes a_m , resulta:

$$\frac{\partial E}{\partial a_m} = 2 \int_0^T e(t) \frac{\partial e(t)}{\partial a_m} dt \quad (5.18)$$

$$\frac{\partial e(t)}{\partial a_m} = \frac{\partial}{\partial a_m} [i_{SF}(t) - i_N(t)] = \cos(m\omega t) - \frac{\partial f(x_1)}{\partial x_1} \frac{\partial x_1}{\partial a_m} \quad (5.19)$$

$$\frac{\partial f(x_1)}{\partial x_1} = I_{SAT} \alpha_D \exp(\alpha_D x_1) \quad (5.20)$$

$$\frac{\partial x_1}{\partial a_m} = -|Z(m\omega)| \cos[m\omega t + \angle Z(m\omega)] \quad (5.21)$$

Da mesma forma, para as derivadas de E em relação aos coeficientes b_m ,

$$\frac{\partial P}{\partial b_k} = 2 \int_0^T \varepsilon(t) \frac{\partial \varepsilon(t)}{\partial b_k} dt \tag{5.22}$$

$$\frac{\partial e(t)}{\partial b_m} = \text{sen}(m\omega t) + I_{SAT} \alpha_D \exp(\alpha_D x_1) |Z(m\omega)| \text{sen}[m\omega t + \angle Z(m\omega)] \tag{5.23}$$

A regra de Simpson, Gerald (1970), foi utilizada no cálculo numérico das integrais (5.5), (5.18) e (5.22). Em seguida, apresenta-se a solução obtida para o circuito retificador, Fig. 5.2. com $v_s(t) = 5 \text{ sen}(120\pi t)$. Foram usadas 15 harmônicas e 400 amostras nesta simulação. O valor final da função custo foi $E = 2,39 \times 10^{-9}$. A Fig. 5.3(a) mostra a aproximação da corrente não linear no diodo. A partir da forma de onda de tensão na carga obtida, Fig. 5.3(b), observa-se a carga e a descarga do capacitor C , a menos das ondulações harmônicas. Convém ressaltar que, esta forma de onda é apenas uma aproximação da resposta exponencial esperada. Para uma forma de onda mais precisa, um maior número de harmônicas deve ser empregado.

Num segundo exemplo de aplicação, considera-se o circuito chave eletrônica com um BJT, Fig. 5.4(a). Com a inclusão do modelo de Ebers-Moll para o BJT, resulta um circuito com dois elementos não lineares, Fig 5.4(b). A sub-rede linear é aumentada por duas fontes de correntes na forma de séries de Fourier, Fig. 5.4(c). As equações que descrevem este circuito são quatro, duas de malha e duas do modelo de Ebers-Moll,

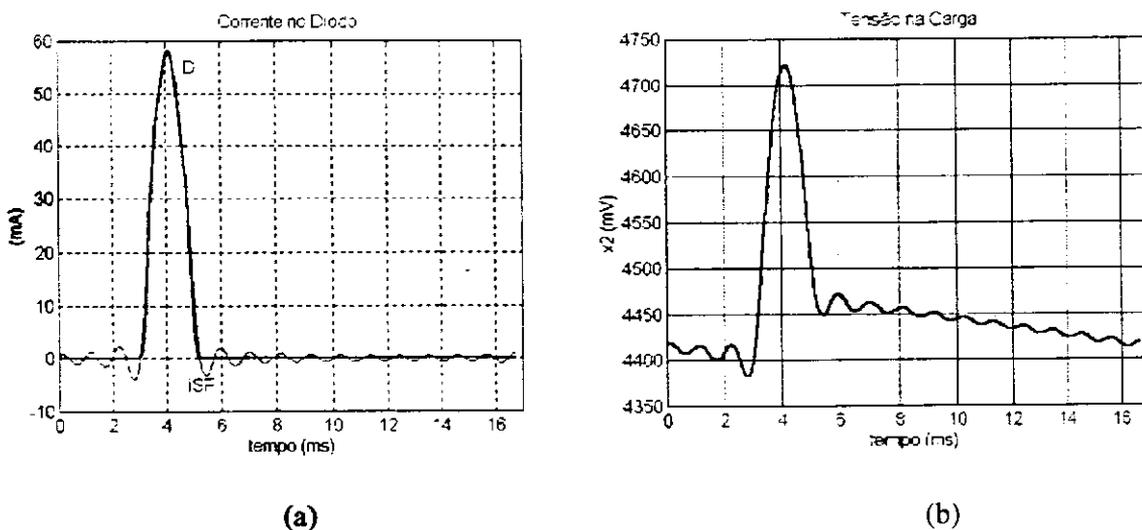


Fig. 5.3 (a) Aproximação através do MBH para a corrente no diodo. (b) Tensão na carga.

$$v_{BE}(t) = v_S(t) - R_B [i_{SF_e}(t) - i_{SF_c}(t)] \tag{5.24}$$

$$v_{BC}(t) = R_C i_{SF_c}(t) + v_{BE}(t) - V_{cc} \tag{5.25}$$

$$i_E(t) = i_F(v_{BE}) - \alpha_R i_R(v_{BC}) \tag{5.26}$$

$$i_C(t) = \alpha_F i_F(v_{BE}) - i_R(v_{BC}) \tag{5.27}$$

respectivamente, em que, $v_S(t)$ é uma fonte de tensão senoidal, (5.9); as correntes i_F e i_R nas junções do BJT são consideradas exponenciais na forma (5.10).

Tendo sido o problema devidamente posicionado, com o emprego do modelo para o BJT e com a divisão do circuito não linear resultante, os quatro coeficientes das séries (5.28) e (5.29) são os parâmetros livres das redes de Fourier:

$$i_{SF_e}(t) = \sum_{m=0}^M a_m \cos(m\omega t) + \sum_{m=1}^M b_m \sin(m\omega t) \tag{5.28}$$

$$i_{SF_c}(t) = \sum_{m=0}^M c_m \cos(m\omega t) + \sum_{m=1}^M d_m \sin(m\omega t) \tag{5.29}$$

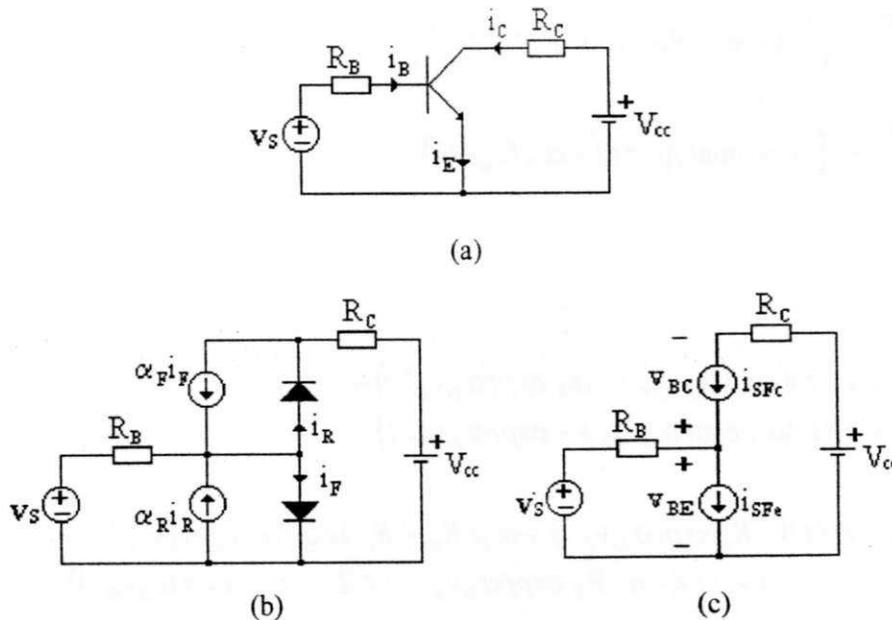


Fig. 5.4 (a) Chave eletrônica a transistor. (b) Inclusão do modelo de Ebers-Moll. (c) Sub-rede linear aumentada. Dados: $R_B = 1K\Omega$, $R_C = 10K\Omega$, $\alpha_F = 0,99$ e $\alpha_R = 0,5$, $V_{cc} = 10V$.

para a obtenção da resposta em regime. Para isto, emprega-se o método de Fletcher-Powell para minimizar o funcional (5.5) modificado. Ou seja, levando-se em conta os dois elementos não lineares, resulta:

$$E(a_m, b_m, c_m, d_m) = \frac{1}{2} \int_0^T [e_e(t)e_e(t) + e_c(t)e_c(t)] dt \quad (5.30)$$

em que,

$$e_e(t) = i_{SF_e}(t) - i_E(t) \quad (5.31)$$

$$e_c(t) = i_{SF_c}(t) - i_C(t) \quad (5.32)$$

Com as expressões (5.24)-(5.32) são obtidas as derivadas de (5.30) em relação aos coeficientes das séries de Fourier, que são necessárias à rotina de otimização:

$$\frac{\partial E}{\partial a_m} = \int_0^T \cos(m\omega t) [e_e(t) + \alpha_D I_{SAT} R_B \Theta_1] \quad (5.33)$$

$$\frac{\partial E}{\partial b_m} = \int_0^T \text{sen}(m\omega t) [e_e(t) + \alpha_D I_{SAT} R_B \Theta_1] \quad (5.34)$$

$$\frac{\partial E}{\partial c_m} = \int_0^T \cos(m\omega t) [e_c(t) + \alpha_D I_{SAT} \Theta_2] \quad (5.35)$$

$$\frac{\partial E}{\partial d_m} = \int_0^T \text{sen}(m\omega t) [e_c(t) + \alpha_D I_{SAT} \Theta_2] \quad (5.36)$$

em que,

$$\Theta_1 = e_e(t) (\exp(\alpha_D v_{BE}) - \alpha_R \exp(\alpha_D v_{BC})) + e_c(t) (\alpha_F \exp(\alpha_D v_{BE}) - \exp(\alpha_D v_{BC})) \quad (5.37)$$

$$\Theta_2 = e_e(t) (-R_B \exp(\alpha_D v_{BE}) + \alpha_R (R_B + R_C) \exp(\alpha_D v_{BC})) + e_c(t) (-\alpha_F R_B \exp(\alpha_D v_{BE}) + (R_B + R_C) \exp(\alpha_D v_{BC})) \quad (5.38)$$

A Fig. 5.5 mostra as formas de onda obtidas para resposta em regime da chave eletrônica.

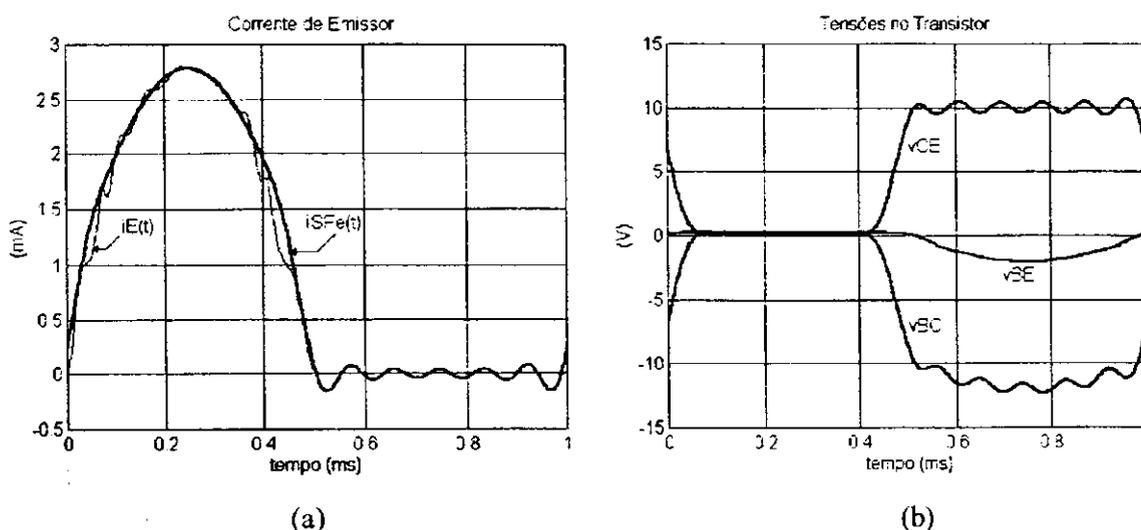


Fig. 5.5 Resposta em regime do circuito chave eletrônico a transistor para a entrada $v_S(t) = 2 \text{ sen}(2\pi \times 10^3 t)$. (a) Aproximação para a corrente de emissor. (b) Tensões no transistor.

Neste exemplo, o MBH simplificado foi iniciado com $M = 3$ e 8 amostras. A convergência ocorreu para $M = 12$ e 32 amostras, resultando $E = 3,05 \times 10^{-11}$.

Num terceiro exemplo, empregou-se o método desenvolvido para simular um amplificador classe C com uma entrada em 100 MHz, Fig. 5.6. O mesmo circuito foi analisado por Nackla e Vlach (1976). A resposta periódica foi obtida com $M = 9$ harmônicas, Fig. 5.7(a). A aproximação para a corrente de emissor $i_{SF_e}(t)$ e a corrente resultante na sub-rede não linear $i_E(t)$, obtida do modelo de Ebers-Moll. Pode-se verificar a partir da Fig. 5.7(c), que a primeira harmônica de $i_{SF_e}(t)$ é a principal componente na aproximação da corrente não linear $i_E(t)$. Isto ocorre em decorrência dos filtros presentes na sub-rede linear, sintonizados na frequência de operação.

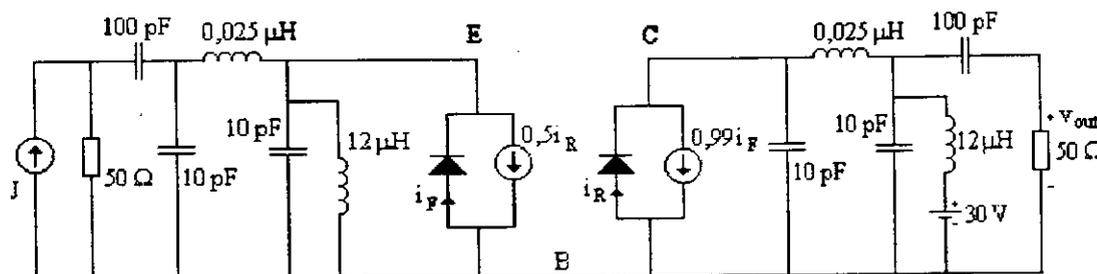
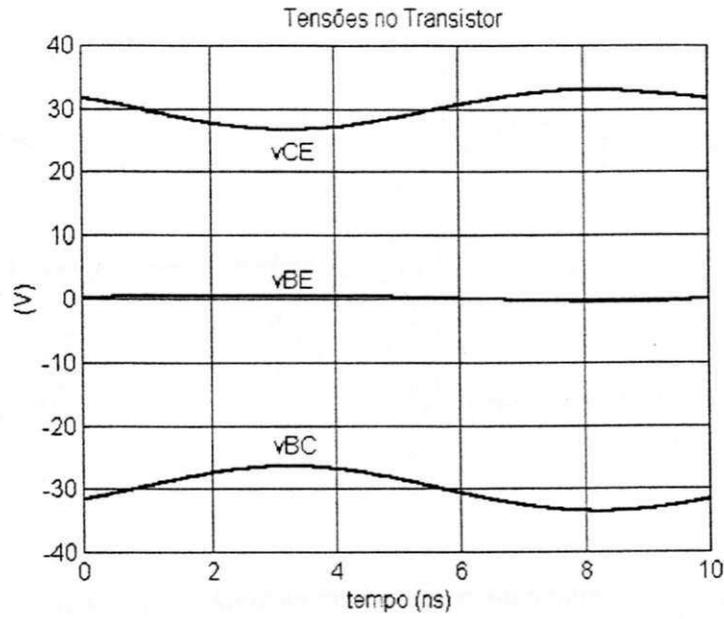
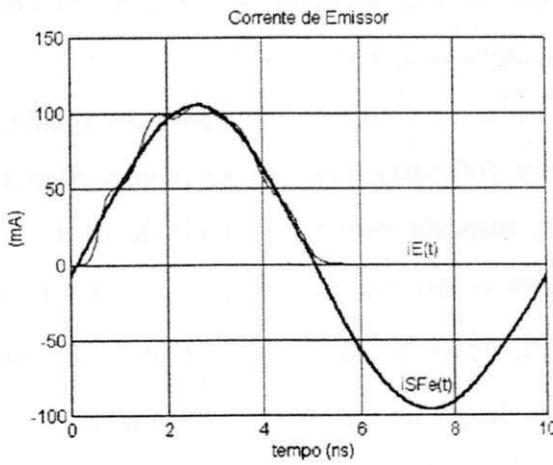


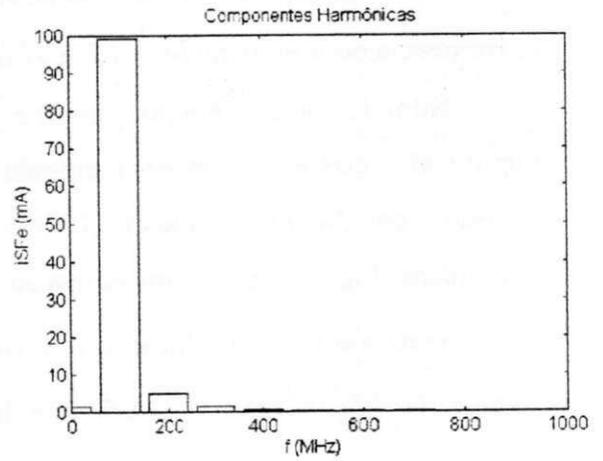
Fig. 5.6 Amplificador classe C com entrada $J(t) = 0,1 \text{ sen}(2\pi \times 10^8 t)$.



(a)



(b)



(c)

Fig. 5.7 Resposta em regime para o amplificador classe C. (a) Tensões no transistor. (b) Corrente de emissor. (c) Componentes harmônicas da solução em regime.

Capítulo 6

MODELOS ATRAVÉS DE REDES NEURAIAS BASEADAS NO CONHECIMENTO EMPÍRICO PARA CIRCUITOS DE RF/MICROONDAS

6.1 INTRODUÇÃO

Neste capítulo, alguns métodos para a modelagem de circuitos de microondas através de redes neurais artificiais baseadas conhecimento empírico são descritos. A habilidade das ANNs para modelar problemas não lineares com várias variáveis é explorado na implementação do conceito de SM. Aproveitando o conjunto numeroso de modelos empíricos existentes para circuitos de RF/microondas, o mapeamento de espaços baseado em redes neurais (SM-ANN) diminui o número de simulações EMs para treinamento, aumenta a habilidade de generalização e reduz a complexidade da topologia da rede neural.

Métodos inovadores, através da inclusão de modelos empíricos, têm sido propostos para solucionar problemas de modelagem na área de microondas, Rayas-Sanchez (2001), Zhang e Gupta (2000). Os principais objetivos do uso de modelos empíricos em conjunto com ANNs são: a redução do conjunto de treinamento e a melhoria da capacidade de generalização das redes neurais. Algumas destas técnicas, descritas neste capítulo, são relacionadas abaixo:

- Método Híbrido EM-ANN;
- Método PKI – Entradas Conhecidas a Prior;
- Método KBNN – Rede Neural Baseada no Conhecimento;
- Método SM-ANN – Mapeamento de Espaços Baseado em Rede Neural.

Uma técnica nova é proposta para realizar o método SM-ANN. Através de simulações, esta e outras técnicas são verificadas para o caso de uma junção-T em microfita.

6.2 MÉTODOS DE MODELAGEM ATRAVÉS DE REDES NEURAIIS PARA CIRCUITOS DE RF/MICROONDAS

As redes neurais artificiais são apropriadas como modelos de dispositivos não lineares encontrados na área de RF/microondas. Isto deve-se, em parte, a sua habilidade de generalizar a partir de exemplos, sua natureza não linear e sua estrutura paralela, Zhang e Gupta (2000).

Tem sido demonstrado por White et al (1992), que as redes MLPs podem aproximar qualquer função em qualquer nível de precisão desejado, desde que exista uma relação determinística entre suas entradas e saídas. Conforme observado por Haykin (1999), redes neurais com poucos parâmetros não conseguem aproximar adequadamente uma relação entrada/saída, enquanto aquelas com muitos parâmetros internos, realizam corretamente esta relação para os exemplos de treinamento, mas, sua capacidade de generalização é ruim.

Do ponto de vista computacional, os modelos através de redes neurais são mais velozes que os modelos EMs ou físicos; sua precisão é maior que a dos modelos empíricos. As ANNs são modelos apropriados para otimização de circuitos de microondas e projeto estatístico, como demonstrado por Zaabab et al (1995), bem como, por Burrascano et al (1998).

Em seguida, descreve-se sucintamente alguns métodos através de ANNs baseadas no conhecimento empírico nas áreas de RF e microondas.

6.2.1 Método Convencional

Na aproximação convencional desta técnica neuro-computacional, uma rede neural é treinada, de tal modo que, para uma dada frequência, ω , e parâmetros do projeto. x_f , a resposta do modelo preciso, R_f , é aproximada numa dada região de interesse, Fig. 6.1. O vetor w contém os parâmetros internos da ANN. Se a rede neural for treinada com um número suficiente de exemplos de treinamentos (isto é, quando for encontrado o valor ótimo para w), a ANN serve como um modelo rápido e preciso na região de interesse. O número de parâmetros internos da rede neural deve ser selecionado, de tal forma que: se obtenha um erro de treinamento reduzido, bem como, uma generalização desejada para a região de interesse.

O aprendizado de uma ANN é um problema onde seus parâmetros são otimizados, até que, a aproximação para os exemplos de treinamento seja considerada adequada. O número elevado de exemplos, necessários para assegurar a precisão do modelo, é o principal obstáculo da aproximação convencional. Para circuitos de RF/microondas, dados precisos são obtidos através de simulações EM/medidas. Por outro lado, as combinações de parâmetros geométricos, materiais e sinais de entrada, na definição da região de interesse, aumenta ainda mais este custo computacional.

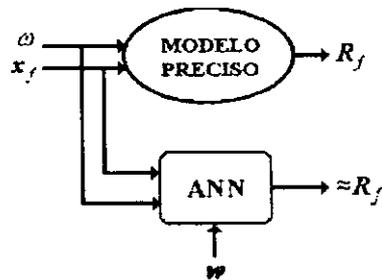


Fig. 6.1 Aproximação convencional de um modelo através de rede neural.

6.2.2 Método Híbrido EM-ANN

Proposto por Watson et al (1998), o modelo híbrido EM-ANN emprega a diferença entre os parâmetros de espalhamento dos modelos empírico e preciso para treinar a rede neural, como ilustrado na Fig. 6.3. Devido a relação simplificada que a ANN precisa mapear, o número de simulações do modelo preciso é reduzido. O método híbrido EM-ANN é chamado também de método da diferença.

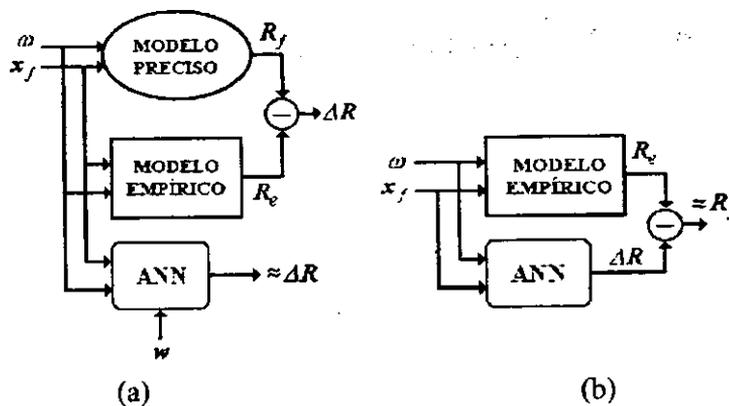


Fig. 6.2 (a) Ilustração do método EM-ANN. Modelo híbrido EM-ANN.

6.2.3 Método PKI

No método PKI, além dos parâmetros físicos e da frequência, a saída do modelo empírico também é utilizada como mais uma entrada para a ANN, Watson et al (1998), Zhang e Gupta (2000), Fig. 6.3. A rede neural é treinada, tal que, a sua resposta seja uma aproximação da resposta do modelo preciso, para todo o conjunto de treinamento. De acordo com Watson et al (1998), a aproximação PKI exibe melhor precisão que a aproximação EM-ANN, mas, requer uma rede neural com maior número de parâmetros.

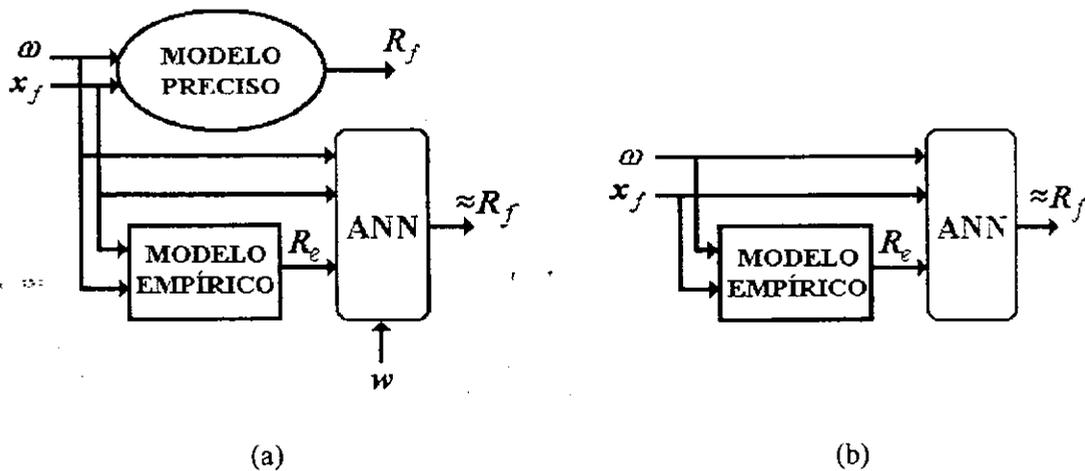


Fig. 6.3 (a) Ilustração do método PKI. (b) Modelo PKI.

6.2.4 Método KBNN

Na aproximação através de rede neural baseada no conhecimento (KBNN), desenvolvida por Wang e Zhang (1997), a informação empírica ou semi-analítica é incorporada na estrutura interna da ANN. As redes neurais baseadas no conhecimento são parcialmente conectadas. As camadas ocultas são divididas em duas partes através da definição de funções empíricas e de funções de ativação de uma ANN, como ilustrado na Fig. 6.4. As funções empíricas são obtidas de modelos empíricos existentes, geralmente, com base em aproximações quase-estáticas de circuitos equivalentes de estruturas de microondas.

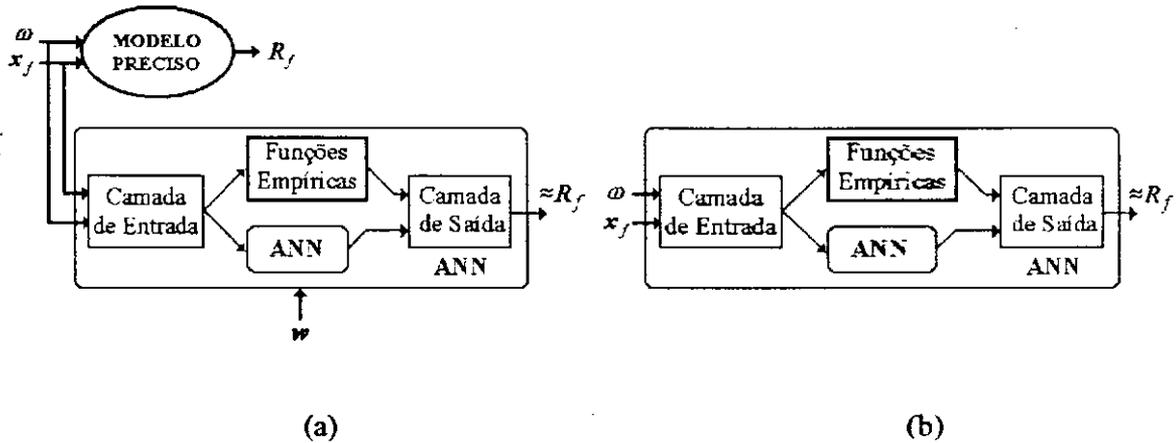


Fig. 6.4 (a) Ilustração do método KBNN. (b) Modelo KBNN.

Com o conhecimento empírico através de fórmulas de microondas incorporado na estrutura da rede neural, estas podem ser refinadas (ou ajustadas) como parte do processo de treinamento. Desde que, ao invés das funções de ativação padrões, funções empíricas são usadas para algumas camadas de neurônios da KBNN, o treinamento do MLP não segue o algoritmo padrão da *backpropagation*, Wang (1998), Zhang e Gupta (2000).

6.2.5 Método SMN

O mapeamento de espaços é um conceito novo para projeto e otimização de circuitos, que combina a eficiência computacional de um modelo empírico com a exatidão de um modelo preciso. Em geral, os modelos empíricos são eficientes do ponto de vista computacional, mas suas respostas são válidas apenas para uma faixa limitada do seu espaço de parâmetros, fora da qual, seus resultados são imprecisos. Por outro lado, modelos precisos (detalhados) são disponíveis através de simuladores EMs, ou mesmo por medidas diretas: eles são muito precisos, porém sua aplicação resulta num esforço computacional elevado.

A técnica SM estabelece uma conexão entre os modelos preciso e empírico (aproximado), mantendo a eficiência do modelo empírico, enquanto preserva a exatidão do modelo preciso, Bandler et al (1994).

Por exemplo, define-se os vetores, \mathbf{x}_f e \mathbf{x}_e , de parâmetros de projeto dos modelos preciso e empírico, respectivamente; $R_f(\mathbf{x}_f)$ e $R_e(\mathbf{x}_e)$, suas respostas correspondentes. O cálculo da resposta $R_e(\mathbf{x}_e)$ é mais rápido, porém menos preciso que o da resposta $R_f(\mathbf{x}_f)$.

Como ilustrado na Fig. 6.5, o objetivo da otimização SM é encontrar um mapeamento apropriado, P , entre o espaço de parâmetros do modelo preciso, \mathbf{x}_f , e o espaço de parâmetros do modelo empírico, \mathbf{x}_e :

$$\mathbf{x}_e = P(\mathbf{x}_f) \tag{6.1}$$

tal que,

$$R_e(P(\mathbf{x}_f)) \approx R_f(\mathbf{x}_f) \tag{6.2}$$

Uma vez que o mapeamento para uma região de interesse for encontrado, o modelo empírico pode ser empregado para simulações rápidas e precisas naquela região.

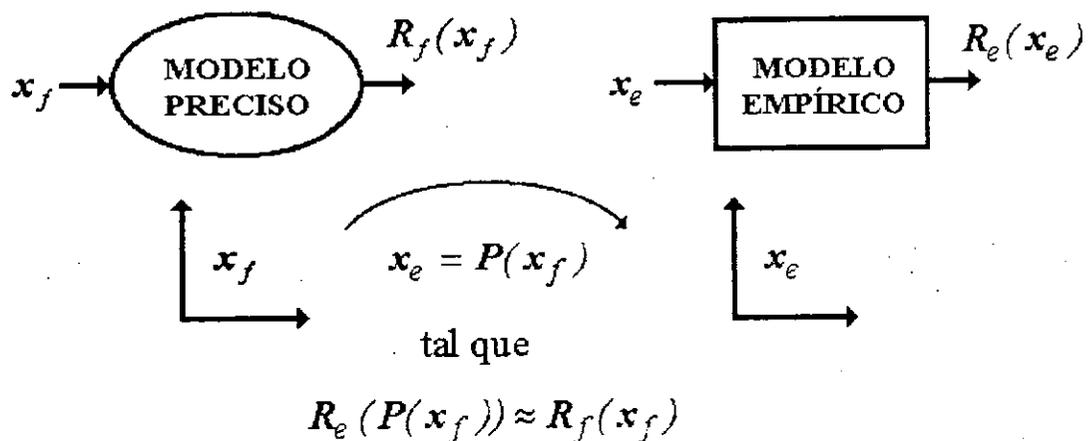


Fig. 6.5 Ilustração da técnica SM.

Na aproximação SMN, o mapeamento do espaço de parâmetros do modelo preciso para o modelo empírico é implementado por uma ANN, Fig. 6.6. O objetivo é encontrar o conjunto ótimo de parâmetros internos da ANN, tal que, a resposta do modelo empírico seja, tanto quanto possível, próxima da resposta do modelo preciso, para todos os exemplos de treinamento.

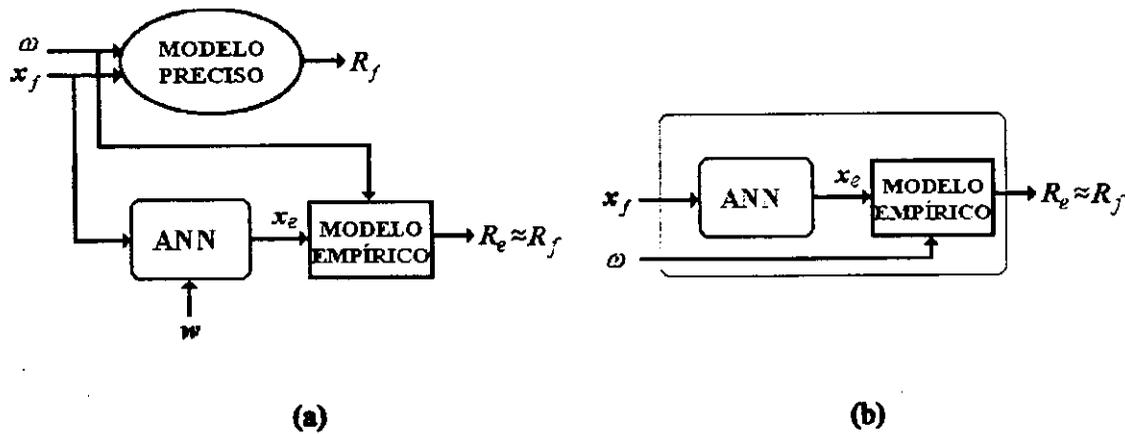


Fig. 6.6 (a) Ilustração do método SMN. (b) Modelo SMN.

O mapeamento desejado é aproximado, por exemplo, através da solução do problema de otimização:

$$\min_{\omega} \|e_1^T \quad e_2^T \quad \dots \quad e_M^T\|^T \quad (6.3)$$

em que, o vetor ω contém os parâmetros internos da rede neural, selecionados como variáveis de otimização; e_k é o vetor erro dado por:

$$e_k = R_f(x_{f_i}, \omega_j) - R_e(x_e, \omega_j) \quad (6.4)$$

$$x_e = P(x_{f_i}) \quad (6.5)$$

com,

$$i = 1, \dots, B_p \quad (6.6)$$

$$j = 1, \dots, F_p \quad (6.7)$$

$$k = j + F_p(i-1) \quad (6.8)$$

em que, B_p é o número de pontos base de treinamento para os parâmetros de entrada e F_p é o número de pontos de frequência por varredura de frequência. Portanto, o número total de exemplos de treinamento é $B_p F_p$. As características específicas de P dependem do paradigma ANN escolhido, cujos parâmetros internos estão em w .

Diferentes técnicas podem ser utilizadas para a solução de (6.3). A norma de Huber foi utilizada por Rayas-Sanchez (2001), explorando suas características robustas para interpolar dados, como demonstrado por Bandler et al (1993). Em Rayas-Sanchez (2001) o software *NeuroModeler* (1999) foi empregado no treinamento das ANNs. Aqui, contudo, utiliza-se um método próprio, que emprega a norma Euclidiana e o algoritmo Rprop para solucionar (6.3), Riedmiller e Braun (1993). Este último foi implementado no ambiente do Matlab™ (1998), para o treinamento das redes neurais MLP.

Quando o erro (6.4) no mapeamento através de uma ANN é menor que aquele estipulado na fase de treinamento; além disso, se a ANN for capaz de generalizar, então, o modelo SMN está pronto para uso em outras aplicações.

6.2.6 Incluindo a Frequência no Método SMN

Muitos dos modelos empíricos existentes para circuitos de microondas foram desenvolvidos baseados em métodos de análise quase-estática. Por exemplo, no caso de circuitos em microfita, assume-se o método de propagação de onda TEM, Gupta et al (1979). Isto implica que a constante dielétrica efetiva ϵ_r e a impedância característica Z_0 não variam com a frequência. Contudo, comportamentos não-TEM fazem com que ϵ_r e Z_0 sejam, de fato, funções da frequência. Portanto, estes modelos empíricos apresentam uma precisão boa apenas numa faixa de frequências limitada.

Para aumentar a faixa de frequências, na qual o modelo é válido, um método é estabelecer um mapeamento do modelo preciso para o empírico, que seja sensível à frequência. Para a ANN, que implementa o mapeamento, isto é possível se a frequência for uma variável extra de entrada. No restante desta seção, descreve-se os seguintes métodos de SMN, que incluem a frequência:

- i. FDSMN – Mapeamento de Espaço Dependente da Frequência Baseado em Redes Neurais
- ii. FSMN – Mapeamentos de Espaço e Frequência Baseados em Redes Neurais
- iii. FMN – Mapeamento da Frequência Baseado em Redes Neurais
- iv. FPSMN – Mapeamentos de Parte do Espaço e Frequência Baseados em Redes Neurais
- v. FDPSMN – Mapeamento Parcial do Espaço Dependente da Frequência Baseado em Redes Neurais

Na aproximação FDSMN, ilustrada na Fig. 6.7, os modelos preciso e empírico são simulados na mesma frequência, mas o mapeamento do espaço de parâmetros do modelo preciso para o empírico depende da frequência. O mapeamento é encontrado através da solução do problema de otimização expresso em (6.3), substituindo-se (6.4) e (6.5) por

$$e_k = R_f(x_{f_j}, \omega_j) - R_e(x_e, \omega_j) \tag{6.9}$$

$$x_e = P(x_{f_j}, \omega_j) \tag{6.10}$$

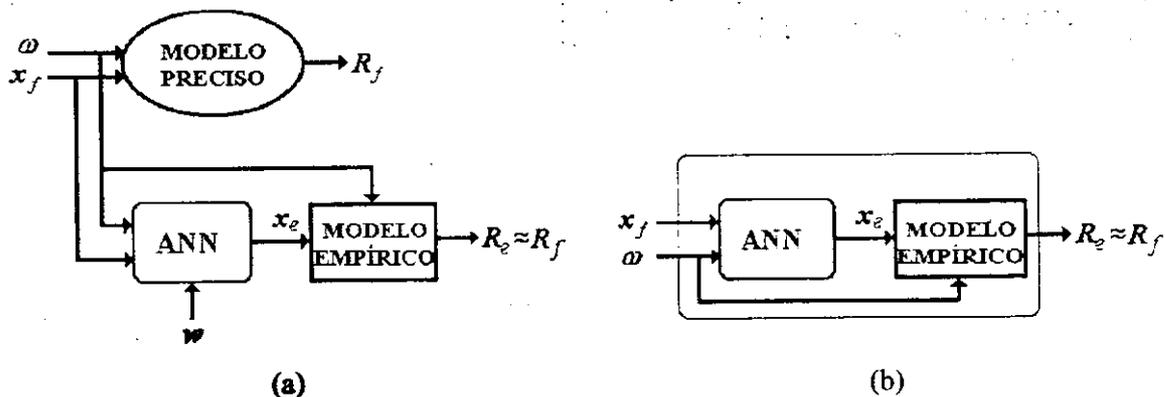


Fig. 6.7 (a) Ilustração do método FDSMN. (b) Modelo FDSMN.

A aproximação FSMN, estabelece um mapeamento não só dos parâmetros de projeto mas também da variável freqüência. Desta forma, a resposta do modelo preciso é aproximada por um modelo empírico simulado numa freqüência ω_e .

A técnica FSMN é realizada por meio da adição de uma saída extra para a ANN que efetua o mapeamento, Fig. 6.8. O mapeamento é encontrado através da solução do problema de otimização expresso em (6.3), substituindo-se (6.4) e (6.5) por

$$e_k = R_f(x_{f_i}, \omega_j) - R_e(x_e, \omega_j) \quad (6.11)$$

$$\begin{bmatrix} x_e \\ \omega_e \end{bmatrix} = P(x_{f_i}, \omega_j) \quad (6.12)$$

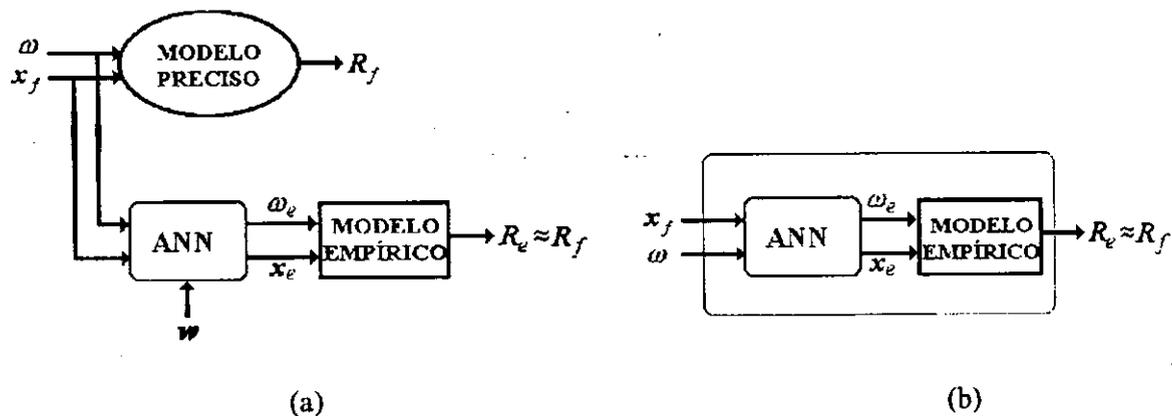


Fig. 6.8 (a) Ilustração do método FSMN. (b) Modelo FSMN.

É comum encontrar problemas de microondas onde o modelo de circuito equivalente comporta-se de forma similar ao modelo eletromagnético, mas com uma resposta deslocada em freqüência. Nestes casos, um alinhamento pode ser alcançado através da simulação do modelo empírico numa freqüência diferente daquela utilizada no modelo preciso. A aproximação FMN implementa esta estratégia, Fig. 6.9, com a simulação do modelo empírico com os mesmos parâmetros do modelo preciso, mas numa freqüência ω_e para alinhar as respostas.

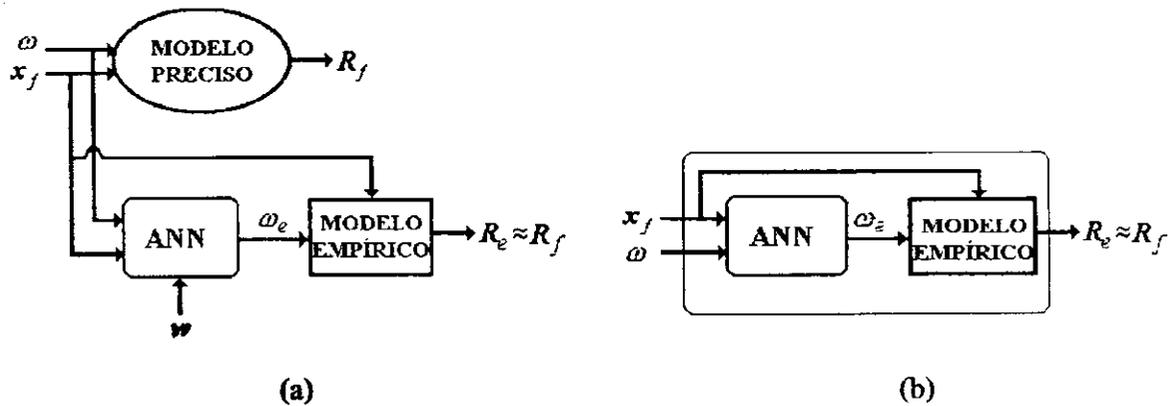


Fig. 6.9 (a) Ilustração do método FMN. (b) Modelo FMN.

Na técnica FMN, o mapeamento é encontrado através da solução do problema de otimização expresso em (6.3), mas substituindo (6.4) e (6.5) por

$$e_k = R_f(x_{f_i}, \omega_j) - R_e(x_{f_i}, \omega_e) \quad (6.13)$$

$$\omega_e = P(x_{f_i}, \omega_j) \quad (6.14)$$

O mapeamento de todos os parâmetros físicos de um modelo empírico, como nas técnicas SMN, FDSMN e FSMN, pode resultar em singularidades na sua resposta, para as quais o algoritmo de treinamento diverge. Este problema é contornado estabelecendo-se um mapeamento parcial para os parâmetros físicos, através do uso mais eficiente do conhecimento implícito no modelo empírico.

Em muitos problemas práticos na área de RF/microondas, o mapeamento de apenas uma parte dos parâmetros físicos pode ser suficiente para se obter uma precisão desejada. Isto permite uma significativa redução do número de parâmetros da rede neural, bem como, do tempo de treinamento. A aproximação FPSMN implementa esta técnica, Fig. 6.10.

A seleção dos parâmetros físicos a serem mapeados na técnica FPSMN pode ser realizado pelo usuário. Mesmo quando não há uma experiência, os parâmetros a serem mapeados no FPSMN podem ser determinados, com reduzido custo computacional, através da análise de sensibilidade do modelo empírico.

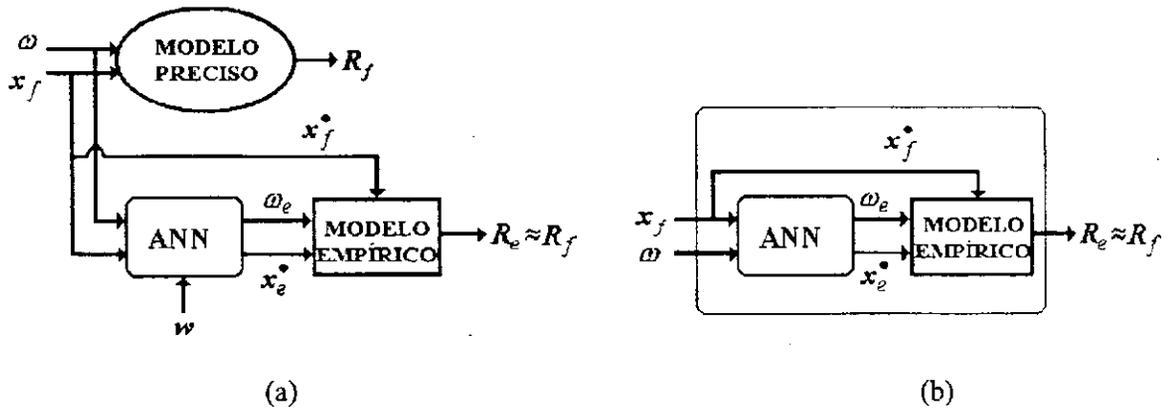


Fig. 6.10 (a) Ilustração do método FPSMN. (b) Modelo FPSMN.

Na técnica FPSMN, o mapeamento é encontrado através da solução do problema de otimização expresso em (6.3), substituindo-se (6.4) e (6.5) por

$$e_k = R_f(x_{f_i}, \omega_j) - R_e(x_{f_i}^*, x_e^*, \omega_e) \quad (6.15)$$

$$\begin{bmatrix} x_e^* \\ \omega_e \end{bmatrix} = P(x_{f_i}, \omega_j) \quad (6.16)$$

em que, $x_{f_i}^*$ é o vetor com o um subconjunto dos parâmetros físicos de projeto x_{f_i} no i -ésimo ponto base de treinamento. A parte $x_{f_i}^*$ dos parâmetros do modelo empírico, deve ser escolhida com base nas informações contidas no modelo empírico.

Finalmente, é natural que existam circuitos de RF/microondas onde o modelo empírico mapeado com uma simples rede neural não seja capaz de aproximar todas as respostas contidas em R_f . Nestes casos, a tarefa de aprendizado pode ser distribuída entre várias ANNs, que por sua vez divide o espaço de saída em diversos subespaços. As ANNs correspondentes podem então ser treinadas individualmente, para casar cada subconjunto da resposta contida em R_f . Isto implica na solução de vários problemas de otimização independentes, ao invés de apenas um.

6.2.7 Um Método SM-ANN Alternativo

Os métodos SM-ANN descritos foram utilizados para modelar os parâmetros de espalhamento de uma junção-T. As dificuldades para o treinamento das redes neurais, neste caso, motivaram o desenvolvimento de outra aproximação para o mapeamento de espaços com base em redes neurais.

Um método SM-ANN alternativo, diferente dos descritos anteriormente, foi desenvolvido tendo por base o FDSMN e o FPSMN. Esta aproximação é denominada FDPSMN (Mapeamento Parcial de Espaço Dependente da Frequência Baseado em Redes Neurais), Fig. 6.11. Na técnica FDPSMN, o mapeamento é encontrado através da solução do mesmo problema de otimização expresso em (6.3), mas substituindo (6.4) e (6.5) por

$$e_k = R_f(x_{f_i}, \omega_j) - R_e(x_{f_i}^*, x_{e_i}^*, \omega_j) \quad (6.17)$$

$$x_e^* = P(x_{f_i}, \omega_j) \quad (6.18)$$

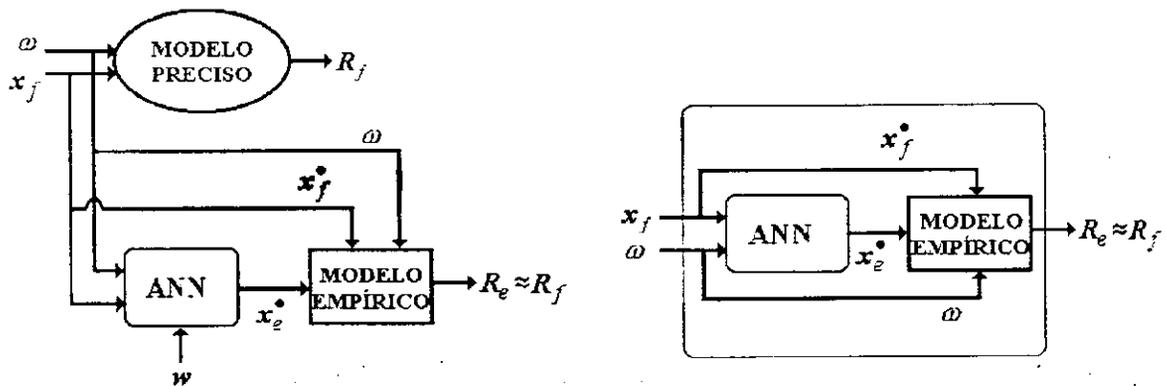


Fig. 6.11 (a) Ilustração do método FDPSMN. (b) Modelo FDPSMN.

Os resultados de simulação para a junção-T, apresentados no final deste capítulo, demonstram um melhor desempenho da técnica desenvolvida com relação aos outros métodos SM-ANN. Ganhos em estabilidade e convergência no treinamento das redes neurais podem ser observados com o emprego do FDPSMN.

6.3 MAPEAMENTO DE ESPAÇOS UTILIZANDO O MLP COM TRÊS CAMADAS

Nesta seção descreve-se as modificações efetuadas na rede MLP padrão, de tal forma que, este paradigma de rede neural torne-se mais adequado nas aplicações que fazem uso do conhecimento empírico em RF/microondas. Duas alterações são realizadas para o treinamento do MLP: o escalonamento de suas entradas e saídas; o mapeamento unitário como ponto de partida para o seu treinamento.

6.3.1 Exemplos de Treinamento e Iniciação do MLP

O ponto de partida para o problema de otimização expresso em (6.3) é o conjunto de parâmetros internos da ANN, denotado por $w^{(0)}$, que é escolhido assumindo-se que o modelo empírico seja o melhor modelo e, portanto, o mapeamento não se faz necessário, Rayas-Sanchez (2001). Em outras palavras, $w^{(0)}$ é escolhido, tal que, a ANN implementa um mapeamento unitário. Isto é válido para os seis métodos SM-ANN descritos anteriormente.

A rede neural é treinada para “aprender” o mapeamento P , numa região de interesse. A fim de manter-se um conjunto de treinamento reduzido, a escolha da complexidade da ANN torna-se crítica. Para se evitar uma generalização pobre, uma ANN mais simples deve ser utilizada – que leve a um erro de aprendizado aceitável dentro da região de treinamento, enquanto, apresenta uma generalização adequada para o conjunto de teste (Haykin, 1999).

Com o emprego de redes MLPs com três camadas, todos os seis métodos de mapeamento de espaços anteriores foram implementadas. A Fig. 6.12 ilustra a realização da aproximação SM-ANN, com o emprego de um MLP com h neurônios ocultos. A frequência ω foi omitida nesta configuração. Para as outras técnicas de mapeamento, que incluem a variável ω , este paradigma pode ser modificado apropriadamente através da adição ou subtração de entradas/saídas.

Para a rede MLP, Fig. 6.12, tem-se que: $x_f \in \mathcal{R}^n$ é o vetor de entradas (parâmetros físicos a serem mapeados); $v \in \mathcal{R}^n$ contém os sinais de entrada escalonados; $z \in \mathcal{R}^h$ é o vetor

contendo os sinais oriundos do h -ésimo neurônio oculto; $\mathbf{y} \in \mathfrak{R}^n$ é o vetor de sinais de saída antes do escalonamento; $\mathbf{x}_e \in \mathfrak{R}^n$ contém a saída do mapeamento neural.

A fim de controlar a importância relativa dos diferentes parâmetros de entrada e, ao mesmo tempo, definir uma faixa dinâmica para a região de interesse, um escalonamento das entradas é efetuado, tal que, $-1 \leq v_i \leq 1$,

$$v_i = -1 + \frac{2(x_{fi} - x_{fi_{min}})}{(x_{fi_{max}} - x_{fi_{min}})}, \quad i = 1, 2, \dots, n \quad (6.19)$$

Os sinais na camada oculta são dados por,

$$z_i = \varphi(b_i^h + \mathbf{v}^T \mathbf{w}_i^h), \quad i = 1, 2, \dots, h \quad (6.20)$$

em que, $\varphi(\cdot)$ é a função de ativação dos neurônios ocultos; \mathbf{w}_i^h é o vetor de pesos para o i -ésimo neurônio oculto; \mathbf{b}^h é o vetor com elementos de polarização,

$$\mathbf{w}_i^h = [w_{i1}^h \quad w_{i2}^h \quad \dots \quad w_{in}^h]^T, \quad i = 1, 2, \dots, h \quad (6.21)$$

$$\mathbf{b}^h = [b_1^h \quad b_2^h \quad \dots \quad b_h^h]^T \quad (6.22)$$

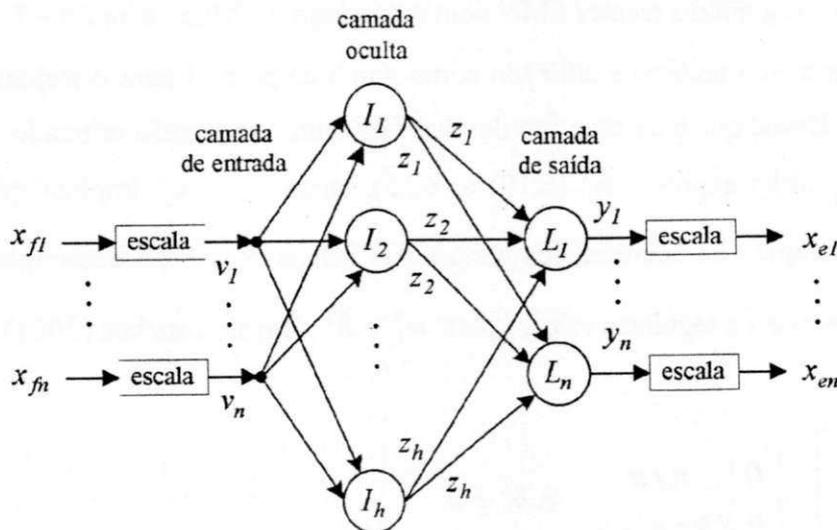


Fig. 6.12 Rede neural MLP com três camadas para mapeamento de espaços.

Os sinais na camada de saída são dados por

$$y_i = b_i^o + z^T w_i^o, \quad i = 1, 2, \dots, n \quad (6.23)$$

e b^o é o vetor com elementos de polarização para os neurônios da camada de saída,

$$b^o = [b_1^o \quad b_2^o \quad \dots \quad b_n^o]^T \quad (6.24)$$

Para proporcionar um escalonamento equivalente dos sinais de saída, para aquele, usado nos sinais de entrada, define-se:

$$x_{ei} = x_{f_{i \min}} + \frac{1}{2}(y_i + 1)(x_{f_{i \max}} - x_{f_{i \min}}), \quad i = 1, 2, \dots, n \quad (6.25)$$

O vetor w contendo o conjunto de todos os parâmetros internos da ANN tomados como variáveis de otimização para um MLP de três camadas é então definido como:

$$w = [(b^h)^T \quad (b^o)^T \quad (w_1^h)^T \quad \dots \quad (w_n^h)^T \quad (w_1^o)^T \quad \dots \quad (w_n^o)^T] \quad (6.26)$$

Através de uma inspeção em (6.26), o número total de variáveis de otimização para a solução de (6.3), seguindo a técnica SMN com o emprego do MLP, é $n(2h + 1) + h$.

O mapeamento unitário é utilizado como ponto de partida para o treinamento da ANN na técnica SM. Desde que o mesmo tipo de escalonamento está sendo utilizado na entrada e na saída da ANN, como expresso em (6.19) e (6.25), então $x_e = x_f$ implica que $y_i = v_i$ para $i = 1, 2, \dots, n$ na região de interesse dos parâmetros físicos x_f . Para desacoplar os neurônios ocultos, escolhem-se os seguintes valores para w_i^h e b^h , Rayas-Sanchez (2001),

$$\begin{bmatrix} (w_1^h)^T \\ \vdots \\ (w_n^h)^T \end{bmatrix} = \begin{bmatrix} \mathbf{D} \\ \mathbf{0} \end{bmatrix} \begin{matrix} n \times n \\ (h-n) \times n \end{matrix} \quad \text{e } b^h = \mathbf{0} \quad (6.27)$$

em que, \mathbf{D} é uma matriz diagonal cujos elementos são w_{ii}^h . Portanto,

$$z_i = \begin{cases} \varphi(b_i^h + \mathbf{v}^T \mathbf{w}_i^h) = \varphi(w_{ii}^h v_i), & i = 1, 2, \dots, n \\ 0 & i = n+1, \dots, h \end{cases} \quad (6.28)$$

Para desacoplar os neurônios ocultos, escolhem-se os seguintes valores

$$\begin{bmatrix} (\mathbf{w}_1^o)^T \\ \vdots \\ (\mathbf{w}_n^o)^T \end{bmatrix} = \begin{bmatrix} w_{11}^o & 0 & \dots & 0 & \dots \\ 0 & w_{22}^o & 0 & \dots & 0 & \dots \\ \vdots & \vdots & \ddots & & 0 & \dots \\ 0 & 0 & 0 & w_{nn}^o & 0 & \dots \end{bmatrix} \quad (6.29)$$

tal que

$$y_i = b_i^o + \mathbf{z}^T \mathbf{w}_i^o = b_i^o + w_{ii}^o \varphi(w_{ii}^h v_i), \quad i = 1, 2, \dots, n \quad (6.30)$$

6.3.2 Função de Ativação Sigmóide

Se uma função sigmóide (logística) for usada, a resposta do k -ésimo neurônio oculto é dada por (Haykin, 1999):

$$z_k = \varphi(s_k) = \frac{1}{1 + e^{-s_k}}, \quad k = 1, 2, \dots, h \quad (6.31)$$

que pode ser aproximada por

$$z_k = \varphi(s_k) = \frac{1}{1 + e^{-s_k}} = \frac{1}{2} \left(\frac{1}{2} s_k + 1 \right), \quad \text{para } |s_k| \ll 1 \quad (6.32)$$

Desde que, $-1 \leq v_i \leq 1$, devido ao escalonamento das entradas, selecionando-se $w_{ii}^h = 0.1$, para $i = 1, 2, \dots, n$ em (6.30), então, com o emprego de (6.32), (6.31) é aproximada por,

$$y_i = b_i^o + \frac{w_{ii}^o}{2} \varphi\left(\frac{1}{20}v_i + 1\right), \quad i = 1, 2, \dots, n \quad (6.33)$$

Por inspeção de (6.33), e com os valores,

$$w_{ii}^o = 40 \text{ e } b_i^o = -20 \text{ para } i = 1, 2, \dots, n \quad (6.34)$$

obtem-se o mapeamento unitário desejado. Portanto, um ponto de partida para os parâmetros internos da ANN (quando funções de ativação do tipo sigmóide são usadas) é dado por (6.27) com $w_{ii}^h = 0.1$, (6.29) e (6.34).

6.4 ANÁLISE DE UMA JUNÇÃO-T EM LINHA DE MICROFITA ATRAVÉS DE MÉTODOS SM-ANN

Nesta seção, os seguintes métodos SM-ANN foram simulados: SMN, FSMN, FDSMN e FDPSMN. Para tanto, segue-se parte do procedimento realizado por Rayas-Sanchez (2001), que abordaram o uso da técnica SM-ANN para uma curva de 90° em microfita. Aqui, considera-se o caso de uma junção-T em microfita. A fim de facilitar a comparação dos métodos implementados, uma região de interesse reduzida foi escolhida para o espaço de parâmetros de projeto da junção-T. Abaixo, descreve-se os procedimentos utilizados e alguns resultados de simulação são apresentados.

Seja a junção-T em linha de microfita, ilustrada na Fig. 6.13(a), com os seguintes parâmetros de projeto: larguras dos condutores W_1 e W_2 , comprimentos dos condutores ℓ_1 e ℓ_2 , altura do substrato h , permissividade relativa do substrato ϵ_r , e, frequência ω .

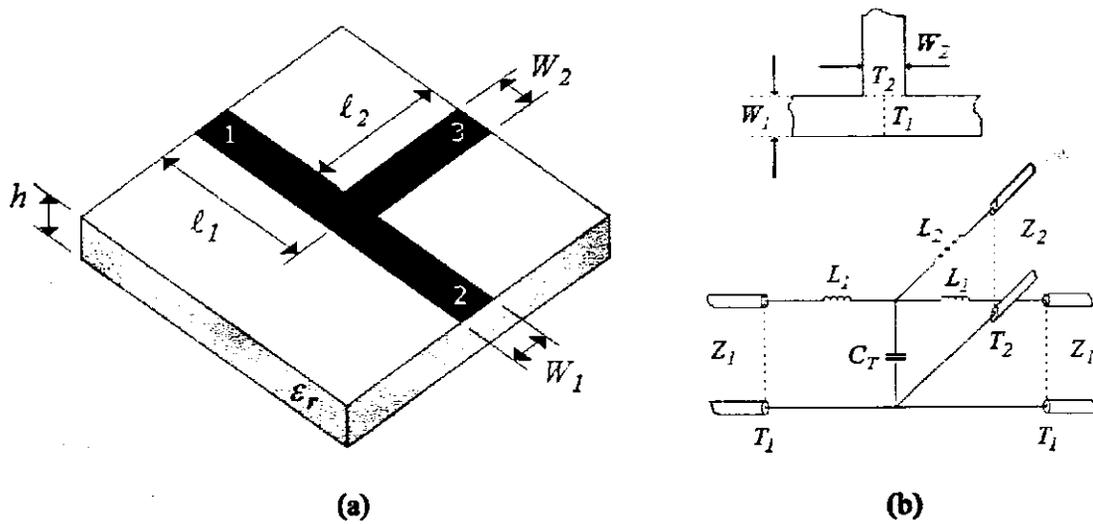


Fig. 6.13 (a) Junção-T em linha de microfita. (b) Circuito equivalente.

O modelo empírico proposto por Gupta et al (1981) foi empregado para aproximar o comportamento da junção-T. O circuito equivalente para a junção-T, Fig. 6.13(b), consiste de uma capacitância, C_T , e duas indutâncias: uma para a linha principal e uma para o toco, L_1 e L_2 , respectivamente. As expressões para os parâmetros do circuito equivalente para a junção-T, com uma impedância da linha principal de 50Ω e para $\epsilon_r = 9,9$, são dadas por:

$$C_T / W_1 = \frac{100}{\tanh(0,0072Z_o)} + 0,064Z_o - 261 \quad (\text{pF/m}) \quad (25 \leq Z_o \leq 100) \quad (6.35)$$

em que, Z_o é a impedância característica do toco em ohms.

$$L_1 / h = -\frac{W_2}{h} \left[\frac{W_2}{h} \left(-0,016 \frac{W_1}{h} + 0,064 \right) + \frac{0,016}{W_1/h} \right] L_{w1} \quad 0,5 \leq (W_1/h \quad W_2/h) \leq 2,0 \quad (\text{nH/m}) \quad (6.36)$$

$$L_2 / h = \left[\left(0,12 \frac{W_1}{h} - 0,47 \right) \frac{W_1}{h} + 0,195 \frac{W_1}{h} - 0,357 + 0,0283 \operatorname{sen} \left(\pi \frac{W_1}{h} - 0,75\pi \right) \right] L_{w2} \quad 1 \leq W_1/h \leq 2 \quad 0,5 \leq W_1/h \leq 2 \quad (\text{nH/m}) \quad (6.37)$$

em que, L_w é a indutância por unidade de comprimento (nH/m) para uma microfita de largura W , sendo obtida por:

$$L_w = \frac{Z_o \sqrt{\epsilon_{re}}}{c} \text{ (H/m)}, \quad c = 3 \times 10^8 \text{ (m/s)} \quad (6.38)$$

A representação através da matriz de espalhamento para a junção-T é obtida por:

$$S_{11} = S_{22} = \frac{\bar{Z}_{in} - 1}{\bar{Z}_{in} + 1} \quad (6.39)$$

$$S_{12} = S_{21} = \frac{(1 - S_{11})\bar{X}_3(1 + \bar{X}_2)}{(1 + \bar{X}_2)(1 + \bar{X}_1 + \bar{X}_3) + \bar{X}_3(1 + \bar{X}_1)} \quad (6.40)$$

$$S_{13} = S_{31} = S_{23} = S_{32} = \frac{(1 - S_{11})\bar{X}_3(1 + \bar{X}_1)}{(1 + \bar{X}_2)(1 + \bar{X}_1 + \bar{X}_3) + \bar{X}_3(1 + \bar{X}_1)} \quad (6.41)$$

$$S_{33} = \frac{\bar{Z}'_{in} - 1}{\bar{Z}'_{in} + 1} \quad (6.42)$$

em que,

$$\bar{X}_1 = \frac{j\omega L_1}{Z_o} \quad (6.43)$$

$$\bar{X}_2 = \frac{j\omega L_2}{Z_o} \quad (6.44)$$

$$\bar{X}_3 = \frac{1}{j\omega Z_o C_T} \quad (6.45)$$

Os dados do modelo preciso (parâmetros de espalhamento), usados no treinamento das ANNs, foram obtidos através de resultados de simulações do programa Sonnet's *em*TM (1997). As respostas dos modelos preciso e empírico (antes do uso dos métodos SM-ANN) são apresentadas na Fig. (6.14). Os seguintes valores para os parâmetros físicos da junção-T foram considerados: $\epsilon_r = 9,9$, $W_1 = W_2 = h = 508 \mu\text{m}$, $\ell_1 = 2,0 \text{ mm}$, $\ell_2 = 1,8 \text{ mm}$. Neste caso, o modelo de Gupta et al (1981) leva a respostas aceitáveis apenas para frequências inferiores a 2,0 GHz.

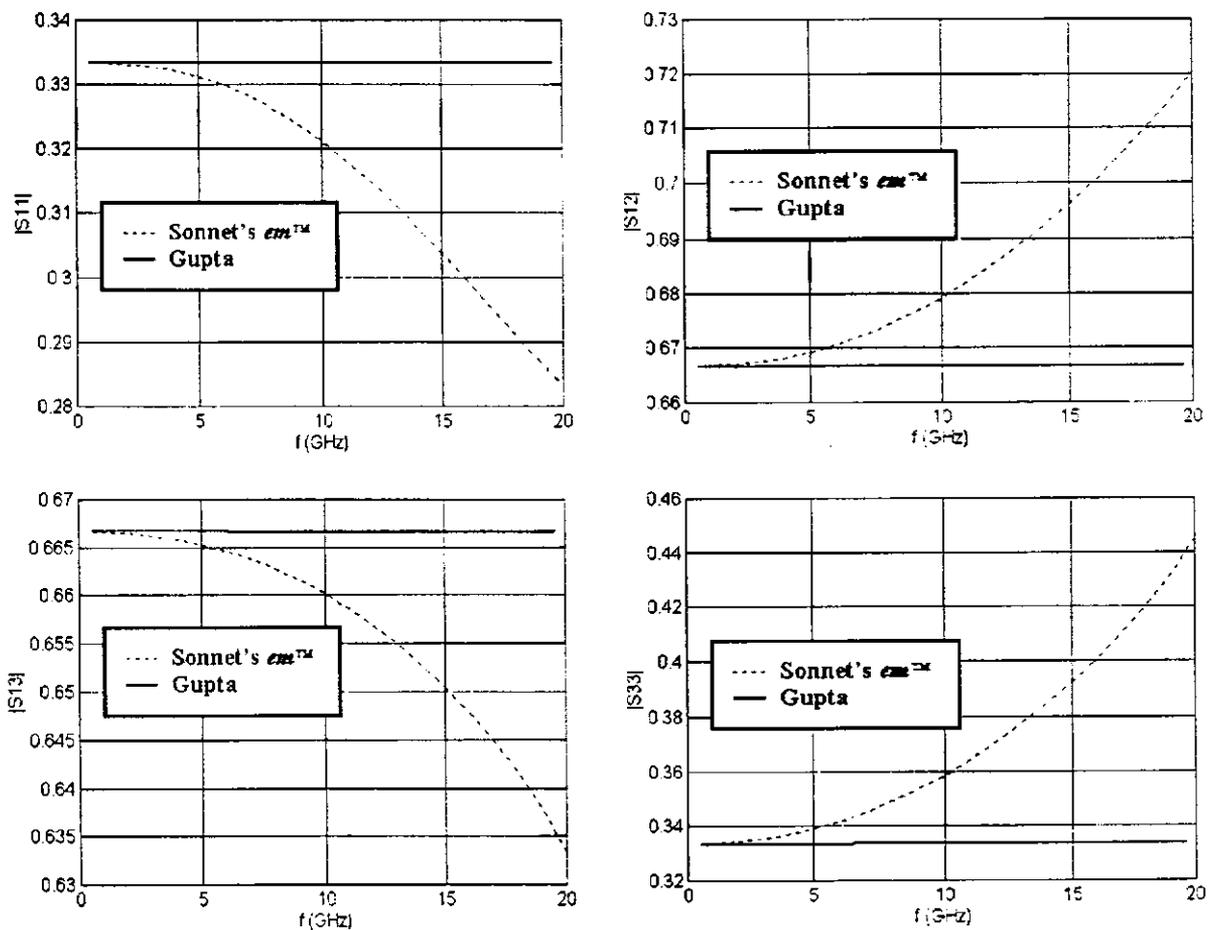


Fig. 6.14 Respostas típicas dos modelos preciso, Sonnet's *em*TM (1997) e empírico, Gupta et al (1981), antes do uso dos métodos SM-ANN.

Nas simulações dos métodos SM-ANN, o algoritmo Rprop foi utilizado no treinamento das redes MLPs, com os seguintes parâmetros: $\Delta_{max} = 50,0$, $\Delta_{min} = 10^{-7}$, $\eta^+ = 1,2$ e $\eta^- = 0,5$. Numa comparação das técnicas de SM baseadas nas redes MLPs implementadas, define-se o critério de parada para 2000 épocas de treinamento.

A fim de simplificar a resposta a ser modelada, considera-se os parâmetros físicos anteriores, com uma variação apenas da largura W_2 . Assim, define-se a região de interesse $127 \leq W_2 \leq 508$ μm para a junção-T. Especificamente, tem-se $B_p = 3$, em que: $W_2 = [127 \ 254 \ 508] \mu\text{m}$, $F_p = 11$, $0,1 \leq f \leq 20$ GHz (em passos de 2,0 GHz), totalizando 33 exemplos de treinamento. Por outro lado, na definição do conjunto de respostas R_f , admite-se apenas a magnitude dos parâmetros de espalhamento para a junção-T.

Como uma primeira tentativa, tentou-se o emprego de apenas uma rede MLP para modelar o comportamento da junção-T. Contudo, verificou-se que, quando erro calculado para o treinamento do MLP leva em conta todo o conjunto de respostas R_f , o modelo empírico não é capaz de responder adequadamente.

Por isso, ao invés de se utilizar apenas uma rede MLP para mapear as entradas do modelo empírico, a tarefa de aprendizado foi distribuída entre quatro redes MLPs, uma para cada parâmetro de espalhamento da junção-T – S_{11} , S_{12} , S_{13} e S_{33} . As redes MLPs podem então ser treinadas individualmente, para aproximar um subconjunto de respostas em R_f . Isto implica na solução de quatro problemas de otimização independentes, ao invés de apenas um.

Os 33 exemplos usados no treinamento ($W_2 = [127 \ 254 \ 508] \mu\text{m}$), mais 22 exemplos de teste ($W_2 = [203,2 \ 381] \mu\text{m}$), bem como, as respostas dos modelos SMN, FSMN, FPSMN e FDPSMN, são apresentados a seguir, Fig. 6.15 – 6.18, respectivamente. Por inspeção das respostas destes modelos SM-ANN, pode-se verificar suas capacidades para interpolar os exemplos de treinamento, bem como, generalizar para exemplos na região de interesse, mas fora do conjunto de treinamento.

Como primeiro exemplo, verificou-se a possibilidade de um mapeamento independente da frequência. Assim, o método SMN foi simulado com os seguintes parâmetros de projeto para a junção-T:

$$\mathbf{x}_e = [\epsilon_r \ h \ W_1 \ W_2 \ \ell_1 \ \ell_2]^T \tag{6.46}$$

A seguinte configuração para as redes MLPs foi empregada: seis unidades de entrada, seis unidades ocultas e seis unidades de saída. A aproximação da resposta R_i através do modelo SMN apresentou um resultado melhor para o parâmetro S_{11} . Algumas dificuldades foram encontradas no mapeamento de S_{12} devido a divergências deste método.

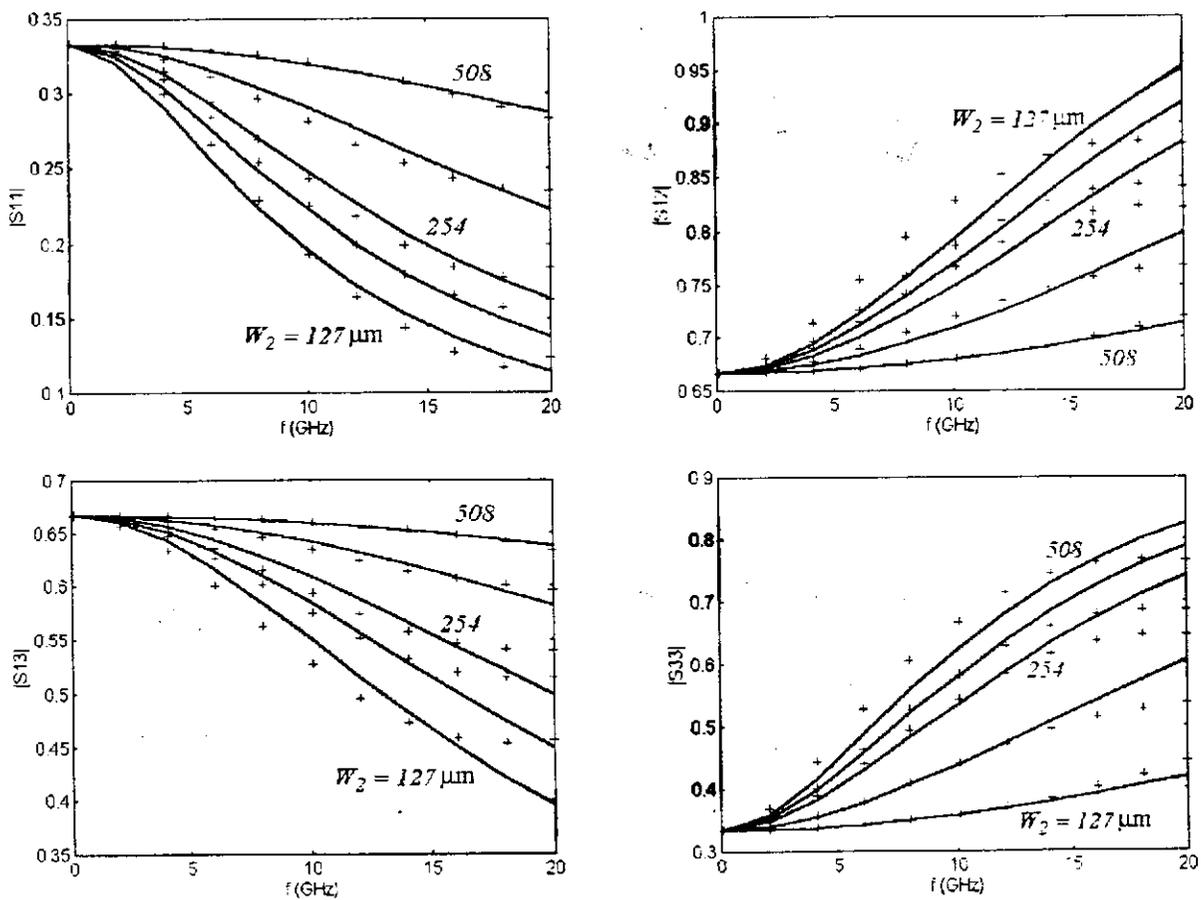


Fig. 6.15 Respostas: (—) modelos SMN; (+) modelo preciso.

O método FSMN foi simulado com os seguintes parâmetros de projeto para a junção-T:

$$\mathbf{x}_e = [\varepsilon_r \quad h \quad W_1 \quad W_2 \quad \ell_1 \quad \ell_2 \quad f]^T \quad (6.47)$$

A seguinte configuração para as redes MLPs foi empregada: sete unidades de entrada, sete unidades ocultas e sete unidades de saída. Com a utilização deste método, obtém-se respostas mais precisas que as do método SMN. Uma maior precisão pôde ser observada na aproximação das respostas S_{11} e S_{13} .

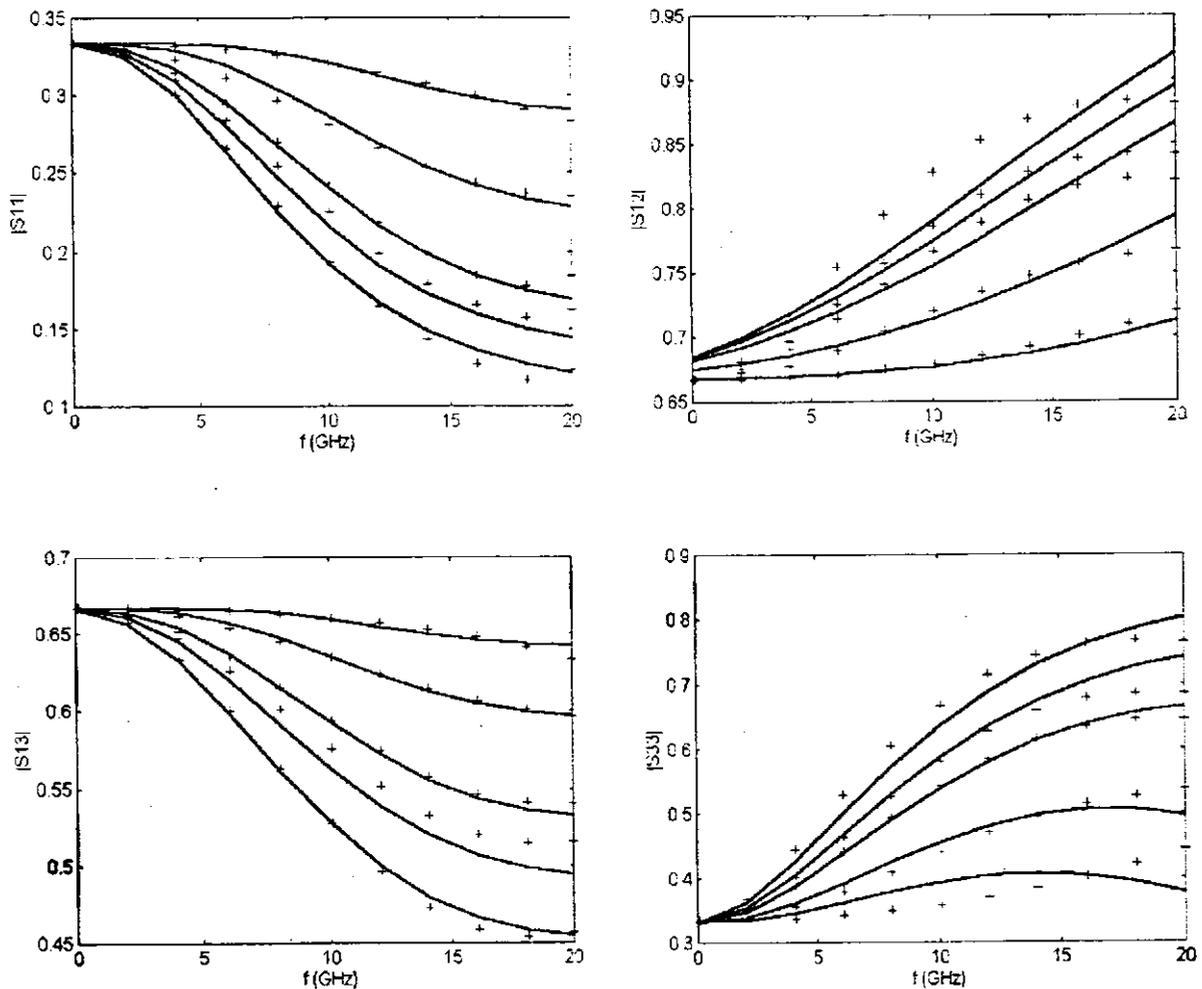


Fig. 6.16 Respostas: (—) modelos FSMN; (+) modelo preciso

O método FPSMN foi simulado com os seguintes parâmetros de projeto:

$$\mathbf{x}_e = [h \quad W_2 \quad \ell_2 \quad f]^T \quad (6.48)$$

Empregou-se a seguinte configuração para três das redes MLPs: quatro unidades de entrada, quatro unidades ocultas e quatro unidades de saída. Para o quarto MLP, que aproxima da resposta S_{33} , levou-se em conta o parâmetro de projeto ℓ_1 , acrescentando-se um neurônio a cada camada do MLP. Os resultados obtidos são melhores que os anteriores.

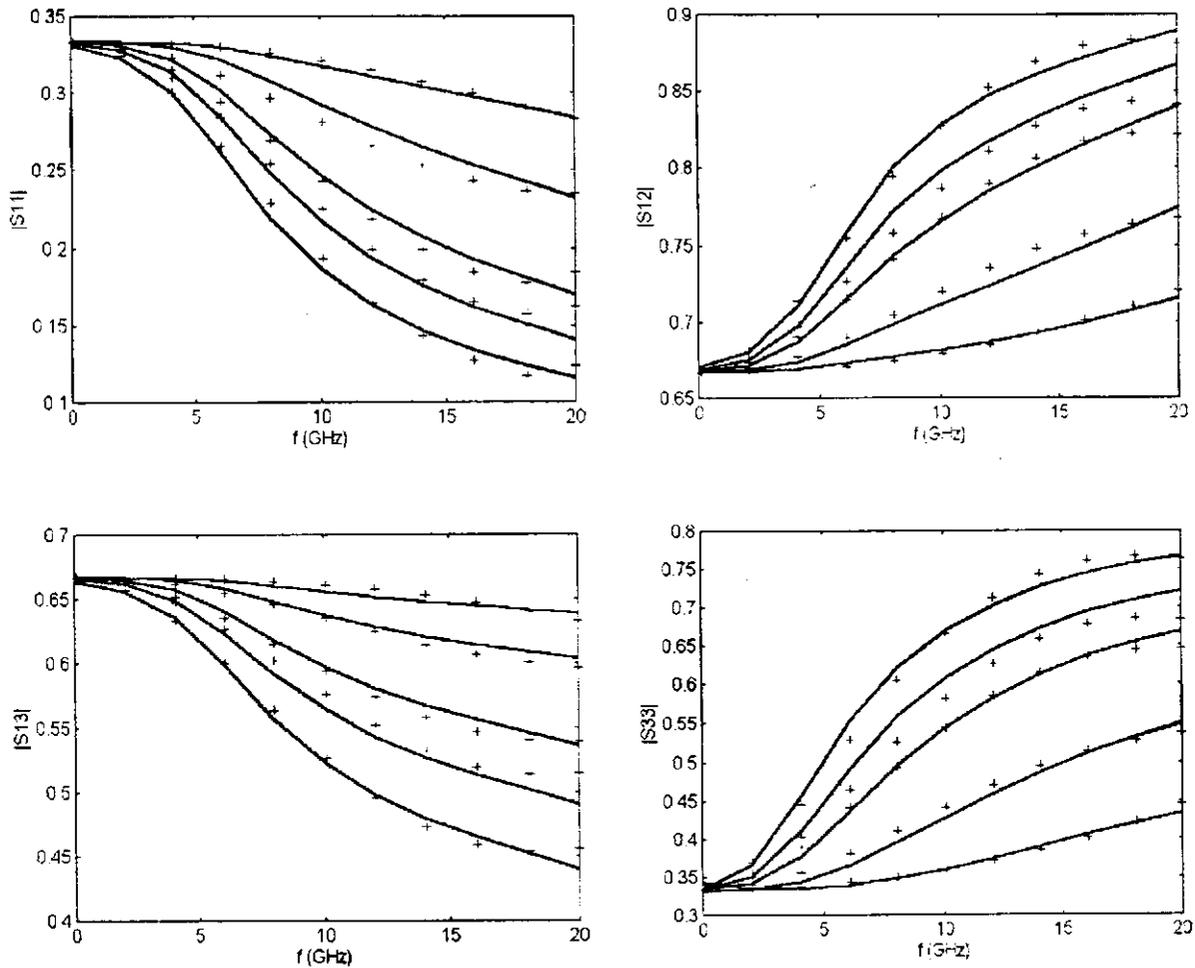


Fig. 6.17 Respostas: (—) modelos FPSMN; (+) modelo preciso

O método FDPSMN foi simulado com os seguintes parâmetros de projeto:

$$\mathbf{x}_e = [h \ W_1 \ W_2 \ l_1 \ l_2 \ f]^T \quad (6.49)$$

Empregou-se a seguinte configuração para as redes MLPs: seis unidades de entrada, seis unidades ocultas e seis unidades de saída. Através deste novo método SM-ANN, obtém-se melhores resultados que os anteriores. Isto é confirmado através da comparação das curvas de erro para cada método, Fig. 6.19.

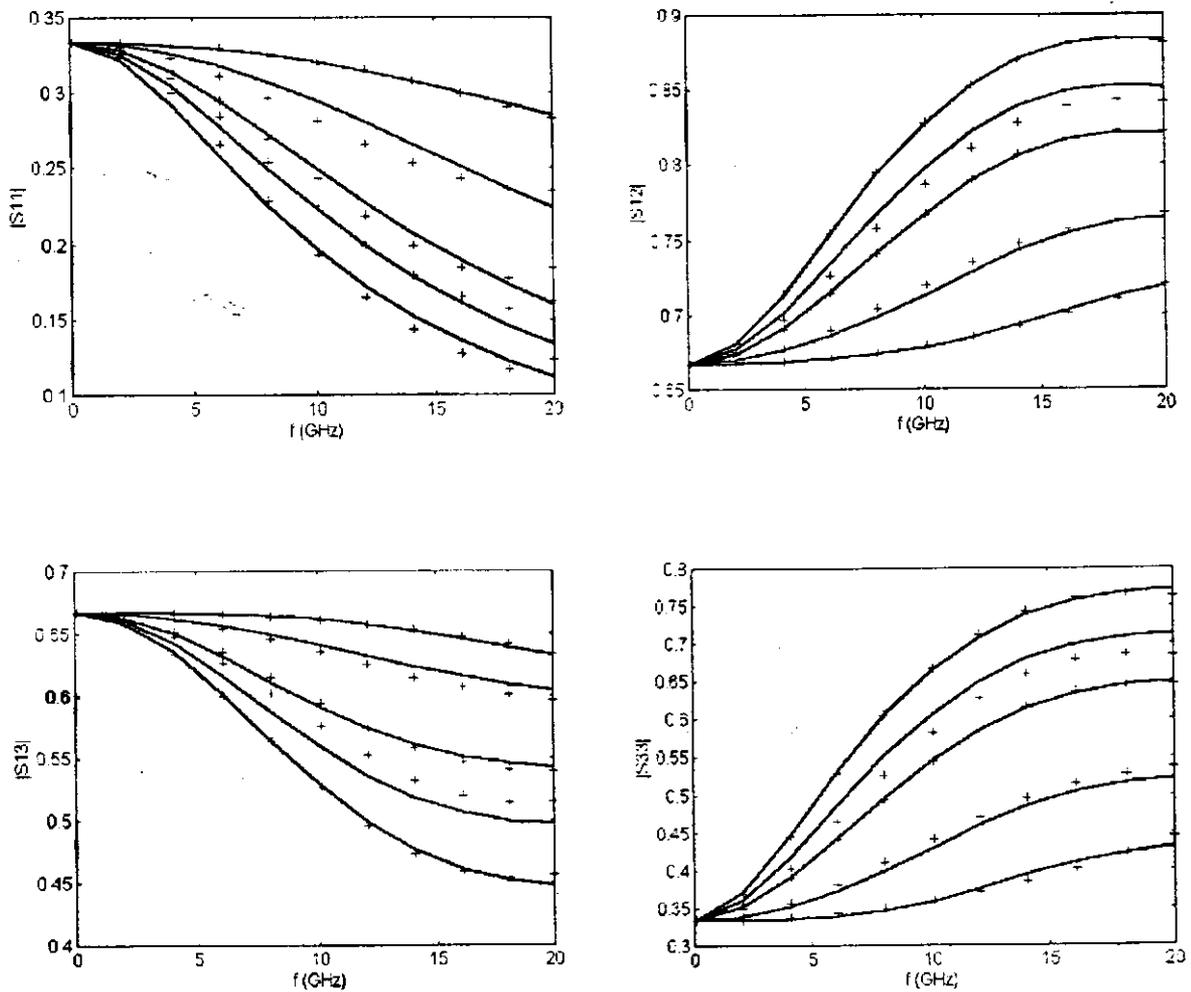


Fig. 6.18 Respostas: (—) modelos FDPSMN; (+) modelo preciso

Por inspeção das respostas dos modelos SM-ANN simulados, Fig. (6.14)-(6.18), constata-se que, o modelo FDPSMN proposto interpola melhor os dados de treinamento. Além disso, sua capacidade de generalização na região de interesse foi superior a dos outros métodos. A convergência dos métodos SM baseados em redes MLPs (treinadas através do algoritmo Rprop) são apresentados na Fig. 6.19. Comparando estes resultados, verifica-se que a técnica FDPSMN, com uma convergência mais estável, apresenta o menor erro de treinamento.

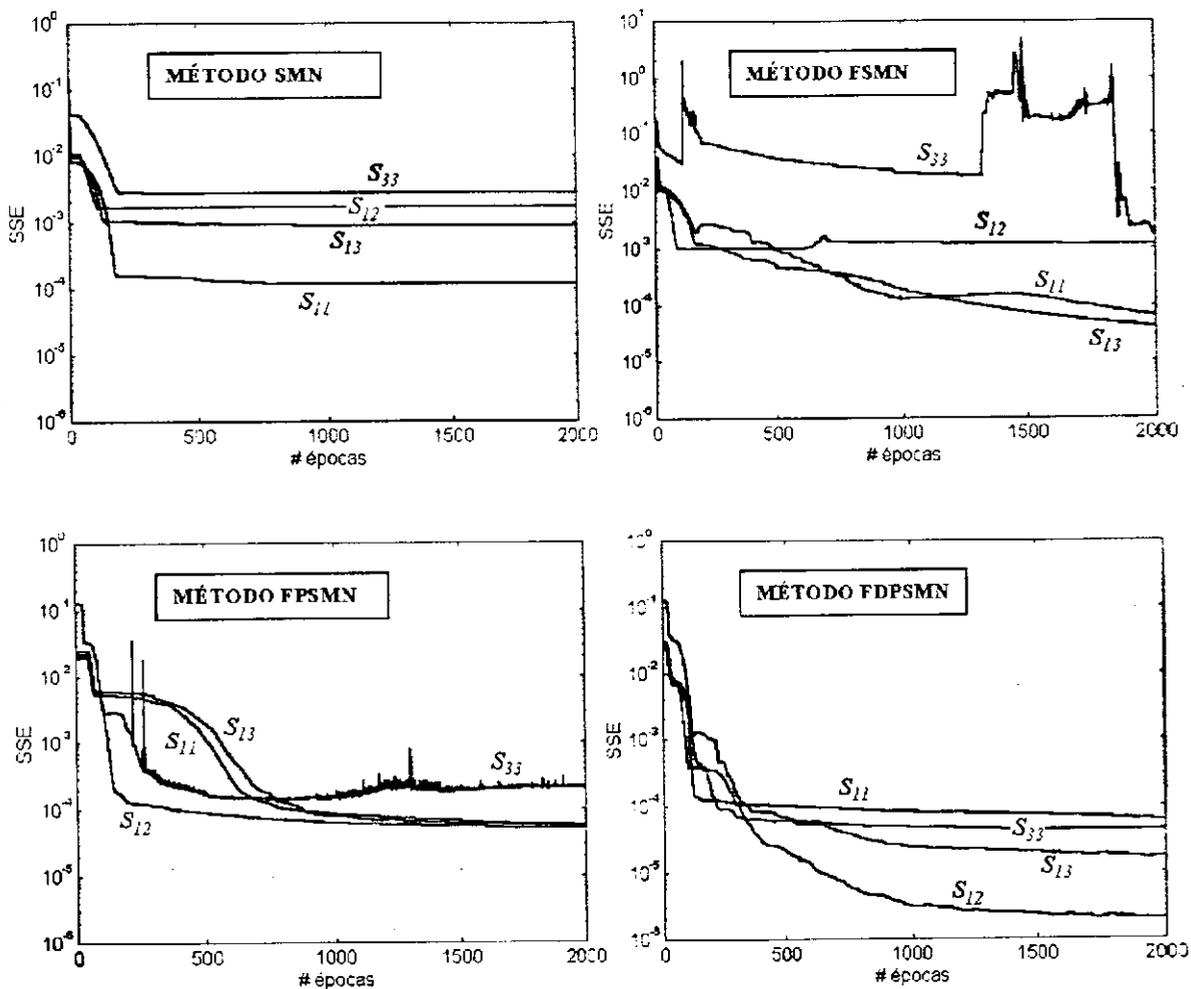


Fig. 6.19 Evolução dos erros nos métodos SM baseados em redes MLPs.

Capítulo 7

CONCLUSÕES

Esta tese limita-se à abordagem das redes neurais sem realimentação, cujos algoritmos de treinamento são supervisionados, usando a estratégia de aprendizado através da correção do erro. Especificamente, analisa-se o uso destas técnicas neuro-computacionais alternativas em duas áreas distintas da engenharia de RF/microondas: modelagem de dispositivos e simulação de circuitos não lineares.

Cinco tipos de redes neurais sem realimentação foram consideradas: MLP, RBF, *wavelet*, Fourier e RFS. Os respectivos algoritmos de treinamento foram formulados e implementados em Matlab™ e C++. Especial atenção foi dada ao aprendizado de redes MLPs. Neste caso, dentre os algoritmos implementados: BP, delta-delta, delta-bar-delta, SuperSAB, Rprop; este último apresentou os melhores resultados (convergência e robustez na escolha dos parâmetros do algoritmo). As demais ANNs foram treinadas diretamente através do método do gradiente.

A viabilidade de aplicações das ANNs para modelar as principais características de alguns dispositivos de RF/microondas, foi verificada. O uso de modelos EM-ANNs alternativos resulta em algumas vantagens, tais como, pode-se citar: baixa ocupação de memória, computação eficiente de mapeamentos não lineares complexos, capacidade de modelagem de diferentes dispositivos com a mesma configuração de rede neural. Estas características tornam os modelos EM-ANN bastante atrativos em implementações de programas de CAD na área de RF/microondas. Neste

sentido, resultados de simulações EMs de diferentes dispositivos podem ser incorporados em ferramentas de CAD através das ANNs.

Se a rede for treinada com um número suficiente de exemplos de treinamentos, a ANN serve como um modelo rápido e preciso na região de interesse. Funcionando como modelos tipo caixa preta, associa desempenho computacional e precisão nos resultados, evitando aproximações analíticas ou o uso de tabelas de consulta, que são recursos comuns em programas de CAD usuais.

Uma desvantagem das ANNs é o grande esforço computacional requisitado durante a fase de treinamento. Em modelagem de dispositivos, contudo, isto não é uma limitação, uma vez que o treinamento é feito *off-line*.

No uso do método EM-ANN, a escolha da configuração da rede é um ponto crucial. Nos exemplos de aplicação, exceto para filtros distribuídos em microfita, a configuração com uma camada oculta foi empregada. Nestes casos, o problema da escolha do número de neurônios ocultos foi resolvido por tentativa e erro. A complexidade da rede neural foi selecionada, de tal forma que: a ANN apresente um pequeno erro de treinamento e uma generalização boa na região de interesse.

Além disso, sua capacidade de generalizar a partir de exemplos, torna bastante atrativa a aplicação para dispositivos novos, onde pouco ou nenhum conhecimento existe sobre suas respostas.

No problema de modelagem de dispositivos e circuitos de RF/microondas certas situações são encontradas, nas quais, os métodos EMs e as populares ANNs, como o MLP, são imprecisos. Em geral, a rede RBF, com a propriedade de mapeamento local, e a rede *wavelet*, com a propriedade de análise em escalas de resolução, apresentam respostas mais precisas.

Motivado por problemas de modelagem com comportamentos localizados, misturando regiões com não linearidades suaves e abruptas (por exemplo, para medidas de parâmetros de espalhamento de circuitos de RF/microondas); e, através das propriedades da função *sinc* em processamento e reconstrução de sinais, a função de ativação do tipo *sample* foi proposta para os neurônios ocultos de uma ANN. A rede neural sem realimentação assim definida foi denominada RFS. Como no caso das redes RBF e *wavelet*, translações e dilatações foram incluídas no modelo RFS.

Os excelentes resultados obtidos com os modelos RFS aplicados à aproximação de parâmetros *S* medidos, para PBGS como filtro em linhas de microfita e cargas padronizadas para a calibração de VNAs, demonstram as propriedades de análise em escalas de resolução da RFS. Um problema encontrado com uso da função *sample*, ondulações devido à sobreposição das saídas dos neurônios ocultos, foi eliminado através do conceito de neurônios de resolução alta e baixa. As relações recursivas para a otimização do modelo RFS são simples, aumentando a eficiência do seu algoritmo de treinamento. Além disso, as diferentes formas que a função *sample* pode assumir, aumenta a classe de mapeamentos que a rede RFS pode aproximar. O emprego de um conjunto de treinamento normalizado e o conhecimento prévio das formas assumidas pela função *sample*, facilitam a escolha dos parâmetros do modelo RFS, como a seleção de seus centros e larguras iniciais.

Em simulação de circuitos não-lineares, descreveu-se uma aplicação inédita dos modelos ANNs de dispositivos ativos para análise transitória no domínio do tempo. Em particular, precisos resultados numéricos de modelos bidimensionais para o GaAs MESFET foram disponibilizados de uma maneira numericamente eficiente, para o uso em programas simuladores de circuito, como o SPICE. Resultados de simulação de uma porta lógica construída com GaAs-MESFETs, mostraram que a aproximação global da rede

MLP (treinada com dados representativos), generaliza para regiões de interesse onde não existe dados de treinamento. A comparação com modelos analíticos mostrou a superioridade do modelo neural MLP.

Uma variação da técnica de balanço harmônico, denominada MBH simplificado, foi proposta e formulado com o emprego de redes de Fourier. O MBH simplificado foi implementado em C++ com interface gráfica do Matlab™ para até 20 harmônicas. Nesta variação do MBH, demonstrou-se como a análise direta possibilita a solução analítica, no domínio da frequência, da parte linear do circuito. Com isto obteve-se, também de forma analítica, o gradiente da função custo, utilizado na rotina de otimização. Desta forma, a implementação numérica do método ficou bastante facilitada, uma vez que foi evitado o cálculo da DFT para análise da parte linear do circuito, como é comum em outros MBHs.

Verificou-se que uma boa escolha para a solução inicial é crucial para a convergência do método. Em geral, inicia-se o MBH com poucas harmônicas, aumentando este número até que um critério de convergência seja alcançado. O método foi testado e validado para diferentes circuitos com poucos elementos não lineares e entradas puramente senoidais em diferentes faixas de frequência.

Os métodos através do emprego de redes neurais e modelos empíricos, têm sido propostos para solucionar problemas de modelagem na área de microondas. O mapeamento de espaços baseado em redes neurais (SM-ANN) diminui o número de simulações EMs para treinamento, aumenta a habilidade de generalização e reduz a complexidade da topologia da rede neural.

Algumas vantagens são verificadas no uso do método SM-ANN: a redução dos dados de treinamento necessários; a utilização de ANNs com configurações simples; a melhoria das capacidade de generalização das redes neurais.

Uma técnica SM-ANN nova, designada FDPSMN, foi proposta. Os resultados obtidos na modelagem de uma junção-T em linha de microfita. Neste caso, a comparação dos resultados obtidos com os de outras técnicas SM-ANN, demonstraram um melhor desempenho do FDPSMN.

A partir da experiência e do conhecimento adquiridos no curso desta pesquisa, são sugeridos os seguintes tópicos para desenvolvimentos futuros:

- (1) Os paradigmas de redes neurais sem realimentação considerados nesta tese são apenas uma pequena amostra das possibilidades desta técnica neuro-computacional. Futuras aplicações nas áreas de RF e microondas podem incluir novos tipos de função de ativação para os neurônios ocultos da ANN. Os resultados obtidos com o modelo RFS aponta as possibilidades nesta área.
- (2) O sucesso das aplicações descritas no uso de modelos EM-ANN para mapeamento de resultados EMs, em conjunto com as vantagens apresentadas pelas ANNs em problemas de modelagem, naturalmente apontam na direção da integração da metodologia EM-ANN em programas comerciais de RF/microondas, como por exemplo: simuladores de circuitos e aplicações de CAD para dispositivos ativos e passivos.
- (3) Na análise de circuitos não lineares empregando modelos de dispositivos ativos através de redes neurais, o ganho computacional e a precisão obtidos nos resultados destes modelos para os transistores MESFET de GaAs, encorajam, nesta linha de pesquisa, a consideração da modelagem de outros dispositivos ativos de RF/microondas.
- (4) Aplicações específicas do método do balanço harmônico em conjunto com as redes de Fourier, por exemplo, para análise de ganho e distorção em circuitos amplificadores nas aplicações de RF/microondas.

- (5) Um primeiro passo foi dado no desenvolvimento de modelos neurais baseados no conhecimento empírico. Futuros desenvolvimentos na área de mapeamentos de espaços, através de novas técnicas como a FDPSMN, são bastante atrativas.

REFERÊNCIAS

Alvager, T., Smith, T. J., e Vijai, F., The Use of Artificial Neural Networks in Biomedical Technologies: An Introduction, *Biomedical Instrumentation and Technology*, v. 28, p. 315-322, 1994.

Aprille Jr., T. J., Trick, T. N., Steady-state analysis of nonlinear circuits with periodics inputs. *IEE Proceedings*, v. 60, p. 108-114, 1972.

Balakrishnan, S. N., Weil, R. D., Neurocontrol: A Literature Survey, *Mathematical and Computer Modeling*, v. 23, No. 1-2, p. 101-117, 1996.

Bandler, J. W., Chen, S. H., Biernacki, R. M., Gao, L., Madsen, K., e Yu, H., Hubber optimization of circuits: a robust approach, *IEEE Trans. on Microwave Theory and Techniques*, v. MTT-41, p. 2279-2287, 1993.

Bandler, J. W., Biernacki, R. M., Chen, S. H., Grobelny, P. A., e Hemmers, R. H., Space mapping technique for eletromagnetic optimization. *IEEE Trans. on Microwave Theory and Techniques*, v. MTT-42, p. 2536-2544, 1994.

Barnes, J. J., Lomax, R. J., e Haddad, G. I., Finite-element simulation of GaAs MESFET's with lateral doping profiles and submicron gates. *IEEE Trans. Electron Devices*, v. ED-23, p. 1042-1048, Set., 1976.

Barby, J. A., Vlach, J., e Singhal, K., Polinomial splines for MOSFET model approximation. *IEE Trans. Computer-Aided Design*, v. CAD-7, p. 557-567, 1988.

Bayly, E. M., Steady state harmonic analysis of nonlinear networks. Tese de Ph.D., Stanford Univ., Stanford, 1968.

Bila, S., et al., Accurate wavelet neural network based model for electromagnetic optimization of microwaves circuits, *Int. Journal of RF and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, Zhang, Q. J., Creech, G. L., (Guest Editors), NY: Willey, v. 9, p. 297-306, 1999.

Bishop, C. M., Neural Networks for Pattern Regonition, New York: Oxford University Press. Inc., 1996.

Burrascano, P., Dionigi, M., Fancelli, C., e Mongiardo, M., A neural network model for CAD and optimization of microwave filters, *IEEE MTT-S Int. Microwave Symp. Dig.*, Baltimore, p. 13-16, 1998.

Carnahan, B., Luther, H. A., Wilkes, J. O., Applied numerical methods. John Willey & Sons Inc., 1969.

- Chawla, B. R., Gummel, H. K., e Kozak, P., MOTIS – an MOS timing simulator. *IEEE Trans. on Circuits and System*, v. CAS-22, p. 901-909, Dez., 1995.
- Chen, Wai-Kai, The circuits and filters handbook. CRC Press, Inc., 1995.
- Cooper, B. S., Selected Applications of Neural Networks in Telecommunication Systems, *Australian Telecommunication Research*, v. 28, p. 9-29, 1994.
- Creech, G. L., et al., Artificial neural networks for fast and accurate EM-CAD of microwave circuits, *IEEE Trans. on Microwave Theory and Techniques*, v. MTT-45, p. 794-802, Mai, 1997.
- Curtice, W. R., A MESFET model for use in the design of GaAs Integrated Circuits. *IEEE Trans. on Microwave Theory and Techniques*, v. MTT-28, p. 448-456, Mai, 1980.
- Da Mata, W., Recuperation assistee des petroles visqueux par ondes electromagnetiques. Tese de Doutorado, INPT, França, 1993.
- Darken, C., J. Chang, e Moody, J., Learning rate schedules for fast stochastic gradient search, *In Neural Network for Signal Processing*, v. 2, *IEEE Workshop*, *IEEE Press*, p. 3-13, 1992.
- Daugman, J., Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acoust. Speech, Signal Proc.*, v. 8, Jul., 1988.
- Davidon, W. C., Variable metric method for minimization. *A.E.C. Research and Development Report*, ANL-5990 (Rev.), 1959.
- Devabhakruni, V., Xi, C., Wang, F., e Zhang, Q. J., Robust training of microwave neural models, *IEEE Int. Conf. Neural Networks*, Houston, TX, p. 266-269. Jun., 1999.
- em*TM Versão 4.0b, Sonnet Software, Inc., 1020 Seventh North Street, Suite 210, Liverpool, NY 13088, 1997.
- Ebers, J. J., Moll, J. L., Large-signal behavior of junction transistors. *Proceedings of the Institute of Radio Engineers*, p. 1761-1772, Dez., 1954.
- Fletcher, R., Powell, M. J. D., A rapidly convergent descent method for minimization. *Computer Journal*, p. 163-168, Jun., 1963.
- Fletcher, R., Reeves, C. M., Function minimization by conjugate gradient. *Computer Journal*, p. 149-154, Jul., 1964.
- Fusco, V. F., *Microwave Circuits Analysis and Computer-Aided Design*, New Jersey: Prentice-Hall, 1987.

Gerald, C. F., Applied numerical analysis. Reading: Addison-Wesley Publishing, 1978.

Gilmore, R. J., Steer, M. B., Nonlinear circuit analysis using the method of harmonic balance — a review of the art. Part I. introductory concepts. *Int. Journal of Microwave and Millimeter-Wave Computer Aided Engineering*, v. 1, p. 22-37, Jan., 1991.

Goita, K., et al., Literature review of artificial neural networks and knowledge based systems for image analysis and interpretation of data in remote sensing, *Canadian Journal of Electrical and Computer Engineering*, v. 19, p. 53-61, 1994.

Greenhouse, H. M., Design of planar rectangular microelectronic inductors. *IEEE Trans. on Parts, Hybrids, and Packaging*, v. PHP-10, Jun., 1974.

Günes, F., Gürgen, F., e Torpi, H., Signal-noise neural network for active microwaves devices. *IEE Proc. Circuits, Devices and Syst.*, v. 143, p. 1-8, Fev., 1996.

Gupta, K. C., Garg, R., Bahl, R., Microstrip Lines and Slotlines, Dedham, MA: Artech House, 1979.

Gupta, K. C., Garg, R., Chadha, R., Computer-Aided Design of Microwave Circuits, Massachusetts: Artech House, 1981.

Gupta, K. C., ANN and knowledge-based approaches for microwave design. *in Directions for the Next Generation of MMIC devices and Systems*, N. K. Das e H. L. Bertoni, Eds., NY: Plenum, p. 389-396, 1996.

Haefner, S. J., Alternating current of rectangular conductors. *Proceedings of the Institute of Radio Engineers*, v. 25, p. 434-447, 1937.

Harkouss, Y., et al., modeling microwave device and circuits for telecommunication system design, *Proc. IEEE Int. Conf. Neural Networks*, Anchorage, Alaska, p. 128-133, Mai., 1998.

Harkouss, Y., et al., Use of artificial neural networks in the nonlinear device and circuit modeling: an application to telecommunication system design, *Int. Journal of RF and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, Zhang, Q. J., Creech, G. L., (Guest Editors), NY: Wiley, v. 9, p. 198-215, 1999.

Haykin, S., Neural networks: A comprehensive foundation. New York: Macmillan, 1999.

Haykin, S., Adaptive filter theory. New Jersey: Prentice Hall, 1996.

Hornig, T., Wang, C., e Alexopoulos, N. G., Microstrip circuit design using neural networks, *IEEE MTT-S Int. Microwave Symp. Dig.*, p. 413-416, 1993.

- Hornik, K., Stinchcombe, M., e White, H., Multi-layered feed-forward neural networks are universal approximations, *Neural Networks*, v. 2, p. 259-266, 1990.
- Hu, Y., Obregon, J. J., e Mollier, J.C., Nonlinear analysis of microwave FET oscillators using Volterra series. *IEEE Trans. Microwave Theory and Techniques*, v. MTT-37, p. 1689-1693, Nov., 1989.
- Jain, N., Agnew, D., e Nakhla, M., Two-dimensional tabel lookup MOSFET model. *Proc. IEEE Int. Conf. on Computer-Aided Design*, Santa Clara, CA, 1983, p. 210-213.
- Jargon, J. A., Kirby, P., Gupta, K. C., Dunleavy, L., e Weller, T., Modeling load variations with artificial neural networks to improve on-wafer OSLT calibrations, work of the U. S. Government.
- Jargon, J. A., Gupta, K. C., Artificial neural network modeling for improved on-wafer line-reflect-match calibrations, *31st European Microwave Conf. Proc.*, v. 2, p. 229-232, Londres, Inglaterra, Set., 2001.
- Khatibzadeh, M. A., Trew, R. J., A large-signal, analytic model for the GaAs MESFET. *IEEE Trans. Microwave Theory and Techniques*, v. MTT-36, p. 231-238, Fev., 1988.
- King, R. W. P., Trembly, B. S., Strohbehn, J. W., The electromagnetic field of na insulated antenna in a conducting or dielectric medium. *IEEE Trans. on Microwave Theory and Techniques*, v. MTT-31, p. 574-583, Jul., 1983.
- Kundert, K. S., Sorkin, G. B., Sangiovanni-Vicentelli, A., Applying harmonic balance to almost-periodic circuits. *IEEE Trans. on Microwave Theory and Techniques*, v. MTT-36, p. 366-378, Fev., 1988.
- Le Cun, Y., Une procedure d'apprentissage pour reseau a seuil assymetrique. *Proc. Cognitiva'85*, p. 599-604, Paris, Jun., 1985.
- Lippmann, R. P., An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4, p. 4-22, 1987.
- Lindenlaub, J. C., An approach for finding the sinusoidal steady state response of nonlinear systems. *Proc. 7th Ann. Allerton Conf. Circuit and System Theory*, Univ. Illibois, Chicago, 1969.
- Litovski, V. B., et al., MOS transistor modeling using neural network, *Eletronics Letters*, v. 28, p. 1766-1768, 1992.
- Lowe, B. Adaptive radial basis function nonlinearity, and the problem of generalization. *IST-IEE International Conference of Artificial Neural Networks*, p. 171-175, London, 1989.

Mammanna, C. I. Z., Zuffo, J. A., Dispositivos e Circuitos Eletrônicos. Rio de Janeiro: McGraw-Hill, 1973.

Marar, J. F., Filho, E. C. B. C., e Vasconcelos, G. C., Wavelets polinomiais: uma família de funções splines para aplicações em processamento de sinais e imagens. *Anais do IX SIBGRAPI*, 305-312, 1996a.

Marar, J. F., Filho, E. C. B. C., e Vasconcelos, G. C., Function approximation by polynomial wavelets generated from powers of sigmoids. *SPIE Aerosense, 10th Annual International Aerosense Symposium, Wavelet Application III*, 365-374, 1996b.

MatlabTM, Versão 5.2, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA, 01760-9889, 1998.

MatlabTM Neural Network Toolbox, Versão 5.2, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA, 01760-9889, 1998.

McCulloch, W. S., Pitts, W., A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, p. 115-133, 1943.

McCamant, A. J., McCormack, G. D., e Smith, D. H., An improved GaAs MESFET model for SPICE. *IEEE Trans. Microwave Theory and Techniques*, MTT-38, p. 822-824, Jun., 1984.

Mishra, R. K., Patnaik, A., Neural network based CAD model for design of square patch antennas, " *IEEE Trans. Antennas Propagat.*, v. AP-46, p. 1890-1891, 1998.

Moody, J. E., Darken, C. J., Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1, p. 281-294, 1989.

Moreira, M., Fiesler, E., Neural Networks with Adaptive Learning Rate and Momentum Terms, IDIAP Technical Report, Out., 1995.

Nakhla, M. S., Vlach, J., A piecewise harmonic balance technique for determination of periodic response of nonlinear systems. *IEEE Trans. Circuits and Syst.*, v. CAS-23, p. 85-91, Fev., 1976.

Neto, A. G. Carvalho, e J. N., Silva, J. C., Caracterização Experimental de Circuitos em Microfita Utilizando Detetores de Microondas. *Principia*, p. 40-42, Mar., 1997.

Oppenheim, A. V., Schafer, R. W., Digital Signal Processing. New Jersey: Prentice Hall, 1975.

NeuroModeler (1999) Versão 1.2b, Prof. Q. J. Zhang, Dept. of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada, K1S5B6.

Nunmaker, J. F. Jr., Sprague Jr., R. H., Applications of Neural Networks in Manufacturing, *Proceedings of the Twenty-ninth, Hawaii International Conference on System Sciences*, v. 2, p. 447-453, 1996.

Oppenheim, A. V., Willsky, A. S., Signals and Systems. New Jersey: Prentice Hall, 1983.

Oppenheim, A. V., Schaffer, R. W., Digital Signal Processing. New Jersey: Prentice Hall, 1975.

Pati, Y. C., Krishnaprasad, P. S., Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. *IEEE Trans. Neural Networks*, v. 4, 73-85, Jan., 1993.

Patnaik, A., Mishra, R. K., Patra, G. K., e Dash, S. K., An artificial neural network model for effective dielectric constant of microstrip line, " *IEEE Trans. Antennas Propagat.*, v. AP-45, p. 1697, Nov., 1997.

Patnaik, A., Mishra, R. K., ANN techniques in microwave engineering. *IEEE Microwave*, p. 55-60, Mar., 2000.

Parker, D. B., Learning-logic. M.I.T. Cen. Computational Res. Economics Management Sci., Cambridge, TR-47, 1985.

Parker, A. E., Skellern, D. J., A realistic large-signal MESFET model for SPICE. *IEEE Trans. Microwave Theory and Techniques*, v. MTT-45, p. 1563-1571, Set., 1997.

Pettenpaul, E., Kapusta, H., Weisgerber, A., Mampe, H., Lungisland, J., e Wolff, I., CAD models of lumped elements on GaAs up to 18 GHz. *IEEE Trans. Microwave Theory Techniques*, v. MTT-36, p. 294-304, Fev., 1988.

Pillage, L. T., Rrohler, R. A., e Visweswariah, C., Electronic Circuit and System Simulation Methods, New York: McGraw-Hill, Inc., 1995.

Poggio, T., Girosi, F. A Theory of Networks for Approximation and Learning. AI Memo No. 1140, *Artificial Intelligence Laboratory, MIT*, 1989.

Poggio, T., Girosi, F. Networks for approximation and learning. *Proceedings of the IEEE*, 78, p. 1481-1497, Set., 1990.

Poggio, T., Girosi, F., e Jones, M., From regularization to radial tensor and additive splines, Int. Joint Conf. On Neural Networks, Nagoya, Japão. Out., 1993.

Polak, E., Ribière, G., Note sure la convergence de methods de directions conjures. *Reveu Française Information Recherche Operationnelle*, 16, p. 35-43, 1969.

Pucel, R. A., Design Considerations for Monolithic Microwave Circuits. *IEEE Trans. Microwave Theory Techniques*, v. MTT-29, p. 513-534, Jun., 1981.

Raghuram, R., Computer simulation of electronic circuits. New Delhi. Wiley Eastern Limited, 1989.

Qian, Y., Radisic, V., e Itoh, T., Simulation and experiment of photonic band-gap structures for microstrip circuits, *in Asia Pacific Microwave Conf.*, Hong Kong, p. 585-588, Dez., 1997.

Radisic, V., Qian, Y., e Itoh, T., Broad-band power amplifier using dielectric photonic bandgap structure, *IEEE Microwave and Guided Wave Lett.*, v. 8, p. 13-14, Out., 1998.

Rayas-Sanchez, J. E., Neural Space Mapping Methods for Modeling and Design of Microwave Circuits, Tese de Doutorado, McMaster University, Jun., 2001.

Reiser, M., A two-dimensional numerical FET model for DC, AC and large-signal analysis. *IEEE Trans. Electron Devices*, v. ED-20, p. 35-45, Jan., 1973.

Rhyne, G. W., Steer, M. B., e Bates, B. D., Frequency-domain nonlinear circuit analysis using generalized power series. *IEEE Trans. Microwave Theory and Techniques*, v. MTT-36, p. 379-387, Fev., 1988.

Riedmiller, M., Braun, H., A direct adaptive method for faster backpropagation learning: the Rprop algorithm. *Proceedings of the IEEE International Conference on Neural Networks*, 586-591, 1993.

Riedmiller, M., Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning algorithms, *Int. Journal of Computer Standards and Interfaces*, 1994.

Rioul, O., Vetterli, M., Wavelets and signal processing, *IEEE SP-Magazine*, p. 14-38, Out., 1991.

Rizzoli, V., Lipparini, A., Mastriarri, A. C. F., Cecchetti, C., Neri, A., e Masotti, D., State-of-the-art harmonic-balance simulation of forced nonlinear microwave circuits by the piecewise technique. *IEEE Trans. Microwave Theory and Techniques*, v. MTT-40, p. 12-27, Jan., 1992.

Robins, H., Monro, S., A stochastic approximation method, *Annals of Mathematical Statistics*, v. 22, p. 400-407, 1951.

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

Rosenblatt, F., Perceptron Simulation experiments. *Proceedings of the Institute of Radio Engineers*, 48, p. 301-309, 1962.

Rosenbrock, H. H., An automatic method for finding the greatest or the least value of a function. *The Computer Journal*, v. 3, p. 175., 1960.

Rumelhart, D., Hinton, D., e Williams, G., Learning internal representations by error propagation. D. Rumelhart e F. McClelland, eds., *Parallel Distributed Processing*, v. 1, Cambridge: M.I.T. Press, 1986.

Rumsey, I., Picket-May, M., Kelly, e P. K., Photonic bandgap structures used as filters in microstrip circuits, *IEEE Microwave and Guided Wave Lett.*, v. 8, p. 316-318, Out., 1998.

Rytting, D. K., Network analyzer error models and calibration methods, *52nd ARFTG Conference*, Short Course on Computer-Aided RF and Microwave Testing and Design, Rohnert Park, CA, Dez., 1998.

Santos, A. L., Romariz, A. R. S., Carvalho, P. H. P., Neural model of electrical devices for circuit simulation. *SBMO/IEEE MTT-S IMOC'97 Proceedings*, p. 253-258.

Schichman, H., Hodges, D. A., Modeling and simulation of insulated-gate field-effect transistor switching circuits. *IEEE J. Solid-State Circuits*, v. SC-3, p. 285-289, Set., 1968.

Schiffmann, W., Joost, M., e Werner, R., Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons, Technical Report, Institute of Physics, University of Koblenz, Set., 1994.

Shirakawa, A., Shimiz, M., Okubo, N., e Daido, Y., A large-signal characterization of an HEMT using a multilayer neural network. *IEEE Trans. on Microwave Theory and Techniques*, v. 45, p. 1630-1633, 1997.

Silva, P. H. da F., Contribuição ao Estudo de Antenas Aplicadas ao Aquecimento Eletromagnético de Meios Dissipativos. Dissertação de Mestrado, UFRN, 1997.

Silva, P. H. da F., Neto, A. D. D., e da Mata, W., Modelagem e simulação de antenas isoladas para o aquecimento eletromagnético de meios dissipativos. *Simpósio Brasileiro de de Microondas e Optoeletrônica*, Joinville, SC, 13 – 25 de Julho de 1998.

Silva, P. H. da F., Melo, M. A. B., e Neto, A. D. D., Projeto, análise e otimização de dispositivos e circuitos de microondas utilizando modelos de redes neurais. *Simpósio Brasileiro de Telecomunicações*, Vitória, ES, 1999.

Silva, P. H. da F., Melo, M. A. B., Método do Balanço harmônico Simplificado para análise em regime permanente senoidal de circuitos não-lineares. *IV Congresso Brasileiro de Eletromagnetismo*, Natal, RN, 19 – 22 Novembro de 2000.

Silva, P. H. da F., Melo, M. A. B., e Neto, A. D. D., Fast and Accurate Neural Network GaAs MESFET Model for Time-Domain Circuit Simulation, *SBMO/IEEE MTT-S Proc. International Microwave and Optoelectronics Conference*, Ago., 2001.

Silva, P. H. da F., Melo, M. A. B., e Neto, A. D. D., Using neural network models for efficient GaAs MESFET time-domain nonlinear circuit analysis, *Int. Joint Conf. On Neural Networks*, Honolulu, Hawaii, Mai., 2002. (a ser publicado)

- Smetanin, Y. G., Neural networks as systems for pattern recognition: A review, *Pattern Recognition and Image Analysis*, v. 5, p. 254-293, 1995.
- Smith, A. E., Mason, A. K., Cost estimation predictive modeling: regression versus neural network, *Engineering Economist*, v. 42, p. 137-160, Fev., 1997.
- Southall, H. L., Simmers, J. A., e O'Donnell, T. H., Direction finding in phased arrays with a neural network beamformer. *IEEE Trans. Antennas Propagat.*, v. 43, p. 1369-1374, Dez., 1995.
- Statz, H., Newman, P., Smith, I. W., Pucel, R. A., e Haus, H. A., GaAs FET device and circuit simulation in SPICE. *IEEE Trans. Electron Devices*, v. ED-34, p. 160-169, 1987.
- Sussman-Fort, S. E., Narasimhan, S., Mayaram, K., A complete GaAs MESFET computer model for SPICE. *IEEE Trans. on Microwave Theory and Techniques*, v. MTT-32, p. 471-472, Abr., 1984.
- Svarer, C., Hansen, L. K., e Larsen, J., On design and evaluation of tapped-delay neural network architectures, *Neural Networks Theory, Technology, and Applications*, IEEE Technical Activities Board, Simpson, K. P., Ed., p. 31-36, 1996.
- Szu, H., Telfer, B., E Kadambe, S., Neural network adaptive wavelets for signal representation and classification. *Optical Engineering*, v. 31, 1907-1916, Set., 1992.
- Thursby, M., Yoo, K., Neural control of smart electromagnetic structures. *IEEE Aerospace Electron. Syst.*, v. 31, p. 1341-1346, Out., 1995.
- Tollenaere, T., SuperSAB: fast adaptive backpropagation with good scaling properties, *Neural Networks*, v. 3, p. 561-573, 1990.
- Tsintikidis, D., et al., A neural network approach to estimating rainfall from spaceborne microwave data, *IEEE Trans. on Geoscience and Remote Sensing*, v. 35, p. 1079-1093, Mai., 1997.
- Vai, M., Prasad, S., Microwave circuit analysis and design by massively distributed computing network. *IEEE Trans. on Microwave Theory and Techniques*, v. 43, p. 1087-1094, 1995.
- Veluswami, A., Nakla, M. S., Zhang, Q. J., The application of neural networks to EM-based simulation and optimization of interconnects in high/speed VLSI circuits. *IEEE Trans. on Microwave Theory and Techniques*, v. 45, p. 712-722, Mai., 1997.
- Waldow, P., Wolff, I., Duals bounds variational formulation of skin effect problems. *IEEE MTT-S Int. Microwave Symp. Dig.*, Las Vegas, p. 333-336, 1987.
- Wang, F., Zhang, Q. J., Knowledge based neuromodels for microwave design, *IEEE Trans. Microwave Theory and Techniques*, v. MTT-45, p. 2333-2343, 1997.

Wang, F., Knowledge Based Neuromodels for Microwave Modeling and Design, Ph. D. Thesis, Department of Electronics, Carleton University, Ottawa, Canada, 1998.

Wang, F., et al., Neural network structures and training algorithms for microwave applications, *International Journal of RF and Microwave Design*, v. 9, p. 216-240, 1999.

Washington, G., Aperture antenna shape prediction by feedforward networks. *IEEE Trans. Antennas Propagat.*, v. AP-45, Abr., 1997.

Watson, P. M., Gupta, K. C., EM-ANN models for microstrip vias and interconnects in datasets circuits. *IEEE Trans. Microwave Theory and Techniques*, v. MTT-44, p. 2495-2503, Dez., 1996.

Watson, P. M., Gupta, K. C., e Mahajan, R. L., Development of knowledge based artificial neural networks models for microwave components, *IEEE MTT-S Int. Microwave Symp. Dig.*, Baltimore, p. 9-12, 1998.

Werbos, P., Beyond regression: new tools for prediction and analysis in the behavior sciences. Tese de Ph.D., Univ. de Harvard, Cambridge, Nov., 1974.

White, H., Gallant, A. R., Hornik, K., Stinchcombe, M., e Wooldridge, J., *Artificial Neural Networks: Approximation and Learning Theory*, Oxford, UK: Blackwell.

Widrow, B., Hoff, M.E., Adaptive switching circuits. *1960 IRE WESCON Conv. Rev.*, parte 4, p. 96-104, 1960.

Widrow, B., Lehr, M. A., 30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation. *Proceedings of the IEEE*, 78, p. 1415-1442, Set., 1990.

Widrow, B., Stearns, S. D., *Adaptive Signal Processing*. Englewood Cliffs: Prentice-Hall, 1985.

Williams, D. F., Marks, R. B., Phillips, K. R., e Miers, T., Progress toward MMIC on-wafer standards, *36th ARFTG Conf. Dig.*, p. 73-83, Nov., 1990.

Yamaguchi, K., Kodera, H., Optimum design of triode-like JFET's by two-dimensional computer simulation. *IEEE Trans. Electron Devices*, v. ED-24, p. 1061-1069, Ago., 1977.

Yamaguchi, K., Asai, S., e Kodera, H., Two-dimensional numerical analysis of stability criteria of GaAs MESFET's. *IEEE Trans. Electron Devices*, v. ED-23, p. 1283-1290, 1976.

Yang, H. D., Alexopoulos, N. G., e Yablonovitch, E., Photonic band-gap materials for high-gain printed circuit antennas, *IEEE Trans. Antennas Propagat.*, v. AP-45, p. 185-187, Jan., 1997.

- Zaabab, A. H., Zhang, Q. J., e Nakhla, M., A neural network modelling approach to circuit optimization and statistical design. *IEEE MTT-S Digest*, 1994.
- Zaabab, A. H., Zhang, Q. J., e Nakhla, M., A neural network modeling approach to circuit optimization and statistical design. *IEEE Trans. on Microwave Theory and Techniques*, v. MTT-43, p. 1349-1358, Jun., 1995.
- Zaabab, A. H., Zhang, Q. J., e Nakhla, M., Device and circuit-level modeling using neural networks with faster training based on network sparsity. *IEEE Trans. Microwave Theory and Techniques*, v. MTT-45, p. 1696-1704, Out., 1997.
- Zhang, Q. H., Using wavelet network in nonparametric estimation, *IEEE Trans. Neural Networks*, v. 8, p. 227-236, 1997.
- Zhang, Q. H., Benvensite, A., Wavelet networks, *IEEE Trans. Neural Networks*, v. 3, p. 889-898, 1992.
- Zhang, Q. J., Wang, F., e Nakhla, M., Optimization of high-speed VLSI interconnects: a review, *Int. Journal of Microwave and Millimeter-Wave CAE*, v. 7, p. 83-107, 1997a.
- Zhang, Q. J., et al., Ultra fast neural models for analysis of electro.opto interconnects, *IEEE Electronic Components and Technology Conference*, San Jose, CA, p. 1134-1137, Mai., 1997b.
- Zhang, Q. J., Creech, G. L., (Guest Editors), Special Issue on Applications of Artificial Neural Networks to RF and Microwave Design, *International Journal of RF and Microwave Computer-Aided Engineering*, v. 9, NY: Willey, 1999.
- Zhang, Q. J., Gupta, K. C., *Neural Network for RF and Microwave Design*, Norwood, MA: Artech House, 2000.

APÊNDICE

MÉTODOS ITERATIVOS PARA A MINIMIZAÇÃO DE FUNÇÕES

O conhecimento de métodos iterativos para a minimização de funções teve uma importância fundamental no desenvolvimento desta tese, sendo aplicados aos diversos problemas de otimização encontrados. Por exemplo, as estratégias de aprendizado pela correção do erro (para o treinamento supervisionado das ANNs), que visam minimizar algum funcional (função erro, objetivo, custo, etc.), fundamentam-se em tais métodos. Neste apêndice, procede-se com uma descrição breve e informal, na qual, em particular, aborda-se os princípios dos seguintes métodos: do gradiente, do gradiente conjugado, de Fletcher-Powell, de Newton-Raphson e de Newton.

A localização do mínimo local irrestrito de uma função de várias variáveis é um problema importante na solução de sistemas de equações algébricas não-lineares. Aqui, define-se uma função de n variáveis, cujo valor, $f(\mathbf{x})$, e o vetor gradiente, $\nabla f(\mathbf{x})$, podem ser calculados em qualquer ponto, \mathbf{x} . Convém classificar $f(\mathbf{x})$ conforme o cálculo do vetor gradiente: analítico (exato) ou numérico (estimado através de diferenças de $f(\mathbf{x})$).

Estes métodos são recursivos, no sentido que, a partir de uma aproximação (arbitrária) inicial, a solução é refinada com o aumento do número de iterações. De um modo geral, todo procedimento de minimização iterativo, com convergência quadrática ou não, localiza o mínimo de um funcional $f(\mathbf{x})$ como o limite de uma seqüência $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots)$, em que, \mathbf{x}_0 é a aproximação inicial para a posição do mínimo. A posição \mathbf{x}_{n+1} , para cada iteração $n \geq 0$, é obtida a partir da reta que passa por \mathbf{x}_n em alguma direção de pesquisa $p(n)$, Fletcher e Reeves (1964).

A.1 Método do Gradiente

Talvez, o método do gradiente seja a mais simples e antiga das técnicas numéricas disponíveis para a minimização de funções (que faz uso de suas derivadas). Na n -ésima iteração, a direção de pesquisa é simplesmente a direção negativa do gradiente de $f(\mathbf{x}_n)$, que indica a maior taxa de decréscimo deste funcional em \mathbf{x}_n . Para o caso bidimensional do método do gradiente, a reta tangente no ponto \mathbf{x}_n , determina a direção para o cálculo da próxima aproximação \mathbf{x}_{n+1} para o mínimo. Em termos matemáticos:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta_G \nabla f(\mathbf{x}_n) \quad (\text{A1})$$

em que, η_G é uma constante que determina a amplitude do ajuste aplicado à cada direção de pesquisa. A aproximação iterativa do mínimo, em geral, segue uma trajetória em ziguezague. Uma boa escolha de η_G é crucial para uma boa convergência do método.

A.2 Método do Gradiente Conjugado

O método do gradiente conjugado é um procedimento de minimização simples, que, no seu ciclo iterativo, admite o cálculo de novas direções de pesquisa. Desta forma, o conhecimento do comportamento local da função pode ser levado em conta, evitando-se o caminho em ziguezague, quando a direção de pesquisa é dada a priori.

Este método foi aplicado pela primeira vez para o problema de minimização sem restrições por Fletcher e Reeves (1964); em que, para funções quadráticas de n variáveis, o mínimo é localizado no máximo em n iterações. Para funções não-quadráticas, o procedimento é iterativo e requer um critério de convergência (parada). No método do gradiente conjugado o ajuste é definido de acordo com a regra:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta_G(n) \mathbf{p}_S(n) \quad (\text{A2})$$

O vetor direção inicial é a mesma do vetor gradiente, mas em sentido oposto:

$$\mathbf{p}_s(0) = -\nabla f(0) \quad (\text{A3})$$

Cada novo vetor direção é calculado recursivamente como uma combinação linear do vetor gradiente e do vetor direção na iteração prévia, sendo expresso por:

$$\mathbf{p}_s(n+1) = -\mathbf{g}_s(n+1) + \chi(n)\mathbf{p}_s(n) \quad (\text{A4})$$

$$\mathbf{g}_s(n) = \nabla f(n) \quad (\text{A5})$$

em que, $\chi(n)$ é um parâmetro variável no tempo. Existem várias regras para a determinação de $\chi(n)$. Dentre elas, destaca-se aquela atribuída à Fletcher e Reeves (1964):

$$\chi(n) = \frac{\mathbf{g}'_s(n+1)\mathbf{g}_s(n+1)}{\mathbf{g}'_s(n)\mathbf{g}_s(n)} \quad (\text{A6})$$

e aquela proposta por Polak e Ribière (1969),

$$\chi(n) = \frac{\mathbf{g}'_s(n+1)[\mathbf{g}_s(n+1) - \mathbf{g}_s(n)]}{\mathbf{g}'_s(n)\mathbf{g}_s(n)} \quad (\text{A7})$$

O seguinte algoritmo de minimização foi proposto por Fletcher e Reeves (1964):

\mathbf{x}_0 = valor arbitrário

$$\mathbf{g}_s(0) = \mathbf{g}_s(\mathbf{x}_0), \quad \mathbf{p}_s = -\mathbf{g}_s(0)$$

\mathbf{x}_{n+1} = posição do mínimo de $f(\mathbf{x})$ sobre a

a linha através de \mathbf{x}_n na direção $\mathbf{p}_n = \mathbf{p}_s(n)$. (A8)

$$\mathbf{g}_{n+1} = \mathbf{g}_s(\mathbf{x}_{n+1})$$

$$\chi_n = \mathbf{g}_{n+1}^2 / \mathbf{g}_n^2$$

$$\mathbf{p}_{n+1} = -\mathbf{g}_{n+1} + \chi_n \mathbf{p}_n$$

Quatro pontos principais requerem atenção na aplicação deste algoritmo:

- i. A escolha de \mathbf{x}_0 .
- ii. O tipo de pesquisa sucessiva para localizar cada \mathbf{x}_{n+1} .
- iii. A taxa de convergência.
- iv. O critério de convergência

Em particular, um método de pesquisa linear foi proposto por Davidon (1959), sendo empregado por Fletcher e Powell (1963). O problema é determinar t_m , tal que,

$$\frac{dy(\eta_G^m)}{d\eta_G} = 0 \quad (\text{A9})$$

em que,
$$y(\eta_G) = f(\mathbf{x}_n + \eta_G \mathbf{p}_n) \quad (\text{A10})$$

e portanto
$$\frac{dy(\eta_G)}{d\eta_G} = \mathbf{p}'_n \mathbf{g}_S(\mathbf{x}_n + \eta_G \mathbf{p}_n) \quad (\text{A11})$$

O cálculo de η_G é feito em três etapas: a primeira estima a sua ordem de magnitude; a segunda estabelece limites sobre esta estimativa; a terceira interpola seu valor. Conforme Fletcher e Powell (1963), inicialmente escolhe-se o seguinte valor:

$$\eta_G(0) = \begin{cases} \eta_{Gi}, & 0 < \eta_{Gi} < 1/\sqrt{\mathbf{p}'_n \mathbf{p}_n} \\ 1/\sqrt{\mathbf{p}'_n \mathbf{p}_n}, & \text{c.c.} \end{cases} \quad (\text{A12})$$

$$\eta_{Gi} = 2(\text{est} - f_n) / \mathbf{p}'_n \mathbf{g}_n \quad (\text{A13})$$

em que, *est* é uma estimativa disponível para o mínimo de *f*; por exemplo, *est* = 0 no cálculo de mínimos quadráticos. Na segunda etapa, a derivada $dy(\eta_G)/d\eta_G$ é examinada nos pontos $\eta_G = 0, h, 2h, 4h, \dots, \eta_a, \eta_b$, em que, $h = \eta_G(0)$, e, η_b é o primeiro valor para o

qual ocorre $dy(\eta_G)/d\eta_G \geq 0$, ou, para o qual a função não tem decrescido. Portanto, η_G^m é limitado no intervalo $\eta_a < \eta_G^m \leq \eta_b$. Na terceira etapa, efetua-se a interpolação cúbica proposta por Davidon (1959) no cálculo da estimativa η_G^e de η_G^m :

$$\eta_G^e = \eta_b - \frac{dy(\eta_b)/d\eta_b + w - z}{dy(\eta_b)/d\eta_b - dy(\eta_a)/d\eta_a + 2w}(\eta_b - \eta_a) \quad (\text{A14})$$

em que,
$$z = 3 \frac{y(\eta_a) - y(\eta_b)}{\eta_b - \eta_a} + dy(\eta_a)/d\eta_a + dy(\eta_b)/d\eta_b \quad (\text{A15})$$

$$w = \left[z^2 - dy(\eta_a)/d\eta_a dy(\eta_b)/d\eta_b \right]^{1/2} \quad (\text{A16})$$

Se $y(\eta_a)$ e $y(\eta_b)$ não forem menores que $y(\eta_G^e)$, então, η_G^e é aceito como uma estimativa para η_G^m . Caso contrário, conforme $dy(\eta_G^e)/d\eta_G^e$ seja positivo ou negativo, a interpolação deve ser repetida no subintervalo (η_a, η_G^e) ou (η_G^e, η_b) , respectivamente.

A.3 Método de Fletcher-Powell

Provavelmente o mais poderoso dos métodos de gradiente baseados em direções de pesquisa é a reformulação de Fletcher e Powell (1963), do procedimento proposto por Davidon (1959). O vetor direção é definido por:

$$p_n = -G_n g_n \quad (\text{A17})$$

em que, G_0, G_1, G_2, \dots é uma seqüência de matrizes definidas-positivas, e, G_0 normalmente é a matriz identidade. Para o cálculo desta seqüência, define-se:

$$G_{n+1} = G_n + \frac{p_n p_n'}{p_n' q_n} - \frac{G_n q_n q_n' G_n}{q_n' G_n q_n} \quad (\text{A18})$$

em que, $q_n = g_{n+1} - g_n$ (A19)

A aproximação para o mínimo é calculada como em (A2), sendo a ótima amplitude η_G^m de passo determinada por intermédio de pesquisa linear sucessiva, (A9)-(A16).

A.4 Método de Newton-Raphson

Este método iterativo procura solucionar um sistema de n equações não-lineares em n variáveis,

$$f(x) = 0 \quad (A20)$$

através de sucessivas linearizações. Considerando-se apenas o primeiro termo da expansão de $f(x)$ em série de Taylor, obtém-se:

$$\Delta f(x) = f(x + \Delta x) - f(x) \approx J(x)\Delta x \quad (A21)$$

Na pesquisa da solução considera-se que

$$f(x) + J(x)\Delta x = 0 \quad (A22)$$

Assim, a iteração de Newton-Raphson é dada por Carnahan (1969) e Ortega (1970):

$$\Delta x_n = -J_n^{-1} f(x_n) \quad (A23)$$

ou, $J_n \Delta x_n = -f(x_n)$ (A24)

em que,

$$\mathbf{J} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \equiv \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (\text{A25})$$

é a matriz Jacobiana. Então, através da solução do sistema (A24) por fatoração LU, determina-se as aproximações sucessivas para a solução de (A20).

A.5 Método de Newton

No método de Newton, a matriz Hessiana desempenha um papel fundamental. Usando-se a expansão em série de Taylor para um funcional $f(\mathbf{x})$, aproxima-se uma mudança incremental $\Delta f(\mathbf{x})$ como uma função quadrática de $\Delta \mathbf{x}$,

$$\Delta f(\mathbf{x}) = f(\mathbf{x} + \Delta \mathbf{x}) - f(\mathbf{x}) \approx \mathbf{g}(\mathbf{x})\Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}' \mathbf{H}(\mathbf{x}) \Delta \mathbf{x} \quad (\text{A26})$$

em que,

$$\mathbf{H} \equiv \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2} \equiv \begin{bmatrix} \frac{\partial^2 f_1}{\partial^2 x_1} & \frac{\partial^2 f_1}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f_1}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f_2}{\partial x_2 \partial x_1} & \frac{\partial^2 f_2}{\partial^2 x_2} & \dots & \frac{\partial^2 f_2}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial^2 f_n}{\partial x_n \partial x_1} & \frac{\partial^2 f_n}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f_n}{\partial^2 x_n} \end{bmatrix} \quad (\text{A27})$$

é a matriz Hessiana. Diferenciando-se (A26) com relação a $\Delta \mathbf{x}$, a mudança $\Delta f(\mathbf{x})$ é minimizada quando:

$$\mathbf{g} + \mathbf{H}\Delta \mathbf{x} = \mathbf{0} \quad (\text{A28})$$

Portanto, dada uma estimativa $\mathbf{x}(0)$ da solução ótima para a minimização de $f(\mathbf{x})$, cada nova estimativa do mínimo é dada através da expressão recursiva:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{H}(\mathbf{x}_n)\mathbf{g}(\mathbf{x}_n) \quad (\text{A29})$$

A.6 Exemplo Numérico

Para se realizar uma comparação entre os métodos descritos, emprega-se a função de Rosenbrock (1960) em duas dimensões,

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1^2)^2 \quad (\text{A30})$$

Esta função é difícil de se minimizar devido ao declive abrupto ao lado do seu vale, seguindo a curva $x_2^2 = x_1$, Fig. A1(b). O contorno do vale desta função tem o formato de uma banana, Fig. A1(b). Através da observação das trajetórias descritas, pode-se comparar as características de convergência de cada método, Fig. A2.

Para o método do gradiente, observa-se uma trajetória em ziguezague a partir do ponto $(-1, 2)$, que realiza um movimento Browniano em torno do mínimo; enquanto isso, a trajetória partindo de $(-1, 0)$ segue uma curva de nível, mas não converge, Fig. A2(a). Os métodos do gradiente com pesquisa são mais eficientes, pois evitam esta trajetória em ziguezague, Fig. A2(b)-(c). Com convergência bastante acelerada, observa-se movimentos bruscos nas trajetórias do método de Newton, Fig. A2(d).

APÊNDICE

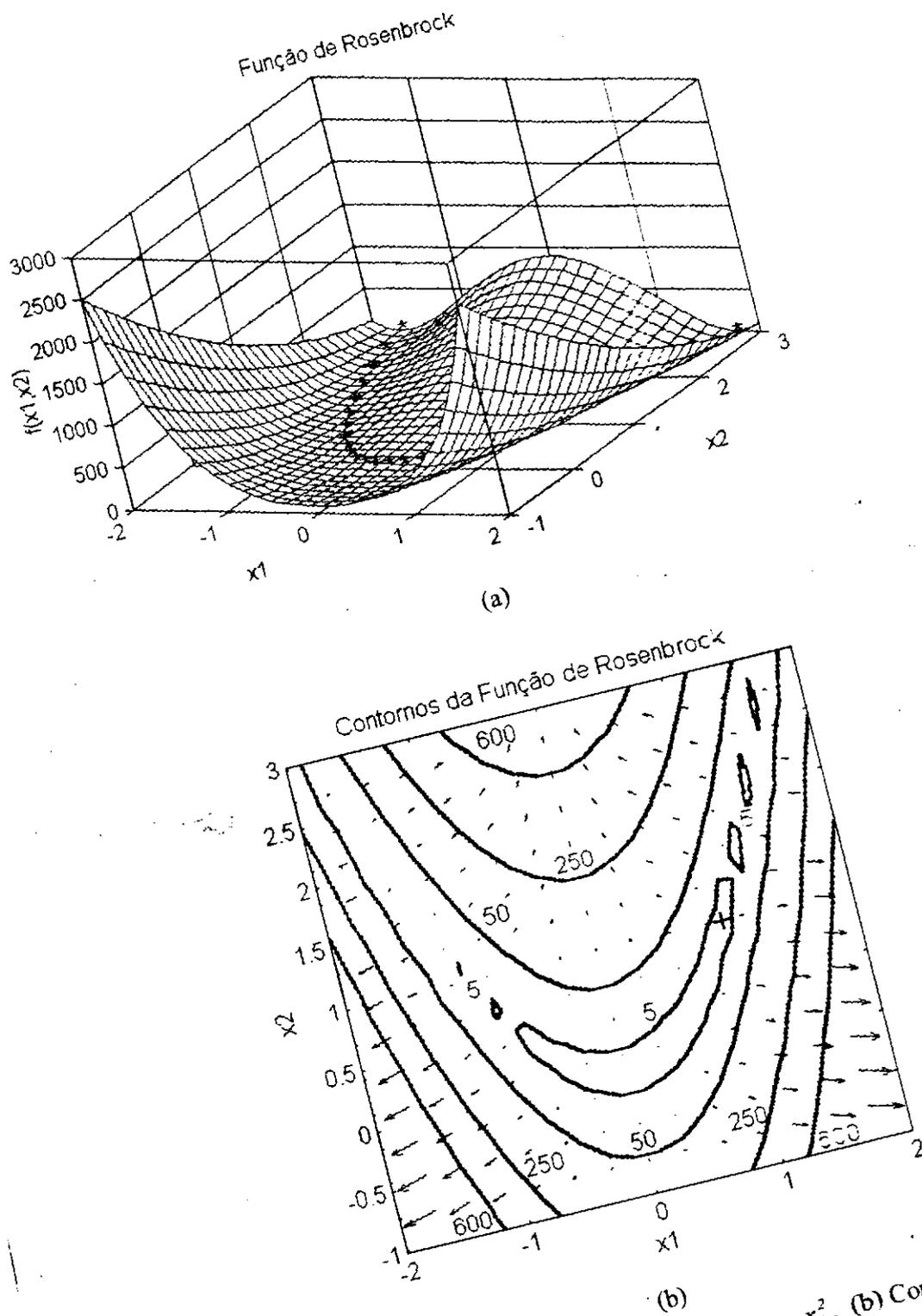


Fig. A1 (a) Função de Rosenbrock, (*) curva $x_2 = x_1^2$. (b) Contornos de nível, direções do vetor gradiente, (+) lugar do mínimo: $x = (1, 1)$.

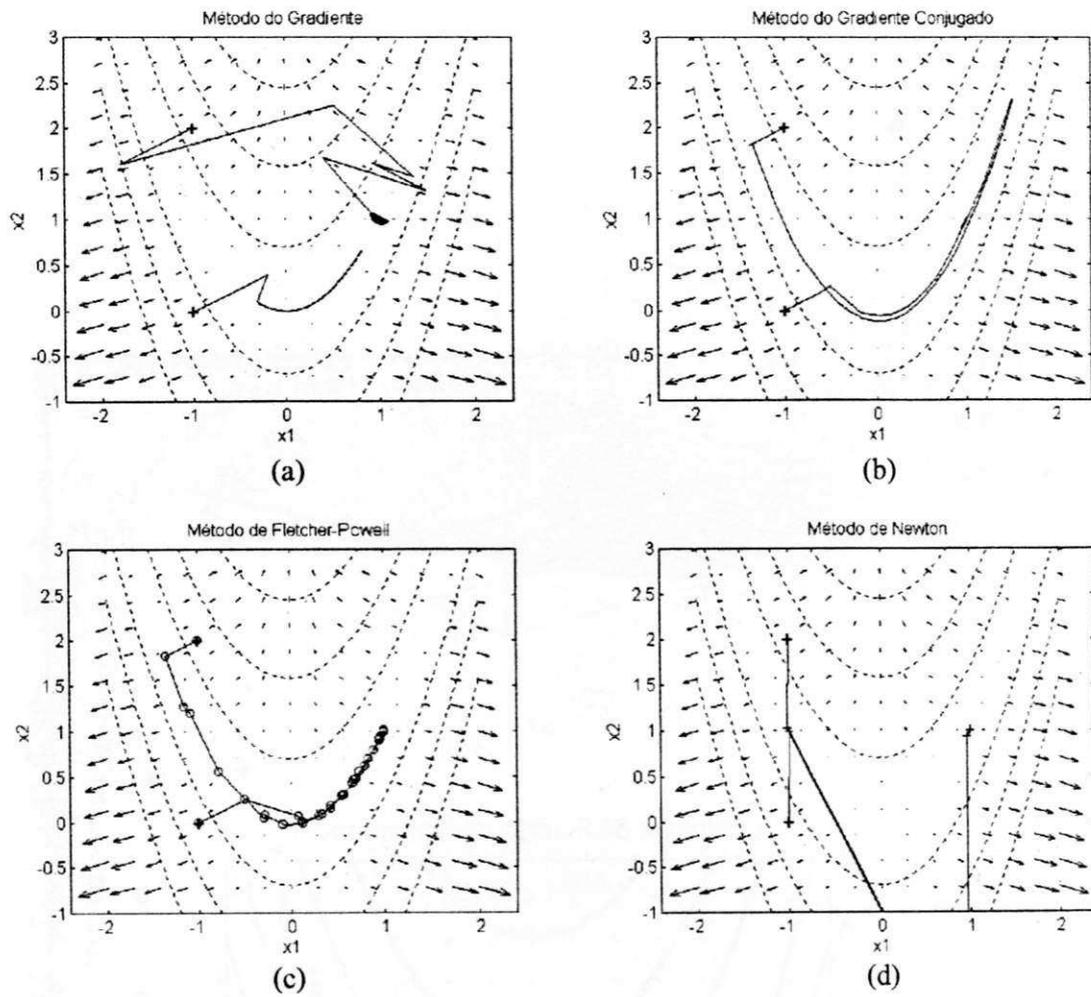


Fig A2. Minimização da função de Rosenbrock; trajetórias a partir dos pontos $(-1, 0)$ e $(-1, 2)$.
MÉTODOS – (a) Gradiente (1000 iterações), $\eta_G = 0,002$. (b) Gradiente conjugado (70 iterações) usando até 100 pesquisas de Fibonacci por iteração. (c) Fletcher-Powell (22 e 17 iterações, respectivamente). (d) Newton (7 iterações).