

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Proposta de Implementação de Redes de Base Radial em
Tecnologias CMOS e BiCMOS

MARCIO BARBOSA LUCKS

Orientador: Prof. Dr. Nobuo Oki

Tese apresentada à Faculdade de
Engenharia - UNESP – Campus de Ilha
Solteira, para obtenção do título de
Doutor em Engenharia Elétrica.

Área de Conhecimento: Automação.

Ilha Solteira – SP
março/2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

*À minha esposa Patrícia,
por seu amor e apoio
em todos os momentos.*

Agradecimentos

Gostaria de agradecer a todos aqueles que de alguma maneira me apoiaram e me incentivaram no transcorrer deste trabalho. Inicialmente, ao meu orientador, Prof. Dr. Nobuo Oki, que sempre me incentivou e é para mim um exemplo de dedicação e sabedoria, sempre me inspirando a trilhar caminhos novos e a buscar novos desafios nos campos acadêmico e profissional. À minha esposa Patrícia, por seu amor, por seu apoio e, principalmente, pela sua presença em minha vida. Aos meus pais, Joaquim Eleutério Lucks e Maria Aparecida Barbosa Lucks, que me proporcionaram a base sobre a qual pude me desenvolver e à minha irmã Marcia e ao meu irmão Maciel pelo apoio e amizade. À UNESP e ao Departamento de Engenharia Elétrica; em especial, aos professores Francisco Villarreal Alvarado, Anna Diva Plasencia Lotufo, Suely Cunha Amaro Mantovani, Carlos Roberto Minussi e Marcelo Carvalho Minhoto Teixeira pelas valiosas contribuições e sugestões para o aperfeiçoamento deste trabalho, aos técnicos Adérson e Everaldo, do Laboratório de Eletrônica pelo auxílio na parte prática da pesquisa e também a todos os funcionários e colegas do departamento. Ao Prof. Rubén e aos demais integrantes do Conselho de Curso do PPGEE pelo apoio que me deram e à Seção de Pós-Graduação, à Onilda e à Marcia, pela cordialidade e atenção que sempre nos dispensaram. À CAPES pelo apoio financeiro no início da pesquisa; e ao Instituto de Aeronáutica e Espaço (IAE) e à Divisão de Eletrônica (AEL) pelo apoio para a conclusão deste trabalho. Aos colegas do IAE que sempre me incentivaram e me apoiaram na realização desta pesquisa, o meu muito obrigado. Em especial, agradeço a Deus, que nos inspira, protege e orienta de várias maneiras, dando-nos as condições para a realização deste trabalho.

Resumo

Neste trabalho, apresenta-se o desenvolvimento de redes de base radial em tecnologia CMOS. Para tanto, dois circuitos unidimensionais, denominados RBF_1 e RBF_2 , são propostos. Sua funcionalidade é demonstrada por meio de simulações SPICE e também pela sua implementação prática com a utilização de conjuntos de MOSFETs presentes em circuitos integrados comerciais. Demonstra-se também o desenvolvimento dos circuitos para o caso multidimensional, com o uso de simulações SPICE e a implementação de um circuito RBF_1 bidimensional. Além disso, são apresentadas versões dos circuitos RBF_1 e RBF_2 para tecnologia BiCMOS. Os circuitos propostos são utilizados no projeto de redes de base radial bidimensionais em processo CMOS AMS 0.35 μm . No intuito de testar sua funcionalidade, as redes foram simuladas para algumas aplicações, apresentando bons resultados.

A questão da quantização no armazenamento dos parâmetros das redes de base radial e da sua influência na aproximação de funções também é tratada na tese. Foram realizadas várias simulações com diferentes níveis de quantização para algumas tarefas de aproximação de funções. Os resultados obtidos mostram que, mesmo com uma quantização severa, as redes apresentam a capacidade de aproximação de funções, porém de maneira limitada, ou seja, o erro obtido aumenta para um número menor de níveis de quantização. A quantização dos parâmetros diminui o tamanho da memória e a complexidade necessária para armazenar os parâmetros das redes, permitindo a implementação de circuitos compactos e adequados para aplicações de baixo consumo de potência.

Abstract

In this work, we present the development of radial basis function circuits in CMOS technology. Two one-dimensional circuits, namely RBF_1 and RBF_2 , are proposed for radial basis function realization, and their functionality is demonstrated by SPICE simulations and by their implementation with commercial MOSFET array integrated circuits. Multidimensional capability is demonstrated by the implementation of a bidimensional RBF_1 circuit and by SPICE simulation results. In addition, BiCMOS versions are also presented for RBF_1 and RBF_2 . The proposed cells are used in the design of bidimensional radial basis function neural networks in AMS 0.35 μ m CMOS process. In order to test their functionality, the networks were simulated for some applications with good results achieved.

The issue of parameter quantization and its influence on the network function approximation capability is also dealt with in this dissertation. We carried out several simulations with different levels of quantization. The results obtained show that the network presents a capability of learning functions, even with a severe parameter quantization. As expected, the error increases for less bits of quantization. Nevertheless, the parameter quantization decreases the memory size and complexity necessary for network parameter storage, allowing the implementation of compact circuits and being adequate for low power applications.

Lista de Figuras

- 1.1a: Conexão entre dois neurônios em uma rede neural biológica.
- 1.1b: Membrana celular. (Adaptado de Dayan e Abbott, 2000).
- 1.2: Potencial de ação (GREENSTEIN e GREENSTEIN, 2000).
- 1.3: Modelo do neurônio de McCulloch-Pitts.
- 1.4: Função de ativação degrau.
- 1.5: MLP com três camadas.
- 2.1: Diagrama esquemático de uma RBFN padrão.
- 2.2: Função gaussiana com parâmetros $a = 1$, $b = 1$ e $c = (0,0)$.
- 2.3: Função Gaussiana bidimensional.
- 2.4a: Função SHQ com parâmetros $a = 1$, $b = 1$ e $c = 0$.
- 2.4b: Função SHQ bidimensional com parâmetros $a = 1$, $b = 1$ e $c = (0,0)$.
- 2.5: Esquema para aprendizado supervisionado da rede.
- 3.1: Função seno e a aproximação da rede.
- 3.2: Função seno e aproximação da rede com parâmetros quantizados com 8 bits (+).
- 3.3: Erro entre a função seno e as saídas da rede contínua (-) e da rede com parâmetros quantizados com 8 bits (o).
- 3.4: Função seno e aproximação da rede com parâmetros quantizados com 5 bits (+).
- 3.5: Erro entre a função seno e as saídas da rede contínua (-) e da rede com parâmetros quantizados com 5 bits (o).
- 3.6: Função seno e aproximação da rede com parâmetros quantizados com 3 bits (+).
- 3.7: Erro entre a função seno e as saídas da rede contínua (-) e da rede com parâmetros quantizados com 3 bits (o).
- 3.8: Erro quadrático médio entre a função seno e a saída da rede com parâmetros quantizados em função do número de bits de quantização.
- 3.9: Erro quadrático médio entre a função seno e a saída da rede com parâmetros quantizados em função do número de bits de quantização (escala logarítmica).
- 3.10: O polinômio de Hermite H_6 e a aproximação da rede contínua (-) e a da rede com parâmetros quantizados com 8 bits (+).
- 3.11: Erro entre o polinômio de Hermite H_6 e as aproximações da rede contínua (-) e da rede com parâmetros quantizados com 8 bits (o).
- 3.12: O polinômio de Hermite H_6 e a aproximação da rede contínua (-) e a da rede com parâmetros quantizados com 5 bits (+).
- 3.13: Erro entre o polinômio de Hermite H_6 e as aproximações da rede contínua (-) e da rede com parâmetros quantizados com 5 bits (o).

- 3.14: O polinômio de Hermite H_6 e a aproximação da rede contínua e a da rede com parâmetros quantizados com 3 bits.
- 3.15: Erro entre o polinômio de Hermite H_6 e as aproximações da rede contínua (-) e da rede com parâmetros quantizados com 3 bits (o).
- 3.16: Saída desejada.
- 3.17: Saída inicial da RBFN.
- 3.18: Variação do erro quadrático da RBFN durante a fase de aprendizado.
- 3.19: Aproximação da RBFN após o aprendizado.
- 3.20: Erro entre a função desejada e a aproximação da RBFN.
- 3.21: Aproximação da QP-RBFN.
- 3.22: Erro entre a função desejada e a aproximação da QP-RBFN.
- 3.23: Diferença entre as aproximações RBFN e QP-RBFN.
- 4.1: Esquema simplificado do par diferencial bipolar.
- 4.2: Curva característica do par diferencial bipolar.
- 4.3: Curva de transcondutância do par diferencial bipolar.
- 4.4: Esquema simplificado do par diferencial NMOS.
- 4.5: Curva característica do par diferencial NMOS.
- 4.6: Curva de transcondutância do par diferencial NMOS.
- 4.7: Esquema simplificado do circuito RBF₁ CMOS.
- 4.8: Corrente de saída do circuito CMOS RBF₁.
- 4.9: Corrente de saída em função da variação de $(V_{b1} - V_{b2})$ de 100 mV a 1 V.
- 4.10: Variação no centro da RBF em função da variação de V_{b1} e V_{b2} .
- 4.11: Corrente de saída em função da variação de $I_{b1}=I_{b2}$ entre 1 μ A e 6 μ A para $V_{b1}=50$ mV e $V_{b2}=-50$ mV.
- 4.12: Corrente de saída em função da variação de $I_{b1}=I_{b2}$ entre 1 μ A e 6 μ A para $V_{b1}=100$ mV e $V_{b2}=-100$ mV.
- 4.13: Circuito RBF₁ Bidimensional.
- 4.14: Corrente de saída do circuito RBF₁ CMOS bidimensional.
- 4.15: Esquema simplificado do circuito RBF₁ BiCMOS.
- 4.16: Corrente de saída do circuito RBF₁ BiCMOS.
- 4.17: Esquema simplificado do circuito CMOS RBF₂.
- 4.18: Corrente de saída do circuito CMOS RBF₂.
- 4.19: Corrente de saída em função da variação de $(V_{b1} - V_{b2})$ de 100 mV a 1 V.
- 4.20: Variação no centro da RBF em função da variação de V_{b1} e V_{b2} .
- 4.21: Corrente de saída em função da variação de $I_{b1}=I_{b2}$ entre 1 μ A e 6 μ A para $V_{b1}=50$ mV e $V_{b2}=-50$ mV.

- 4.22: Corrente de saída em função da variação de $I_{b1}=I_{b2}$ entre $1\ \mu\text{A}$ e $6\ \mu\text{A}$ para $V_{b1}=100\ \text{mV}$ e $V_{b2}=-100\ \text{mV}$.
- 4.23: Circuito RBF₂ Bidimensional.
- 4.24: Corrente de saída do circuito RBF₂ CMOS bidimensional.
- 4.25: Circuito para variação de centro e largura.
- 5.1: Montagem experimental do circuito CMOS RBF₁. (a) Detalhe do circuito.
(b) Implementação de dois circuitos RBF₁.
- 5.2: Resultados experimentais obtidos para o circuito CMOS RBF₁.
- 5.3: Montagem experimental para os dois circuitos RBF₁.
- 5.3: Resultados experimentais obtidos para o circuito CMOS RBF₁ mostrando a onda triangular na entrada e a função de base radial obtida.
- 5.4: (a) Resultados experimentais obtidos para os dois circuitos RBF₁ conectados em paralelo. (b) Ampliação da escala de tempo. (c) Saída do circuito 1.
(d) Saída do circuito 2.
- 5.5: Resultados obtidos para saídas conectadas e R_L em aberto.
- 5.6: Aproximação de uma senóide.
- 5.7: Aproximação de uma senóide: circuito 1.
- 5.8: Aproximação de uma senóide: circuito 2.
- 5.9: Resultados obtidos para variações nas tensões de polarização V_{b1} e V_{b2} .
- 5.10a: Resultados obtidos para o circuito RBF₁ bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 10\ \text{k}\Omega$.
- 5.10b: Resultados obtidos para o circuito RBF₁ bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 10\ \text{k}\Omega$ (continuação).
- 5.10c: Resultados obtidos para o circuito RBF₁ bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 10\ \text{k}\Omega$ (continuação).
- 5.11a: Resultados obtidos para o circuito RBF₁ bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 100\ \text{k}\Omega$.
- 5.11b: Resultados obtidos para o circuito RBF₁ bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 100\ \text{k}\Omega$ (continuação).
- 5.12a: Resultados obtidos para o circuito RBF₁ bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = \infty$ (aberto).
- 5.12b: Resultados obtidos para o circuito RBF₁ bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = \infty$ (aberto) (continuação).
- 5.12c: Resultados obtidos para o circuito RBF₁ bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = \infty$ (aberto) (continuação).

- 5.13a: Resultados obtidos para o circuito RBF₁ bidimensional para Vi2 triangular e variações na tensão Vi1 com $R_L = \infty$ (aberto).
- 5.13b: Resultados obtidos para o circuito RBF₁ bidimensional para Vi2 triangular e variações na tensão Vi1 com $R_L = \infty$ (aberto) (continuação).
- 5.13c: Resultados obtidos para o circuito RBF₁ bidimensional para Vi2 triangular e variações na tensão Vi1 com $R_L = \infty$ (aberto) (continuação).
- 5.14: Montagem experimental para o circuito CMOS RBF₂.
- 5.15: Montagem experimental do circuito CMOS RBF₂ completo.
- 5.16: Resultados experimentais obtidos para o circuito CMOS RBF₂.
- 5.17a: Resultados obtidos para variações nas tensões de polarização Vb1 e Vb2 do circuito CMOS RBF₂ unidimensional básico.
- 5.17b: Resultados obtidos para variações nas tensões de polarização Vb1 e Vb2 do circuito CMOS RBF₂ unidimensional básico.
- 5.18a: Resultados obtidos para variações na corrente de polarização Ib1 do circuito CMOS RBF₂ completo.
- 5.18b: Resultados obtidos para variações na corrente de polarização Ib1 do circuito CMOS RBF₂ completo.
- 5.19a: Resultados obtidos para variações na tensão de centro Vc do circuito CMOS RBF₂ completo.
- 5.19b: Resultados obtidos para variações na tensão de centro Vc do circuito CMOS RBF₂ completo.
- 5.20: Resultados obtidos para variações nas tensões de polarização Vb1 e Vb2 do circuito CMOS RBF₂ unidimensional completo.
- 6.1: Arquitetura da rede bidimensional proposta.
- 6.2: Circuito RBF1 básico.
- 6.3: Ib2 variando de 1 μ A a 10 μ A passo=1 μ A. $R_L=1\Omega$. Vb1=+100mV e Vb2=-100mV.
- 6.4: Ib2 variando de 5 μ A a 25 μ A passo=5 μ A. $R_L=1\Omega$. Vb1=+100mV e Vb2=-100mV.
- 6.5: Variação de (Vb1-Vb2) de 100 mV a 1 V. Ib2=1 μ A.
- 6.6: Variação de centro da RBF₁ básica AMS035. Ib2=1 μ A. (Vb1-Vb2) de 100 mV.
- 6.7: Configuração utilizada nas análises de temperatura, Monte Carlo e pior caso.
- 6.8: Saída de RBF₁ obtida para temperatura variando de -50 a +125 °C.
- 6.9: RBF₁ - Análise de pior caso.
- 6.10: Análise Monte Carlo com variação de $\pm 1\%$ em L e $\pm 1\%$ em W.
- 6.11: Análise Monte Carlo com variação de $\pm 5\%$ em L e $\pm 5\%$ em W.
- 6.12: Análise Monte Carlo com variação de $\pm 1\%$ em VTH0.
- 6.13: Análise Monte Carlo com variação de $\pm 5\%$ em VTH0.

- 6.14: Análise Monte Carlo com variação de $\pm 1\%$ em U_0 .
- 6.15: Análise Monte Carlo com variação de $\pm 5\%$ em U_0 .
- 6.16: Análise Monte Carlo com variação de $\pm 1\%$ em TOX.
- 6.17: Análise Monte Carlo com variação de $\pm 5\%$ em TOX.
- 6.18: Análise Monte Carlo com variação combinada de V_{TH0} , U_0 , TOX, W e L em $\pm 1\%$.
- 6.19: Análise Monte Carlo com variação combinada de V_{TH0} , U_0 , TOX, W e L em $\pm 5\%$.
- 6.20: Circuito RBF_2 básico.
- 6.21: *Test bench* para o Circuito RBF_2 básico.
- 6.22: I_{b2} variando de $1\mu A$ a $10\mu A$ passo= $1\mu A$. $R_L=1\Omega$. $V_{b1}=+100mV$ e $V_{b2}=-100mV$.
- 6.23: I_{b2} variando de $5\mu A$ a $50\mu A$ passo= $5\mu A$. $R_L=1\Omega$. $V_{b1}=+100mV$ e $V_{b2}=-100mV$.
- 6.24: Variação de $(V_{b1}-V_{b2})$ de 100 mV a 1 V.
- 6.25: Variação de centro da RBF_2 básica AMS035.
- 6.26: Saída de RBF_2 obtida para temperatura variando de -50 a $+125$ °C.
- 6.27: Análise de pior caso.
- 6.28: Análise Monte Carlo com variação de $\pm 1\%$ em L e $\pm 1\%$ em W.
- 6.29: Análise Monte Carlo com variação de $\pm 5\%$ em L e $\pm 5\%$ em W.
- 6.30: Análise Monte Carlo com variação de V_{TH0} em $\pm 1\%$.
- 6.31: Análise Monte Carlo com variação de V_{TH0} em $\pm 5\%$.
- 6.32: Análise Monte Carlo com variação de U_0 em $\pm 1\%$.
- 6.33: Análise Monte Carlo com variação de U_0 em $\pm 5\%$.
- 6.34: Análise Monte Carlo com variação de TOX em $\pm 1\%$.
- 6.35: Análise Monte Carlo com variação de TOX em $\pm 5\%$.
- 6.36: Análise Monte Carlo com variação combinada de V_{TH0} , U_0 , TOX, W e L em $\pm 1\%$.
- 6.37: Análise Monte Carlo com variação combinada de V_{TH0} , U_0 , TOX, W e L em $\pm 5\%$.
- 6.38: Magnitude da corrente de saída em função da frequência (RBF_1).
- 6.39: Fase da corrente de saída em função da frequência (RBF_1).
- 6.40: Magnitude da corrente de saída em função da frequência (RBF_2).
- 6.41: Fase da corrente de saída em função da frequência (RBF_2).
- 6.42: Circuito RBF_1 bidimensional. (a) Esquema interno. (b) Símbolo do circuito RBF_1 bidimensional.
- 6.43: Circuito RBF_2 bidimensional. (a) Esquema interno. (b) Símbolo do circuito RBF_2 bidimensional.
- 6.44: (a) Corrente de saída e (b) regiões de contorno da RBF_1 básica bidimensional.
- 6.45: (a) Corrente de saída e (b) regiões de contorno da RBF_2 básica bidimensional.
- 6.46: Diagrama esquemático de uma RBF_N com duas entradas, uma saída e quatro bases RBF_1 bidimensionais na camada intermediária.

6.47: Diagrama esquemático de uma RBFN com duas entradas, uma saída e quatro bases RBF₂ bidimensionais na camada intermediária.

7.1: RBFN com duas bases que implementa uma função OU-Exclusivo.

7.2: Regiões do espaço de entrada em que as RBFs estão localizadas para a realização da função OU-Exclusivo.

7.3: Resultados de simulação para uma rede bidimensional.

7.4: Corrente de saída de uma RBFN RBF₂ bidimensional com 05 unidades.

7.5: Exemplo de aplicação em classificação de padrões não linearmente separáveis.

7.6: Análise DC.

7.7: Análise transiente.

7.8: FFT do sinal de saída da rede.

7.9: Sinal de saída da rede em relação à função senoidal

$$y = f(t) = 1,4195 \cdot 10^{-6} \cdot \text{sen}(8\pi t + 1,5) + 1,7364 \cdot 10^{-6}$$

7.10: Erro absoluto entre o sinal de saída da rede e a função senoidal.

7.11: Corrente de saída para $V_{b1} = 1.5 \text{ V}$ e $V_{b2} = 0 \text{ V}$.

7.12: Corrente de saída para $V_{b1} = 1.5 \text{ V}$ e V_{b2} entre -500 mV e $+500 \text{ mV}$.

7.13: Corrente de saída para $V_{b1} = 0 \text{ V}$ e $V_{b2} = -1.5 \text{ V}$.

7.14: Corrente de saída para V_{b1} entre -500 mV e $+500 \text{ mV}$ e $V_{b2} = -1.5 \text{ V}$.

Lista de Siglas e Abreviaturas

ACO:	Algoritmo colônia de formigas (<i>Ant Colony Optimization</i>);
AIS:	Sistemas imunológicos artificiais (<i>Artificial Immune System</i>);
AMS:	Austria Micro Systems;
BiCMOS:	Tecnologia Bipolar e MOS;
CI:	Circuito Integrado;
CM:	Espelho de Corrente;
CMOS:	Tecnologia MOS complementar;
FGMOS:	Transistor MOS de porta flutuante;
GA:	Algoritmo Genético;
IWN:	Rede com pesos inteiros (<i>Integer Weight Network</i>);
LMS:	Algoritmo Mínimos Quadrados;
MFN:	Rede sem multiplicador (<i>Multiplier Free Network</i>);
MLP:	Perceptron Multicamadas;
MOS:	Tecnologia Metal-Óxido-Semicondutor;
MOSFET:	Transistor de Efeito de Campo Metal-Óxido-Semicondutor;
OTA:	Amplificador operacional de transcondutância;
PSO:	Algoritmo enxame de partículas (<i>Particle Swarm Optimization</i>);
QP-RBFN:	RBFN com Parâmetros Quantizados;
RBF:	Função de Base Radial;
RBFN:	Rede Neural com Função de Base Radial;
RNA:	Rede Neural Artificial;
RP:	Algoritmo de Retropropagação do Erro;
SHQ:	Secante Hiperbólica Quadrática;
SPICE:	Simulation Program with Integrated Circuit Emphasis;
VLSI:	Very Large Scale Integration;

Lista de Símbolos

\mathbf{x} :	Variável de entrada da RNA;
\mathbf{y} :	Variável de saída da RNA;
M :	Número de RBFs;
N :	Número de pares-exemplos usados na fase de aprendizado da rede;
w_i :	Peso ou Amplitude da i -ésima RBF;
b_i :	Largura da i -ésima RBF;
c_i :	Centro da i -ésima RBF;
\mathbf{p}_i :	vetor de parâmetros de forma da i -ésima RBF;
d :	Máxima distância euclidiana entre os centros das RBFs;
p :	Parâmetro da rede;
Δp :	Alteração de um parâmetro p ;
μ :	Taxa de aprendizado;
E :	Erro quadrático;
β :	Ganho de corrente do transistor bipolar;
V_t :	Tensão térmica (V);
V_{dif} :	Tensão diferencial na entrada do par diferencial;
I_{dif} :	Corrente diferencial na saída do par diferencial;
I_t :	Corrente de polarização do par diferencial ou do OTA;
V_{DD} :	Tensão positiva da fonte de alimentação;
V_{SS} :	Tensão negativa da fonte de alimentação;
x_p :	Padrão de entrada usado no treinamento da RNA;
y_p :	Saída da RNA para um padrão x_p aplicado na entrada;
d_p :	Saída desejada para um padrão x_p aplicado à entrada da RNA;
\mathbb{R}^d :	Espaço vetorial real de dimensão d ;
$C(\mathbb{R}^d)$:	Espaço das funções contínuas em \mathbb{R}^d ;
g_m :	Transcondutância do par diferencial.
V_{TH0} :	Parâmetro da tensão de limiar de um MOSFET (modelo SPICE);
U_0 :	Parâmetro da mobilidade de portadores (modelo SPICE);
TOX :	Parâmetro da espessura do óxido da porta de um MOSFET (modelo SPICE);

- W: Largura do canal de um MOSFET
L: Comprimento do canal de um MOSFET

Lista de Tabelas

- 3.1: Erro absoluto obtido na aproximação da função seno.
- 3.2: Erro absoluto obtido na aproximação da função polinomial de Hermite.
- 3.3: Parâmetros da rede quantizada obtidos após o treinamento.
- 3.4: Erro absoluto obtido na aproximação da função trigonométrica bidimensional.
- 5.1: Polarização do circuito RBF₁ bidimensional.
- 6.1: Variação na amplitude da RBF₁ em função da variação dos parâmetros V_{TH0} , U_0 , TOX , W e L .
- 6.2: Variação na amplitude da RBF₂ em função da variação dos parâmetros V_{TH0} , U_0 , TOX , W e L .

Sumário

Introdução	18
Motivação	18
Estrutura da Tese	20
Contribuições da Tese	21
Parte I: Redes Neurais com Função de Base Radial	22
Capítulo 1: Redes Neurais Artificiais	23
1.1 Redes Neurais Biológicas	24
1.2 Redes Neurais Artificiais	26
1.3 Tipos de Aprendizado	29
1.3.1 Aprendizado supervisionado	29
1.3.2 Aprendizado não-supervisionado	29
1.4 Aplicações das RNAs	30
Capítulo 2: Redes de Base Radial	31
2.1 Fundamentos Teóricos	31
2.2 Função Gaussiana	33
2.3 Função Secante Hiperbólica Quadrática (SHQ)	34
2.4 Métodos de Aprendizado	36
2.4.1 Método Clássico	36
2.4.2 Método de Moody-Darken	37
2.4.3 Método do Gradiente	37
Capítulo 3: Redes de Base Radial com Parâmetros Quantizados	42
3.1 Redes Neurais Implementadas com Parâmetros Quantizados	42
3.2 Revisão da Literatura	43
3.3 Redes de Base Radial com Parâmetros Quantizados	45
3.4 Aplicações da QP-RBFN	47
3.4.1 Função Senoidal	47
3.4.2 Função Polinomial de Hermite	52
3.4.3 Função Trigonométrica Bidimensional	55

3.5 Conclusões	59
Parte II: Implementação Eletrônica de Redes de Base Radial	60
Capítulo 4: Propostas de Implementação Eletrônica de Redes de Base Radial	61
4.1 Tipos de Implementação	61
4.1.1 Implementação Analógica	61
4.1.2 Implementação Digital	62
4.1.3 Implementação Mista	63
4.2 Tipos de Aprendizado	63
4.3 Revisão Bibliográfica	63
4.4 Circuitos Propostos	67
4.4.1 Par diferencial bipolar	68
4.4.2 Par diferencial MOS	70
4.5 Circuito RBF ₁ CMOS	73
4.6 Circuitos RBF ₁ Multidimensionais	78
4.7 Circuito RBF ₁ BiCMOS	79
4.8 Circuito RBF ₂ CMOS	81
4.9 Circuitos RBF ₂ Multidimensionais	85
4.10 Circuito CMOS RBF ₂ Completo	86
Capítulo 5: Implementação Prática e Resultados Obtidos	87
5.1 Circuito RBF ₁	87
5.1.1 Circuito RBF ₁ Unidimensional	88
5.1.2 Dois Circuitos RBF ₁ Conectados em Paralelo	89
5.1.3 Aproximação de uma função senoidal	90
5.1.4 Variação de parâmetros da RBF ₁	92
5.2 Circuito RBF ₁ Bidimensional	93
5.3 Circuito RBF ₂	104
5.3.1 Variação de parâmetros da RBF ₂	105
5.4 Conclusões	111
Parte III: Proposta de Implementação de uma Rede de Base Radial em Tecnologia CMOS	112

Capítulo 6: Redes de Base Radial em Processo CMOS	113
6.1 Arquitetura da Rede	113
6.2 Circuito RBF ₁ básico	114
6.3 Análise do circuito RBF ₁ com relação a variações de processo e temperatura	117
6.3.1 Análise de temperatura	118
6.3.2 Análise de pior caso	118
6.3.3 Análise Monte Carlo	119
6.4 Circuito RBF ₂ básico	125
6.5 Análise do circuito RBF ₂ com relação a variações de processo e temperatura	128
6.5.1 Análise de temperatura	128
6.5.2 Análise de pior caso	129
6.5.3 Análise Monte Carlo	130
6.6 Análise AC	136
6.7 Circuito RBF ₁ bidimensional	138
6.8 Circuito RBF ₂ bidimensional	138
6.9 RBFN bidimensional	140
6.10 Conclusões	139
Capítulo 7: Resultados Obtidos para a RBFN CMOS	142
7.1 Classificação de Padrões	142
7.2 Aproximação de Funções	145
7.2.1 Função Senoidal	145
7.2.2 Funções de Pertinência para Lógica Nebulosa	148
Considerações Finais	150
Propostas para Trabalhos Futuros	151
Referências Bibliográficas	152
Apêndices	159
Anexos	176

Introdução

As Redes Neurais Artificiais (RNA) são estruturas computacionais inspiradas nas redes neurais biológicas que se têm mostrado promissoras na resolução de diversos problemas científicos e tecnológicos da atualidade. Elas se caracterizam pelo paralelismo de unidades simples e são capazes de modelar funções não-lineares complexas e com múltiplos extremos, utilizando valores da função desconhecida para aprendê-la. O uso de uma RNA se justifica em dois casos. O primeiro é se ela apresentar melhor desempenho ou menor custo quando comparada a métodos convencionais. O segundo, se ela tornar possível a solução de um problema difícil, para o qual não haja outra solução. Entre os diversos tipos de RNAs, a mais difundida é o Perceptron multicamadas (MLP), que utiliza funções sigmoidais na camada intermediária e o algoritmo de retropropagação do erro (RP) na fase de aprendizado. Recentemente, a Rede Neural com Função de Base Radial (RBFN) tem sido extensivamente estudada. Embora seja um paradigma relativamente novo no campo das RNAs, com suas raízes nos trabalhos de Broomhead e Lowe (1988), Moody e Darken (1989) e Girosi e Poggio (1989), a RBFN tem sido aplicada com sucesso a uma variada gama de problemas tais como: reconhecimento e classificação de padrões (ZANCHETTIN e LUDERMIR, 2007; HAYKIN, 1994; ROSENBLUM, YACOOB e DAVIS, 1996), aproximação e linearização de funções (LENDASSE, 2003, LUCKS, 1998), controle automático (DONG, 2006; ZHAO, 2008), tratamento e melhoramento de sinais (LIN et al., 2007) e modelamento e otimização de sistemas (MCDONALD et al, 2007).

Motivação

Uma vez que a abordagem com redes neurais se aplique a um determinado problema, é preciso definir como será realizada sua implementação. As RNAs podem ser

implementadas em software ou em hardware. Em geral, as implementações em software não conseguem aproveitar o paralelismo inerente às redes neurais. Em contrapartida, implementações em hardware podem atingir velocidades de operação maiores aproveitando esse paralelismo (DONG, 2006). Há várias formas de implementação eletrônica de RNAs, e particularmente, de RBFNs. Segundo Haykin (1994), a densidade de transistores e o baixo custo da tecnologia CMOS a tornam a principal tecnologia escolhida para a implementação VLSI de redes neurais, as quais podem ser divididas em digitais, analógicas ou mistas. A implementação digital apresenta maior imunidade a ruídos e a variações de processo. Além disso, permite uma maior flexibilidade na programação e na configuração da rede. No entanto, em aplicações que exigem maior rapidez de processamento ou interface com circuitos biológicos, a alternativa analógica torna-se atraente, visto que sua abordagem é, em geral, totalmente paralela, além de ser mais adequada para aplicações de baixo consumo de potência (LEHMANN, 1994). Dessa forma, a implementação mista surge naturalmente como uma forma de aproveitar as vantagens das duas abordagens, ou seja, a programabilidade das implementações digitais com a velocidade de processamento e a compacidade das implementações analógicas.

Em Lucks (1998), foram propostos dois circuitos para realização de funções de base radial utilizando transistores bipolares. Um dos circuitos foi implementado utilizando amplificadores operacionais de transcondutância (OTA), o qual apresentou um bom desempenho na realização das funções gaussiana e secante hiperbólica quadrática, e permitiu o ajuste de todos os parâmetros de forma dessas funções, quais sejam, amplitude, centro e largura. Esse circuito foi utilizado para implementar uma RBFN, a qual foi aplicada com sucesso à resolução de diversos problemas de aproximação de funções e linearização de circuitos não-lineares (LUCKS, 1998; LUCKS e OKI, 1999a, 1999b). A arquitetura proposta foi posteriormente implementada em circuito integrado em tecnologia BiCMOS (OLIVEIRA e OKI, 2001, 2002), tendo sido aplicada a algumas tarefas e a fase de aprendizado *chip-in-the-loop* realizada com sucesso com o uso de algoritmos genéticos (NAGASHIMA e OKI, 2005).

Esta tese enfoca o desenvolvimento de redes de base radial em tecnologia CMOS. Para tanto, são propostos dois circuitos unidimensionais, denominados RBF_1 e RBF_2 (do inglês, *radial basis function*), cujo funcionamento está baseado na aproximação da curva

de transcondutância de pares diferenciais, utilizando a técnica da aproximação da derivada. Sua funcionalidade é demonstrada por meio de simulações SPICE e também pela sua implementação prática, com o uso de conjuntos de transistores MOS presentes em circuitos integrados comerciais (*MOSFET arrays*). Demonstra-se também o desenvolvimento dos circuitos para o caso multidimensional, com o uso de simulações SPICE e a implementação de um circuito RBF₁ bidimensional. Além disso, são apresentadas versões dos circuitos RBF₁ e RBF₂ para tecnologia BiCMOS. Os circuitos propostos foram aplicados ao projeto de uma rede de base radial bidimensional em processo CMOS AMS 0.35 μ m.

Nesta tese, trata-se, também, da questão da quantização no armazenamento dos parâmetros das redes de base radial e sua influência na aproximação de funções. A quantização dos parâmetros diminui o tamanho da memória e a complexidade necessária para armazenar os parâmetros das redes, permitindo a implementação de circuitos mais compactos e mais adequados para aplicações de baixo consumo de potência. Porém, há uma diminuição na capacidade de aproximação das redes e não existe a garantia dada pela propriedade de aproximação universal encontrada nas redes com parâmetros contínuos.

Estrutura da Tese

A tese está dividida em três partes. A primeira parte trata da teoria das redes de base radial e possui três capítulos. No Capítulo 1 é feita uma introdução às Redes Neurais Artificiais de maneira geral. No Capítulo 2 são apresentadas as Redes de Base Radial e alguns algoritmos utilizados para a realização da fase de aprendizado. No Capítulo 3 são apresentadas as Redes de Base Radial com parâmetros quantizados e alguns resultados obtidos por meio de simulação computacional. A segunda parte da tese trata dos circuitos básicos propostos e está dividida em dois capítulos. No Capítulo 4 são apresentados os circuitos propostos para a implementação de funções de base radial e alguns resultados, obtidos via simulação SPICE. Os resultados experimentais obtidos para os circuitos implementados são apresentados no Capítulo 5. A terceira e última parte da tese tem seu enfoque no projeto de circuitos integrados em tecnologia CMOS e consiste em dois capítulos.

No Capítulo 6, é apresentada a arquitetura da RBFN que será desenvolvida na forma de circuito integrado, com base nos circuitos desenvolvidos no Capítulo 4. As RBFNs propostas foram simuladas para algumas aplicações e os resultados obtidos são apresentados no Capítulo 7. Finalmente, são apresentadas as considerações finais, com algumas propostas para trabalhos futuros, seguidas pelas Referências Bibliográficas, Apêndices e Anexos.

Contribuições da Tese

As principais contribuições desta tese são descritas a seguir:

1. Proposta e implementação de dois circuitos de função de base radial unidimensionais em tecnologia CMOS, denominados circuitos RBF_1 e RBF_2 , e proposta de suas respectivas versões em tecnologia BiCMOS;
2. Desenvolvimento dos circuitos RBF_1 e RBF_2 para o caso multidimensional e implementação do circuito RBF_1 bidimensional;
3. Projeto de uma rede de base radial bidimensional em processo CMOS utilizando os circuitos RBF_1 e RBF_2 propostos, e simulação da rede em algumas aplicações;
4. Análise do erro obtido em redes de base radial programadas com parâmetros quantizados aplicadas em aproximação de funções.

Parte I:
Redes Neurais com Função de Base Radial

Capítulo 1

Redes Neurais Artificiais

O princípio da inteligência constitui um dos mais impressionantes fenômenos da natureza. Diversas tentativas têm sido realizadas no sentido de produzir um sistema de inteligência artificial capaz de simular as capacidades demonstradas pelo cérebro. Sem dúvida, houve um progresso considerável no campo do processamento de informação, com o advento do computador digital e o aumento crescente da capacidade computacional. Entretanto, ainda que o computador digital tenha atingido um elevado grau de confiabilidade, rapidez e precisão, o fato é que o cérebro humano executa algumas operações de maneira muito superior aos mais modernos computadores. Alguns exemplos que se podem citar são: reconhecimento de padrões, reconhecimento de voz, memória associativa e muitas outras tarefas que envolvem o processamento paralelo de grande quantidade de informação e solução de ambigüidades.

Essa capacidade demonstrada pelo cérebro na solução de problemas difíceis tem levado ao estudo da maneira como a natureza trata essas questões há milhões de anos. Novos paradigmas têm sido propostos no campo da inteligência artificial, fundamentados nas propriedades apresentadas pelos sistemas biológicos. Entre os principais, podemos citar, no campo da modelagem de funções e sistemas, as Redes Neurais Artificiais (RNA) e a Lógica Nebulosa; e, no campo da otimização de sistemas, o Algoritmo Colônia de Formigas (ACO, do inglês *Ant Colony Optimization*) (DORIGO e GAMBARDELLA, 1997; HO et al., 2006), o Algoritmo Enxame de partículas (PSO, do inglês *Particle Swarm Optimization*) (EBERHART e SHI, 2001; CLERC e KENNEDY, 2002), o Algoritmo Genético (GA, do inglês *Genetic Algorithm*) e os Sistemas Imunológicos Artificiais (AIS, do inglês *Artificial Immune System*) (NUNES DE CASTRO, 2002).

As redes neurais artificiais são estruturas computacionais que apresentam como principais características: o uso de unidades simples em um paralelismo maciço, a capacidade de adaptação ou aprendizado, a capacidade de generalização e a robustez. As redes neurais artificiais surgiram a partir do estudo das redes neurais biológicas e são tentativas de emular sua capacidade de processamento paralelo. Antes de detalhar as RNAs, descreveremos sucintamente a estrutura e o princípio de funcionamento das redes neurais biológicas.

1.1 Redes Neurais Biológicas

O esquema simplificado de uma rede neural biológica é apresentado na Figura 1.1a. Basicamente, o neurônio é um processador de informação com múltiplas entradas, denominadas dendritos, um corpo celular ou soma, e uma única saída, denominada axônio, a qual pode se interligar a milhares de outros neurônios por meio de conexões denominadas sinapses. As informações trafegam pelos neurônios na forma de pulsos de tensão (*spikes*) relacionados a correntes de íons positivos e negativos. Como outras células, o neurônio contém um grande número e variedade de íons e moléculas. Na maior parte do tempo, há um excesso de cargas negativas (íons) no interior do neurônio. Essas cargas negativas se repelem mutuamente e se concentram na superfície interna da membrana celular, atraindo uma densidade equivalente de cargas positivas que se concentram na superfície externa da membrana (DAYAN e ABBOTT, 2000). A membrana celular é composta de uma camada lipídica essencialmente impermeável às moléculas carregadas e, devido a esta propriedade isolante, ela atua como um capacitor. No entanto, existem canais de íons presentes na membrana celular os quais diminuem sua resistividade em torno de 10.000 vezes em relação a uma camada lipídica pura (Figura 1.1b). A condutância resultante depende da densidade e dos tipos de canais. A quantidade de canais presentes em uma célula pode variar de centenas de milhares a milhões e muitos desses canais são seletivos, permitindo a passagem de apenas um único tipo de íon. A capacidade de condução dos canais pode variar em função de vários fatores tais como: potencial da membrana (canais dependentes de tensão), concentração de mensageiros intracelulares (Ca^{2+}) e concentração extracelular de neurotransmissores e neuromoduladores (no caso dos receptores sinápticos). A membrana

celular também contém bombas de íons seletivas que consomem energia para manter a diferença entre as concentrações iônicas interna e externa (DAYAN e ABBOTT, 2000).

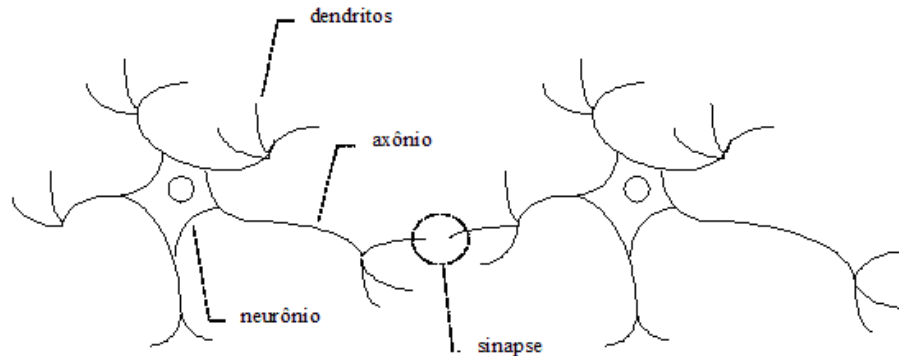


Figura 1.1a: Conexão entre dois neurônios em uma rede neural biológica.

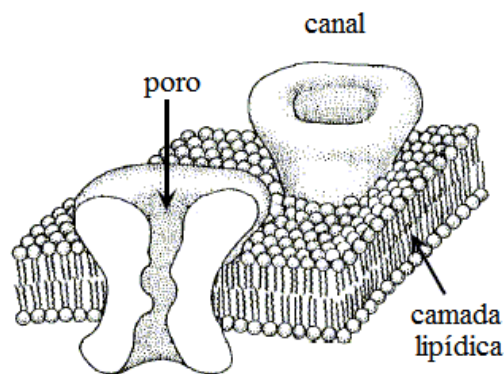


Figura 1.1b: Membrana celular. Adaptado de Dayan e Abbott (2000).

Segundo Dayan e Abbott (2000), o potencial externo ao neurônio é definido como zero por convenção. Quando ele está inativo, o potencial interno é negativo devido ao excesso de cargas negativas. Esse potencial varia se o equilíbrio iônico é quebrado pela abertura ou fechamento dos canais iônicos. Em geral, o potencial do neurônio está na faixa de -90 mV a +50 mV. Essa faixa é aproximadamente igual a $-3V_T$ e $+2V_T$, sendo $V_T = kT/q$ a tensão térmica. Para as temperaturas típicas encontradas nos seres vivos homeotérmicos, V_T varia de 24 mV a 27 mV. A Figura 1.2 apresenta um exemplo de um pulso gerado por um neurônio. Este pulso também é denominado potencial de ação.

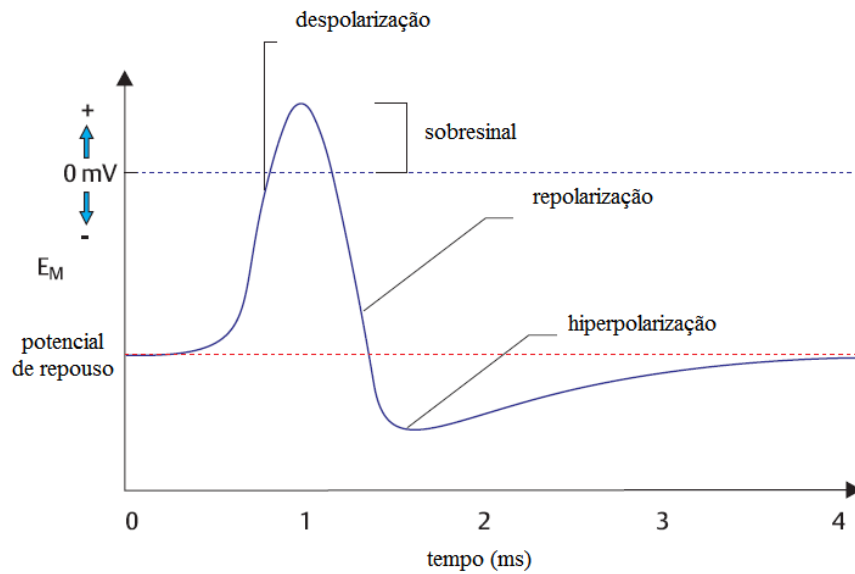


Figura 1.2: Potencial de ação (GREENSTEIN e GREENSTEIN, 2000).

1.2 Redes Neurais Artificiais

Em 1943, McCulloch e Pitts apresentaram o primeiro modelo computacional do neurônio (Figura 1.3). Esse modelo consiste em um combinador linear seguido por uma função de ativação não-linear do tipo degrau (Figura 1.4). Este comportamento pode ser descrito como “tudo ou nada”.

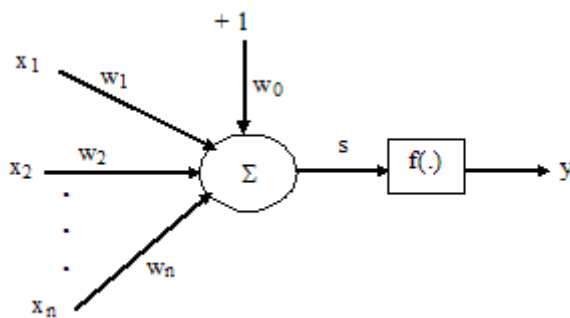


Figura 1.3: Modelo do neurônio de McCulloch-Pitts.

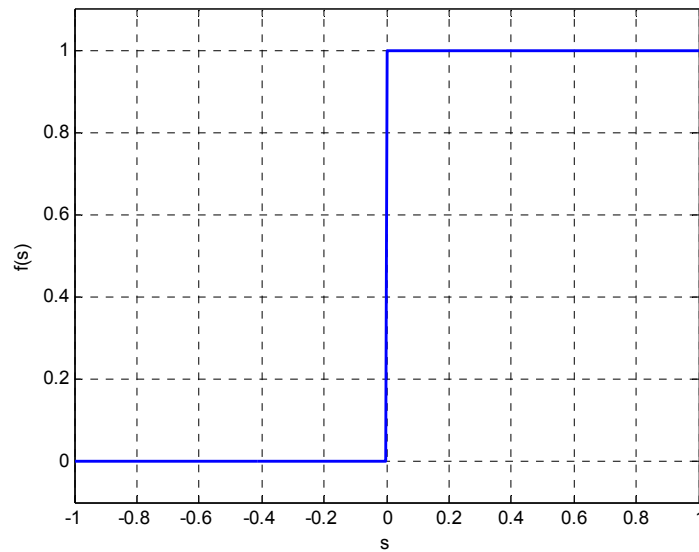


Figura 1.4: Função de ativação degrau.

Em termos matemáticos, a operação realizada pelo modelo de McCulloch-Pitts é dada por $y=f(s)$, sendo:

$$s = \sum_{i=1}^n w_i x_i + w_0 \quad (1.2.1)$$

$$f(s) = \begin{cases} +1, & \text{se } s \geq 0 \\ 0, & \text{se } s < 0 \end{cases} \quad (1.2.2)$$

Além da função degrau, várias outras funções podem ser utilizadas como função de ativação, dentre as quais destacamos:

$$\text{Função sinal: } f(s) = \begin{cases} +1, & \text{se } s \geq 0 \\ -1, & \text{se } s < 0 \end{cases} \quad (1.2.3)$$

$$\text{Função linear por partes: } f(s) = \begin{cases} +1, & \text{se } s \geq b \\ s, & \text{se } a < s < b \\ -1, & \text{se } s \leq a \end{cases} \quad (1.2.4)$$

$$\text{Função logística: } f(s) = \frac{1}{1 + \exp(-as)} \quad (1.2.5)$$

$$\text{Função tangente hiperbólica: } f(s) = \tanh(s) \quad (1.2.6)$$

$$\text{Função gaussiana: } f(s) = \exp(-as^2) \quad (1.2.7)$$

Tanto a função logística quanto a tangente hiperbólica são chamadas de funções sigmóides em virtude de sua forma em S. Baseados nesse primeiro modelo, surgiram os estudos iniciais com redes compostas por unidades de processamento simples conectadas em paralelo e em camadas sucessivas. A essa estrutura, por analogia, se denominou Rede Neural Artificial. Novas configurações de rede e novos processos de aprendizado foram propostos. Como destaques principais, podem-se citar as redes de Hopfield, Kohonen, Hebb, o Perceptron de Rosenblatt (WASSERMAN, 1989, HAYKIN, 1994) e, principalmente, a criação do algoritmo de retropropagação para redes de múltiplas camadas utilizando funções de ativação diferenciáveis como, por exemplo, funções sigmóides. Essa rede é denominada Perceptron Multicamadas (MLP, do inglês *Multilayer Perceptron*). O desenvolvimento do algoritmo de retropropagação marcou o ressurgimento das Redes Neurais Artificiais. Essas haviam sofrido um descrédito quando ficou demonstrada a impossibilidade das primeiras topologias de aproximar funções simples como a função OU-Exclusivo e também devido à falta de um algoritmo eficiente para o aprendizado. O algoritmo de retropropagação substituiu as funções degrau por funções diferenciáveis (sigmóides), o que permite a utilização do método do gradiente para se obter o aprendizado das redes de múltiplas camadas. Esse método de otimização realiza uma busca da solução ótima na direção contrária ao gradiente da função de erro quadrático na saída da rede. Além das redes supracitadas, devemos assinalar também o desenvolvimento das redes de base radial, que são o objeto de estudo desta tese. Essas redes serão mais detalhadas no capítulo seguinte.

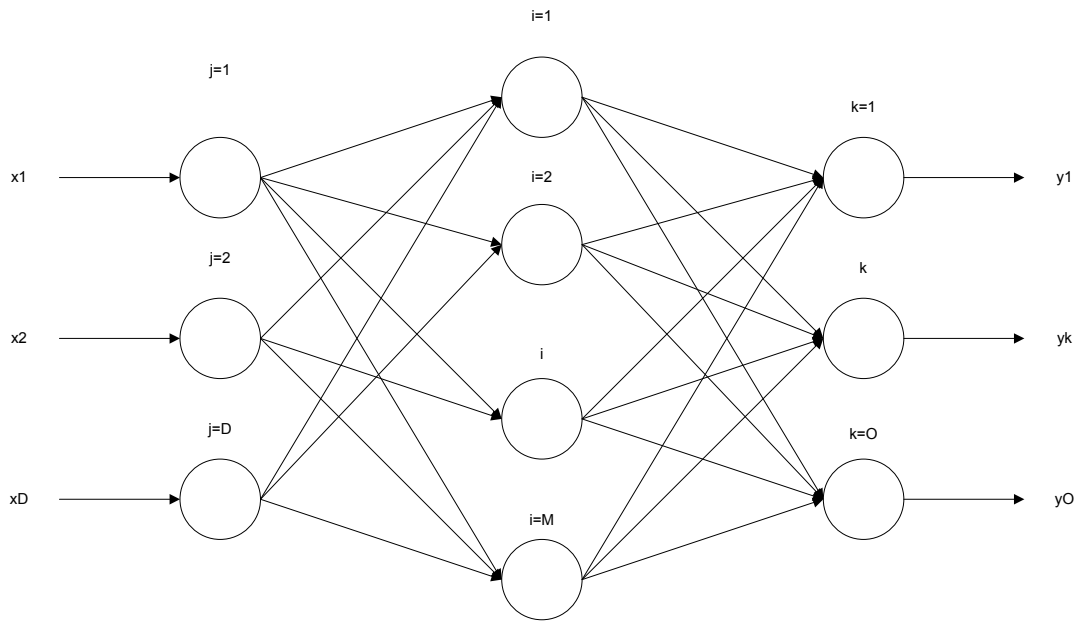


Figura 1.5: MLP com três camadas.

1.3 Tipos de Aprendizado

1.3.1 Aprendizado supervisionado

O aprendizado supervisionado distingue-se pela figura do supervisor, ou seja, pela apresentação das saídas desejadas para um conjunto de vetores de entrada. Em síntese, o funcionamento da RNA usando aprendizado supervisionado é o seguinte:

1. Apresentação de pontos da função a ser aprendida (entrada-saída desejada);
2. Fase de aprendizado (ajuste);
3. Fase de execução (em que a rede simula a função aprendida).

1.3.2 Aprendizado não-supervisionado

No aprendizado não-supervisionado não existe a figura do supervisor, ou seja, não há apresentação das saídas desejadas. Assim sendo, o funcionamento da RNA usando aprendizado não-supervisionado é o seguinte:

1. Apresentação de padrões na entrada;
2. Fase de aprendizado (ajuste);
3. Fase de execução.

1.4 Aplicações das RNAs

O problema de reconstruir uma função, a partir de informações incompletas ou contaminadas por erros ou ruído, é denominado aproximação de função. A capacidade das redes neurais artificiais de simular uma função aprendida pode ser encontrada na literatura também com outras denominações: identificação de sistemas, interpolação, síntese de funções, etc. Em outras palavras, a rede “aprende” os pares-exemplo e generaliza para valores de entrada não utilizados na fase de aprendizado, fornecendo saídas coerentes mesmo para valores de entrada que não tenham sido apresentados anteriormente. As RNAs podem ser utilizadas também para classificar um conjunto de padrões de entrada em classes determinadas. Esse problema é denominado reconhecimento ou classificação de padrões. É um problema similar ao de aproximação de funções, no entanto, a quantidade de classes na saída é limitada.

Capítulo 2

Redes de Base Radial

As redes neurais usando funções de base radial representam um paradigma relativamente novo no campo das redes neurais artificiais. Embora houvesse a aplicação desse tipo de estrutura na resolução de problemas de interpolação com múltiplas variáveis e em estatística, apenas recentemente elas foram incluídas no campo de estudos das RNAs. Broomhead e Lowe (1988) foram os primeiros a utilizar funções de base radial no projeto de redes neurais. Outras contribuições importantes foram introduzidas por Moody e Darken (1989) e Girosi e Poggio (1989). As RBFNs têm sido aplicadas à solução de diversos tipos de problemas. Este capítulo tem por finalidade apresentar os principais aspectos teóricos referentes aos fundamentos matemáticos e ao processo de aprendizado das RBFNs.

2.1 Fundamentos Teóricos

Uma função $y = f(\mathbf{x})$ pode ser aproximada por uma combinação linear de um número finito de funções básicas $\phi(\cdot)$, de acordo com a seguinte equação:

$$f(\mathbf{x}) \equiv \sum_{i=1}^M w_i \phi(\mathbf{x}, \mathbf{p}_i) + w_0 B \quad (2.1.1)$$

sendo que:

\mathbf{x} denota o padrão de entrada;

M é o número de funções da base considerada;

w_i é o peso referente à i -ésima função da base;

w_0 é o peso referente ao *bias*;

B é o *bias* (em geral, $B = +1$);

\mathbf{p}_i é o vetor de parâmetros, incluindo os fatores de forma e de deslocamento da i -ésima função de base.

Os parâmetros M , w_i , w_0 e \mathbf{p}_i são sujeitos a ajuste através de aprendizado. Usando a equação (2.1), pode-se aproximar qualquer função arbitrária com um grau de precisão desejável, desde que se disponha de um número suficientemente grande de unidades intermediárias (*hidden units*). Essa propriedade é denominada de aproximação universal. As RBFNs são aproximadores universais em subconjuntos compactos de \mathbb{R}^n . As funções de base $\phi(\mathbf{x}, \mathbf{p}_i)$ devem satisfazer alguns requisitos, tais como serem absolutamente integráveis ou serem funções monotônicas limitadas (LEE e KIL, 1991). Há muitas funções que satisfazem essas condições. As RBFs são denominadas funções radiais por apresentarem uma resposta monotonicamente crescente ou decrescente a partir de um centro. Dentre as RBFs encontradas na literatura, podem-se citar:

(1) gaussiana $\phi(x, b) = \exp(-bx^2)$;

(2) triangular;

(3) *thin-plate spline* $\phi(x, b) = x^2 \ln(bx)$;

(4) multiquadrática $\phi(x, b) = (x^2 + b^2)^{0.5}$;

(5) multiquadrática inversa $\phi(x, b) = (x^2 + b^2)^{-0.5}$;

(6) cosseno levantado (*raised cosine*);

(7) secante hiperbólica quadrática $\phi(x, b) = \text{sech}^2(bx)$;

Pode-se visualizar a construção de uma rede de base radial (RBFN) como composta basicamente por três camadas. A primeira é constituída de nós de distribuição dos sinais de entrada, a segunda é uma camada intermediária (ou oculta) com uma quantidade de unidades suficientemente alta e a última camada fornece a resposta da rede aos padrões de entrada por meio de uma combinação linear. A transformação do espaço de entrada para o espaço da camada oculta é não-linear, enquanto que a transformação da camada oculta para a saída é linear. Desta forma, a rede é projetada para executar um mapeamento não-linear do

espaço de entrada para o espaço oculto, seguido de um mapeamento linear do espaço oculto para o espaço de saída. O esquema de uma rede de base radial é mostrado na Figura 2.1.

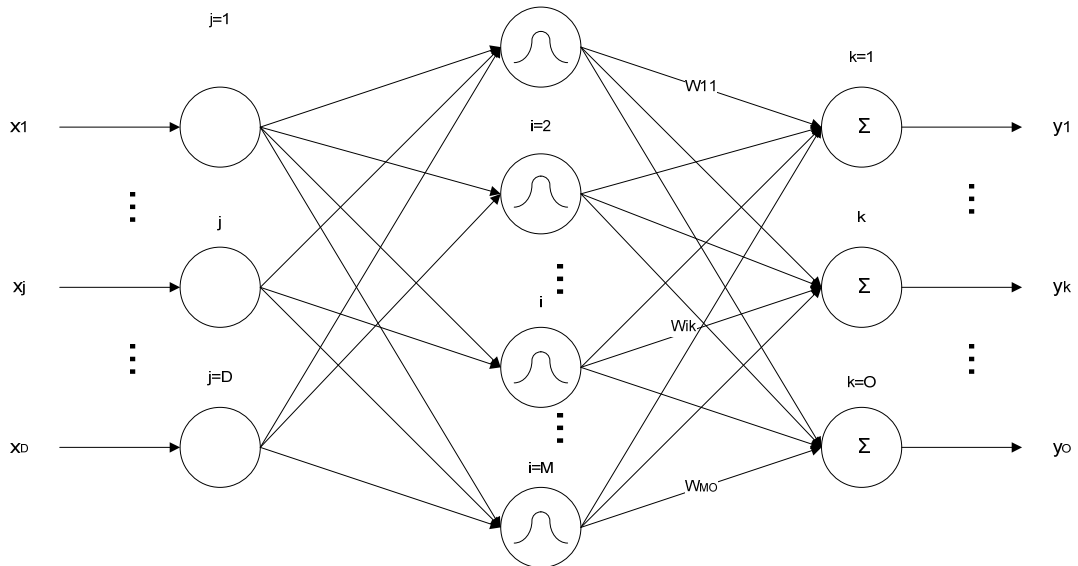


Figura 2.1: Diagrama esquemático de uma RBFN padrão.

A seguir apresentar-se-ão as duas funções de base radial que serão utilizadas nesta tese: a função gaussiana e a função secante hiperbólica quadrática.

2.2 Função Gaussiana

A função gaussiana é dada pela equação:

$$\phi(\mathbf{x}, a, b, \mathbf{c}) = a \cdot \exp(-b \|\mathbf{x} - \mathbf{c}\|^2) \quad (2.2.1)$$

sendo que os parâmetros de forma são:

a , que indica a amplitude.

b , que indica a largura,

\mathbf{c} , que denota o centro.

Na Figura 2.2 é mostrado o gráfico de uma função gaussiana unidimensional normalizada.

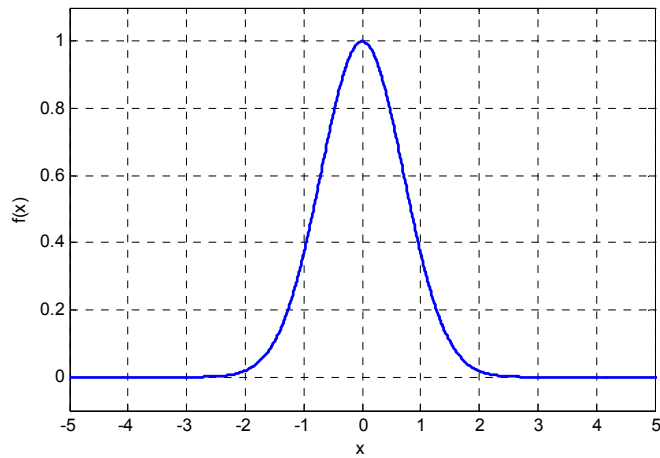


Figura 2.2: Função gaussiana com parâmetros $a = 1$, $b = 1$ e $c = 0$.

Na Figura 2.3 é mostrado o gráfico de uma função gaussiana bidimensional normalizada.

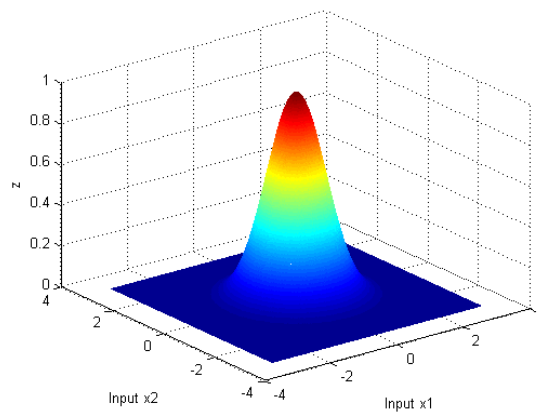


Figura 2.3: Função gaussiana bidimensional com parâmetros $a = 1$, $b = 1$ e $c = (0,0)$.

2.3 Função Secante Hiperbólica Quadrática (SHQ)

A função SHQ é dada por:

$$\phi(x, a, b, c) = a \cdot \text{sech}^2(b \| \mathbf{x} - \mathbf{c} \|) \quad (2.3.1)$$

sendo: c o centro

b a largura

a a amplitude.

Na Figura 2.4a pode ser visualizada uma função SHQ típica e na Figura 2.4b uma SHQ bidimensional.

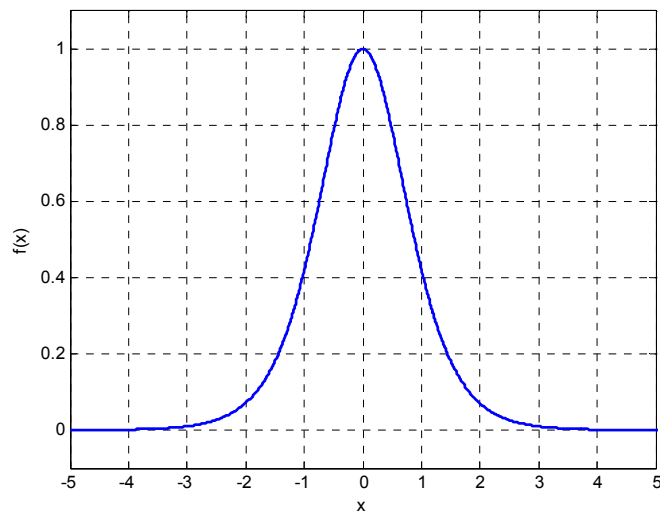


Figura 2.4a: Função SHQ com parâmetros $a = 1$, $b = 1$ e $c = 0$.

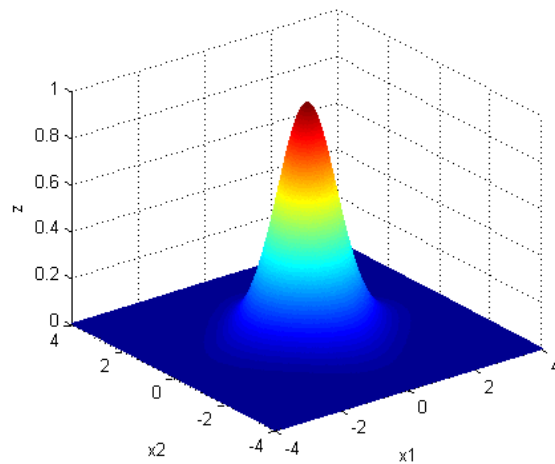


Figura 2.4b: Função SHQ bidimensional com parâmetros $a = 1$, $b = 1$ e $c = (0,0)$.

2.4 Métodos de Aprendizado

A RBFN é geralmente dividida em três camadas. A primeira é constituída por funções lineares e serve apenas para distribuir os sinais de entrada para a segunda camada. A segunda camada compõe-se das funções de base radiais e executa uma transformação não-linear no espaço de entrada. A última camada fecha o processo e constitui-se de uma simples combinação linear.

O processo de aprendizagem de uma RBFN envolve quatro passos:

1. Determinação do número de unidades da camada intermediária;
2. Localização dos centros;
3. Determinação da largura das funções;
4. Ajuste dos pesos de ligação entre as camadas intermediária e de saída.

A seguir, são apresentados três métodos para a execução do processo de aprendizagem: o método clássico, o método de Moody-Darken e o método do gradiente.

2.4.1 Método Clássico

É o mais simples. Inicialmente, arbitra-se o número M de unidades na camada intermediária. Em seguida os centros são determinados aleatoriamente a partir dos padrões de entrada. A largura é calculada por:

$$b = \frac{M}{d^2} \quad (2.4.1)$$

sendo d a máxima distância euclidiana entre os centros. Por fim, a camada de saída é otimizada com o uso da matriz pseudo-inversa.

2.4.2 Método de Moody-Darken

É um método híbrido em que a camada intermediária se auto-organiza e a camada de saída é ajustada de modo supervisionado. No ajuste dos centros utiliza-se um algoritmo de agrupamento como, por exemplo, o *k-means*. Para determinação da largura das funções utiliza-se um algoritmo que gere uma hiper-superfície contígua e suave, como o *p-nearest*. A determinação dos pesos lineares da camada de saída é feita com o algoritmo dos mínimos quadrados (LMS).

2.4.3 Método do Gradiente

Este método foi bastante difundido pelo algoritmo de retropropagação. Ele se baseia na busca em uma direção de decréscimo do erro quadrático na saída em função dos parâmetros ajustáveis da rede. Define-se a função de erro quadrático a ser minimizada no aprendizado, por:

$$E = \sum_{p=1}^N \frac{1}{2} (d_p - y_p)^2 \quad (2.4.2)$$

A alteração do parâmetro Δp no método do gradiente descendente padrão é calculada por:

$$\Delta p = -\mu \frac{\partial E}{\partial p} \quad (2.4.3)$$

em que μ é a taxa de aprendizado, p é o parâmetro a ser ajustado e E representa a função Erro Quadrático decorrente da apresentação de todos os padrões de treinamento.

Desta forma, durante o processo de aprendizado, o conjunto de parâmetros na iteração $k+1$ é dado por:

$$p(k+1)=p(k)+\Delta p(k) \quad (2.4.4)$$

O esquema básico para implementação desse método parte da configuração mostrada na Figura 2.5.

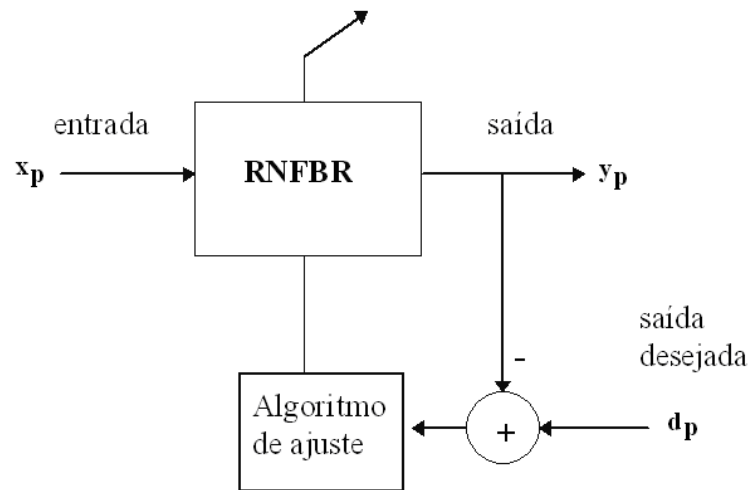


Figura 2.5: Esquema para aprendizado supervisionado da rede.

sendo: x_p o p-ésimo padrão de entrada;
 y_p a saída da rede para entrada x_p ;
 d_p a saída desejada para entrada x_p (corresponde ao valor da função a ser aprendida).

A fim de implementar o algoritmo de aprendizado (Equações. 2.4.2 e 2.4.3) para o treinamento da RBFN multidimensional com função de ativação gaussiana e uma única saída, as seguintes derivadas parciais com respeito aos parâmetros da rede foram obtidas:

$$\frac{\partial E}{\partial w_i} = \sum_{p=1}^N g_{ip} \cdot (y_p - d_p) \quad (2.4.5)$$

$$\frac{\partial E}{\partial c_{ij}} = \sum_{p=1}^N 2 \cdot b_{ij} \cdot w_i \cdot g_{ip} \cdot (y_p - d_p) \cdot (x_{jp} - c_{ij}) \quad (2.4.6)$$

$$\frac{\partial E}{\partial b_{ij}} = \sum_{p=1}^N -w_i \cdot g_{ip} \cdot (y_p - d_p) \cdot (x_{jp} - c_{ij})^2 \quad (2.4.7)$$

sendo:

$$g_{ip} = \exp\left(\sum_{j=1}^D -b_{ij} \cdot (x_{jp} - c_{ij})^2\right) \quad (2.4.8)$$

$$y_p = \sum_{i=1}^M w_i \cdot g_{ip} \quad (2.4.9)$$

- c_{ij} o j-ésimo centro da i-ésima RBF;
- b_{ij} a j-ésima largura da i-ésima RBF;
- w_i o peso da conexão entre a i-ésima RBF e a saída;
- i o índice da i-ésima RBF;
- j o índice da j-ésima dimensão da RBF;
- D a dimensão da RBFN;
- M o número de RBFs na camada intermediária;
- N o número de padrões de entrada;

Para o treinamento da RBFN Gaussiana com múltiplas entradas e múltiplas saídas (MIMO, do inglês: *Multi-Input Multi-Output*), as equações para o erro e para as derivadas parciais apresentam-se da seguinte maneira:

$$E = \sum_{k=1}^O E_k = \sum_{k=1}^O \sum_{p=1}^N \frac{1}{2} E_{kp}^2 \quad (2.4.10) \quad \text{e} \quad E_{kp} = d_{kp} - y_{kp} \quad (2.4.11)$$

$$\frac{\partial E_k}{\partial w_{ik}} = \sum_{p=1}^N g_{ip} \cdot (y_{kp} - d_{kp}) \quad (2.4.12)$$

$$\frac{\partial E_k}{\partial c_{ij}} = \sum_{p=1}^N 2 \cdot b_{ij} \cdot w_{ik} \cdot g_{ip} \cdot (y_{kp} - d_{kp}) \cdot (x_{jp} - c_{ij}) \quad (2.4.13)$$

$$\frac{\partial E_k}{\partial b_{ij}} = \sum_{p=1}^N -w_{ik} \cdot g_{ip} \cdot (y_{kp} - d_{kp}) \cdot (x_{jp} - c_{ij})^2 \quad (2.4.14)$$

sendo:

$$g_{ip} = \exp\left(\sum_{j=1}^D -b_{ij} \cdot (x_{jp} - c_{ij})^2\right) \quad (2.4.15)$$

$$y_{kp} = \sum_{i=1}^M w_{ik} \cdot g_{ip} \quad (2.4.16)$$

- c_{ij} o j-ésimo centro da i-ésima RBF;

- b_{ij} a j -ésima largura da i -ésima RBF;
 w_{ik} o peso entre a i -ésima RBF e a k -ésima saída;
 i o índice da i -ésima RBF;
 j o índice da j -ésima dimensão da RBF;
 k o índice da k -ésima saída;
 D a dimensão da RBF;
 M o número de RBFs na camada intermediária;
 N o número de padrões de entrada;
 O a dimensão da saída da RBFN;

Desta forma, as equações para o caso multidimensional MIMO são as seguintes:

$$\frac{\partial E}{\partial w_{ik}} = \sum_{k=1}^o \frac{\partial E_k}{\partial w_{ik}} \quad (2.4.17)$$

$$\frac{\partial E}{\partial c_{ij}} = \sum_{k=1}^o \frac{\partial E_k}{\partial c_{ij}} \quad (2.4.18)$$

$$\frac{\partial E}{\partial b_{ij}} = \sum_{k=1}^o \frac{\partial E_k}{\partial b_{ij}} \quad (2.4.19)$$

Com o intuito de diminuir a probabilidade de o processo de aprendizado estacionar em um mínimo local, podem-se aplicar variantes do método padrão como, por exemplo, a técnica do momento, a taxa de aprendizado adaptativa, entre outras. Nos últimos anos, tem havido muitos aprimoramentos no processo de aprendizado incluindo, por exemplo, o uso de técnicas metaheurísticas como algoritmo genético (LACERDA, CARVALHO e LUDERMIR, 2000, 2001, 2003), *simulated annealing* e busca tabu, (LUDERMIR, YAMAZAKI e ZANCHETTIN, 2006), *particle swarm optimization* (AL-KAZEMI e MOHAN, 2002) e sistemas imunológicos artificiais. A grande vantagem desse tipo de algoritmo é que eles realizam uma busca global evitando ou minimizando a probabilidade de o processo de aprendizado estacionar em um mínimo local (YAMAZAKI, LUDERMIR e SOUTO, 2002). Além disso, esses métodos não necessitam da estimativa do gradiente do

erro. Em contrapartida, são métodos iterativos que, em geral, consomem um tempo de processamento maior do que os métodos anteriores.

Conclui-se que uma série de fatores influi no processo de aprendizado como, por exemplo: a RBF utilizada, o número de componentes do vetor de entrada e o grau de liberdade permitido no ajuste dos parâmetros da rede (centros, larguras e amplitudes das RBFs). A dimensão do espaço dos vetores de entrada é importante, pois o número de unidades necessárias na camada intermediária cresce geometricamente com a dimensão do espaço de entrada. O grau de liberdade dos parâmetros influi na implementação eletrônica da rede, pois está relacionado diretamente com as limitações dos circuitos utilizados. Além disso, existe a possibilidade de se auto-ajustar o número de unidades da camada intermediária. A rigor, pode-se afirmar que o método clássico é mais rápido que o método do gradiente; porém, em geral, não é o mais otimizado.

Capítulo 3

Redes de Base Radial com Parâmetros Quantizados

No capítulo anterior, vimos que as RBFNs apresentam a propriedade de aproximação universal, a qual fornece uma garantia da existência de uma configuração de rede capaz de aproximar qualquer função com o nível de precisão desejado. Neste capítulo será investigado o efeito que a quantização dos parâmetros da rede produz em sua capacidade de aproximação de funções. Um dos objetivos deste trabalho é mostrar que, apesar de uma RBFN com precisão limitada não possuir a propriedade de aproximação universal, ela apresenta uma capacidade de aproximação relativa, que pode ser útil em determinados tipos de aplicação como, por exemplo, em aproximação de funções, reconhecimento e classificação de padrões. Além disso, do ponto de vista da implementação em hardware, o uso de parâmetros limitados e discretos torna os circuitos de armazenamento mais compactos e simples.

3.1 Redes Neurais Implementadas com Parâmetros Quantizados

Em geral, as RNAs são simuladas em computadores digitais com pouca restrição aos valores dos parâmetros. Porém, quando implementadas em hardware, em circuitos analógicos ou digitais, deve-se lidar com as limitações apresentadas pelos circuitos eletrônicos. Uma questão que surge naturalmente é: qual a capacidade de uma RNA quando os parâmetros são limitados a valores discretos? Esse problema tem sido investigado por

alguns pesquisadores para o caso MLP e RBFN. Dessa forma, é conveniente apresentar uma revisão dos trabalhos encontrados na literatura a esse respeito.

3.2 Revisão da Literatura

Xie e Jabri (1992) apresentam um modelo estatístico para analisar os efeitos da quantização na implementação digital de MLPs. Os autores concluem que o desempenho da rede sofre uma degradação quando o número de bits utilizado é diminuído. Além disso, mostram que um aumento no número de unidades da camada oculta não produz nenhum efeito na degradação e que um aumento no número de camadas causa uma piora no desempenho, se forem mantidas as outras características da rede. Dündar e Rose (1995) apresentam equações para avaliar a degradação no desempenho de MLPs em função da quantização dos pesos. Kwon e Bang (1998) apontam algumas correções nas equações apresentadas nesse artigo.

Em Wray e Green (1995) é mostrado que mesmo uma rede MLP implementada em um computador digital não apresenta a propriedade da aproximação universal. No entanto, apesar de o erro entre a função desejada e a função obtida não poder ser diminuído além de certo valor, essa rede pode ser empregada na solução de diversos problemas práticos. Khan (1996) corrobora essa tese para o caso das redes multicamadas com função de ativação sigmóide.

Voitovetsky e Guterman (1995) investigam o problema da quantização dos valores das sinapses em redes Hopfield com saídas binárias (função de ativação sinal) e contínuas (função de ativação arctan). A análise foi feita variando o ruído presente nos dados e o número de padrões a serem armazenados. O desempenho da rede foi verificado via simulação, demonstrando a viabilidade de utilização da quantização com poucos bits de precisão.

Watkins e Chau (1995) apresentam uma implementação digital de uma RBFN utilizando 10 bits de precisão e comparam os resultados obtidos em relação a uma implementação anterior utilizando um DSP de 32 bits em ponto flutuante. É mostrado que o erro é muito pequeno e o circuito economiza 88% em área e consumo de potência.

Khan (1996) propõe duas redes MLP com parâmetros limitados: redes com pesos inteiros (IWN, do inglês *Integer Weight Network*) e redes sem multiplicador (MFN, do inglês *Multiplier Free Network*). O autor prova que a rede IWN não possui a propriedade da aproximação universal, enquanto a rede MFN possui essa propriedade para o caso unidimensional. Baseando-se em simulações, o autor conjectura que isso também é válido para o caso multidimensional. Apesar das restrições nos parâmetros, ambas as redes apresentaram uma boa capacidade de aproximação, a qual foi demonstrada para vários problemas de classificação de padrões.

A capacidade de uma RBFN simulada em ponto fixo foi investigada por Sen (1997) para tarefas de classificação de padrões. Os erros entre a simulação com ponto flutuante e ponto fixo foram estudados utilizando análise estatística. Foram realizados testes de reconhecimento de três sinais de voz (consoantes P, T e K). A rede foi treinada em ponto flutuante e posteriormente, na fase de execução, foi utilizado ponto fixo. Foi verificado que a taxa de reconhecimento diminui quando utilizados sete bits ou menos. Entretanto, nenhuma análise foi realizada para problemas de aproximação de funções.

Draghici e Miller (1999) apresentam uma proposta de implementação CMOS de uma rede MLP com pesos inteiros. Os pesos são armazenados digitalmente utilizando código termométrico, sendo utilizada uma faixa de inteiros limitada. Os autores afirmam que em virtude dos pesos serem inteiros, não existe erro de quantização em seu armazenamento. As operações de multiplicação e adição são efetuadas analogicamente, em modo corrente, e a saída é obtida por um circuito comparador de correntes. São apresentados resultados de simulação SPICE para um neurônio.

Kamio et al (2000) apresentam um algoritmo de retropropagação para redes LOGO-NN (*logic oriented neural networks*). Essas redes utilizam a quantização dos pesos e

das funções de saída da rede. São mostrados alguns exemplos de simulação para tarefas de reconhecimento de padrões.

Draghici (2001) apresenta o algoritmo de decomposição baseada em restrições (CBD), o qual possui convergência garantida, podendo lidar com problemas discretos e contínuos. O mesmo autor, em 2002, apresenta um estudo sobre a capacidade das redes neurais usando pesos com precisão limitada em tarefas de classificação de padrões, o qual relaciona a dificuldade do problema de classificação com a faixa dos pesos inteiros. O autor sugere que a faixa dos pesos e a arquitetura da rede são questões independentes, já que a arquitetura está relacionada a certas propriedades do problema como, por exemplo, a separabilidade linear, enquanto a precisão dos pesos está relacionada à mínima distância entre padrões. Ambas as questões devem ser levadas em conta para garantir a solução do problema.

Plagianakos e Vrahatis (2002) apresentam um algoritmo evolucionário paralelo para o treinamento de redes MLP com pesos inteiros e função sinal. A princípio, é utilizada a função de ativação sigmóide; no entanto, no decorrer do treinamento, a função é alterada paulatinamente para a função sinal. Os pesos são restritos ao intervalo $[-3,3]$ podendo ser representados por apenas três bits. São apresentados vários resultados de simulação mostrando a eficácia do algoritmo proposto.

3.3 Redes de Base Radial com Parâmetros Quantizados

Khan (1996, p. 39) afirma que “uma rede IWN ou qualquer rede *feedforward*, com pesos limitados e discretos, não pode ser um aproximador universal.” Utilizando o mesmo raciocínio seguido pelo autor para as redes com pesos inteiros, pode-se demonstrar que a RBFN com parâmetros discretos não apresenta a propriedade da aproximação universal. De fato, dado o conjunto de funções de base:

$$Z = \left\{ w_i \exp\left(-b_i (x - c_i)^2\right) \right\} \quad (3.3.1)$$

sendo w_i , b_i e c_i pertencente a um subconjunto finito e discreto, segue que Z é um conjunto finito e não pode ser denso em $C(\mathbb{R}^d)$ para todo d (Khan, 1996). Portanto, a RBFN com parâmetros quantizados (QP-RBFN) não apresenta a propriedade da aproximação universal. Apesar disso, essa rede apresenta uma capacidade de aproximação relativa ou limitada.

Resultados experimentais obtidos via simulação mostram que essas redes são capazes de aprender funções com boa aproximação, conforme será mostrado nas próximas seções. O erro dessa aproximação depende, entre outros fatores, do grau de quantização dos parâmetros utilizado. Verifica-se que esse erro aumenta para menos bits de quantização e diminui para mais bits. No entanto, acima de certo nível de quantização, nota-se que a diminuição do erro é praticamente desprezível. Esse fato se deve ao limite que é possível obter com a rede contínua, o qual depende do algoritmo de aprendizado utilizado e da capacidade da rede.

É possível também calcular um limite superior para o erro para uma rede com zero bit de quantização, ou seja, utilizando apenas um nível para todos os parâmetros. Se considerarmos que esse nível seja zero, então o erro (E_0) obtido para essa rede nula será proporcional à soma dos valores quadráticos da função nos pontos utilizados. Se o valor dos parâmetros for diferente de zero, então o erro MSE assumirá um valor diferente de E_0 . Dessa forma, considerando que o erro quadrático médio na saída da rede é dado por:

$$E = \frac{1}{N} \sum_{p=1}^N \frac{1}{2} (d_p - y_p)^2 \quad (3.3.2)$$

se $y_p = 0$, então:

$$E = E_0 = \frac{1}{N} \sum_{p=1}^N \frac{1}{2} (d_p)^2 \quad (3.3.3)$$

3.4 Aplicações da QP-RBFN

Para uma verificação de sua funcionalidade, a RBFN com parâmetros quantizados foi aplicada a algumas tarefas de aproximação de funções. A primeira tarefa foi o aprendizado de uma função seno. Foram realizados estudos para vários níveis de quantização, sendo apresentados os resultados obtidos para 8 bits, 5 bits e 3 bits. A segunda função utilizada para avaliação da rede foi um polinômio de Hermite de sexta ordem. Também foram avaliados três níveis de quantização. Outra tarefa foi a aproximação de uma função trigonométrica de duas variáveis. Nesta tarefa foi utilizada uma quantização de três bits apenas. Os resultados apresentam os erros obtidos entre redes com parâmetros quantizados e redes com parâmetros em ponto flutuante com 64 bits de precisão.

3.4.1 Função Senoidal

A primeira tarefa foi a aproximação de uma senóide. A função foi definida no intervalo de -4π a $+4\pi$. A RBFN possui 20 bases.

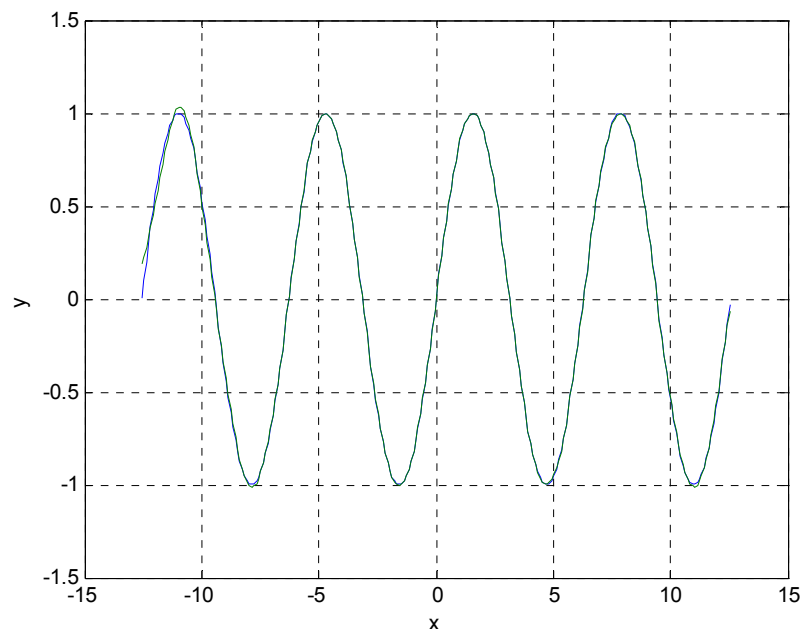


Figura 3.1: Função seno e a aproximação da rede.

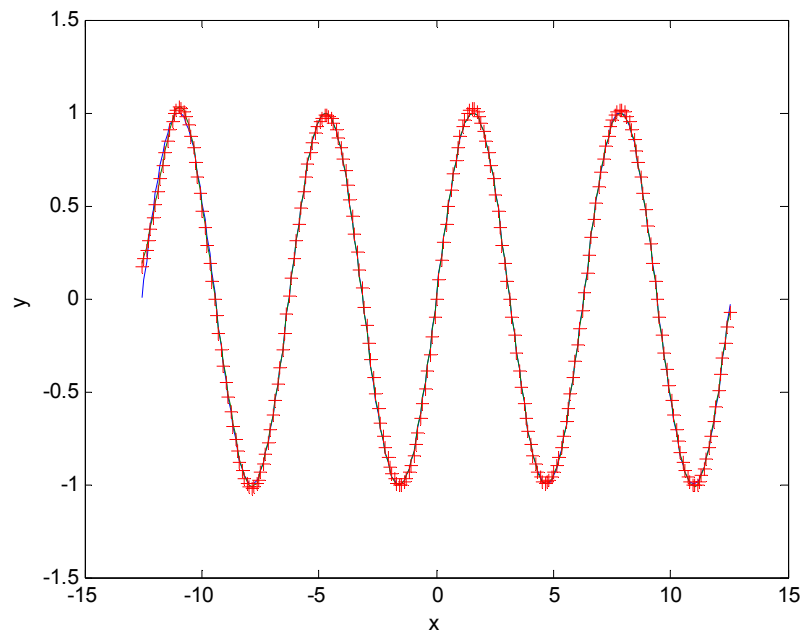


Figura 3.2: Função seno e aproximação da rede com parâmetros quantizados com 8 bits (+).

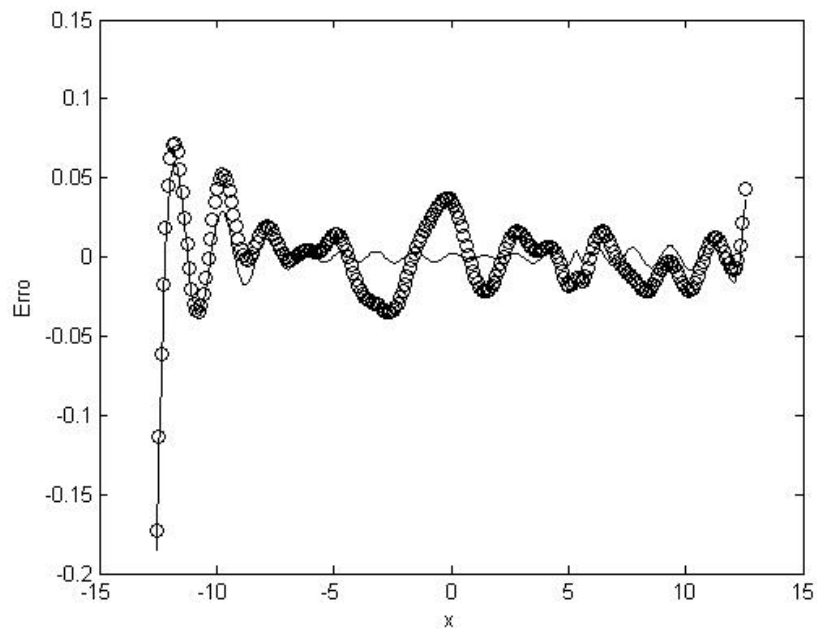


Figura 3.3: Erro entre a função seno e as saídas da rede contínua (-) e da rede com parâmetros quantizados com 8 bits (o).

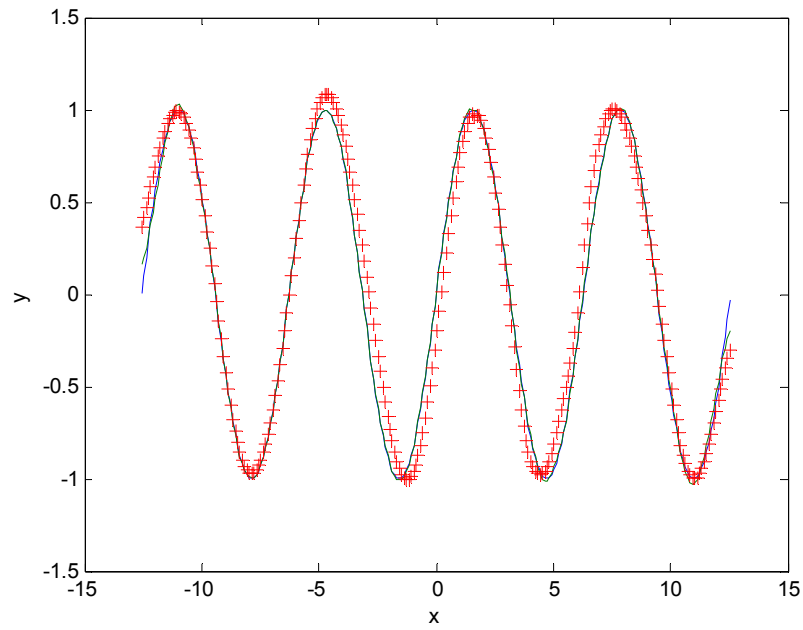


Figura 3.4: Função seno e aproximação da rede com parâmetros quantizados com 5 bits (+).

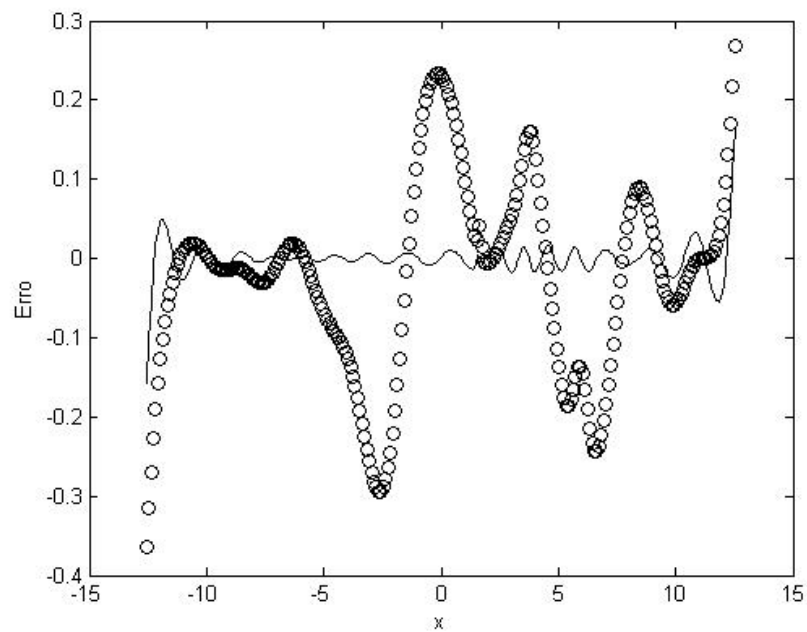


Figura 3.5: Erro entre a função seno e as saídas da rede contínua (-) e da rede com parâmetros quantizados com 5 bits (o).

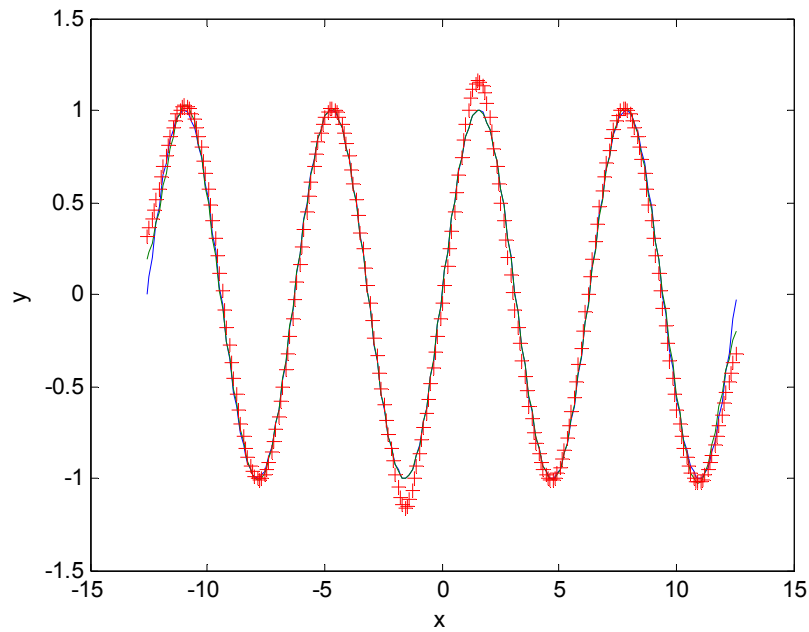


Figura 3.6: Função seno e aproximação da rede com parâmetros quantizados com 3 bits (+).

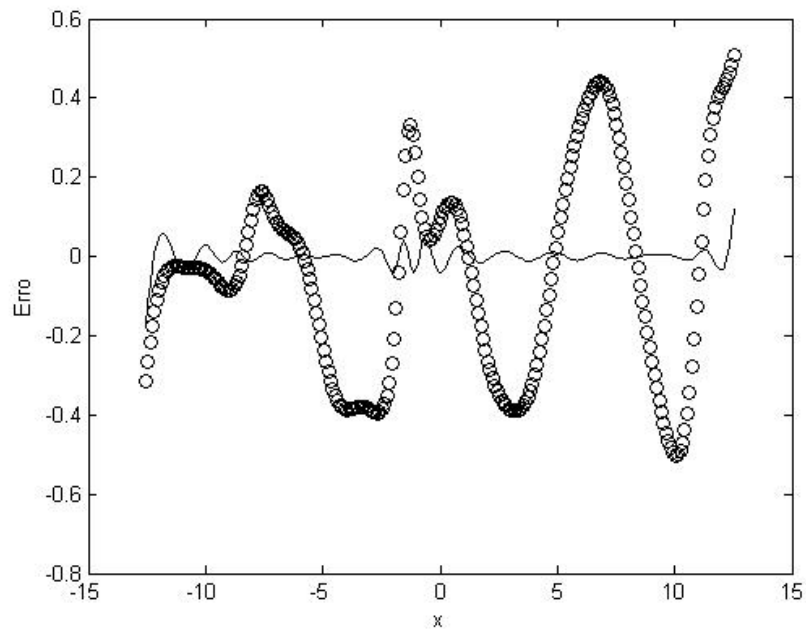


Figura 3.7: Erro entre a função seno e as saídas da rede contínua (-) e da rede com parâmetros quantizados com 3 bits (o).

A figura abaixo mostra a variação de erro quadrático médio (MSE) com relação ao número de bits de quantização dos parâmetros. Os valores de erro quadrático

médios correspondem aos menores valores obtidos após a fase de aprendizado em uma série de 50 inicializações diferentes para cada rede.

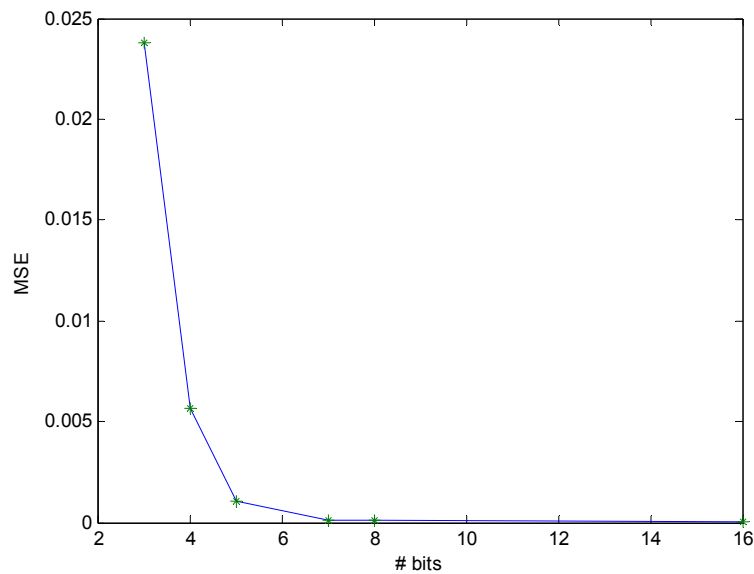


Figura 3.8: Erro quadrático médio entre a função seno e a saída da rede com parâmetros quantizados em função do número de bits de quantização.

Em escala logarítmica pode-se visualizar melhor o gráfico:

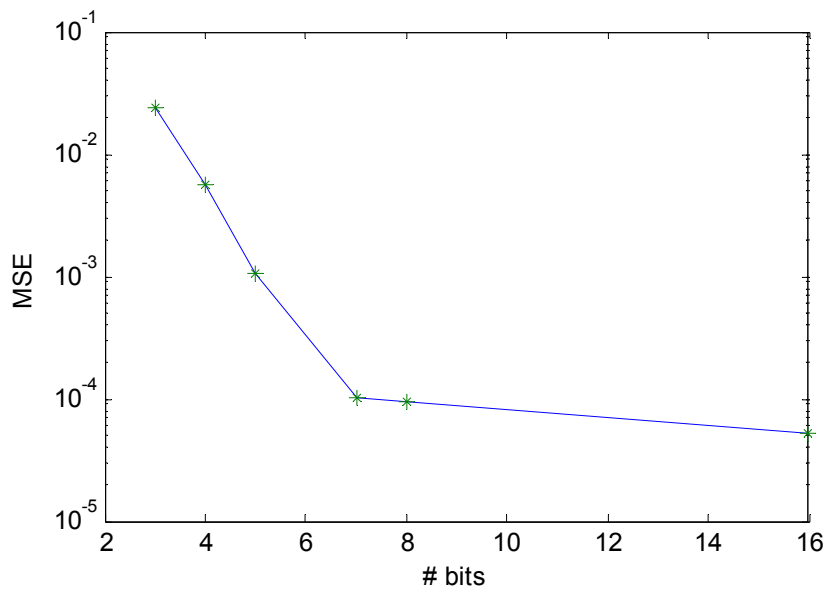


Figura 3.9: Erro quadrático médio entre a função seno e a saída da rede com parâmetros quantizados em função do número de bits de quantização (escala logarítmica).

Tabela 3.1: Erro absoluto obtido na aproximação da função seno.

		08 bits	05 bits	03 bits
RBFN	E_{\max}	0,1847	0,1641	0,1747
	E_{med}	0,0082	0,0120	0,0136
QP-RBFN	E_{\max}	0,1725	0,3647	0,5084
	E_{med}	0,0171	0,0904	0,2143

3.4.2 Função Polinomial de Hermite

A segunda tarefa foi a aproximação de um polinômio de Hermite de sexta ordem. A função foi definida no intervalo de -3 a +3 e a saída $y = H_6(x)$ foi normalizada na faixa de -1 a +1. A RBFN utilizada possui 10 bases.

$$H_6(x) = x^6 - 15x^4 + 45x^2 - 15 \quad (3.4)$$

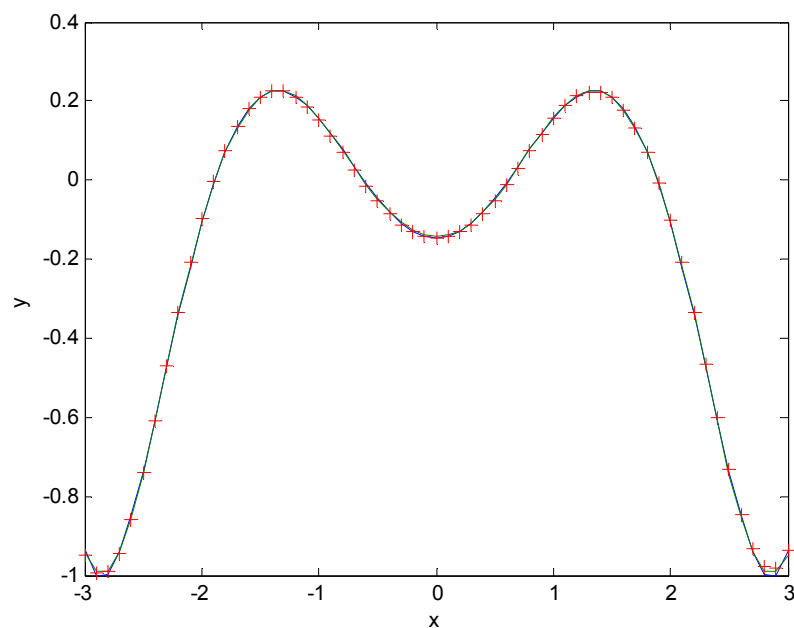


Figura 3.10: O polinômio de Hermite H_6 e a aproximação da rede contínua (-) e a da rede com parâmetros quantizados com 8 bits (+).

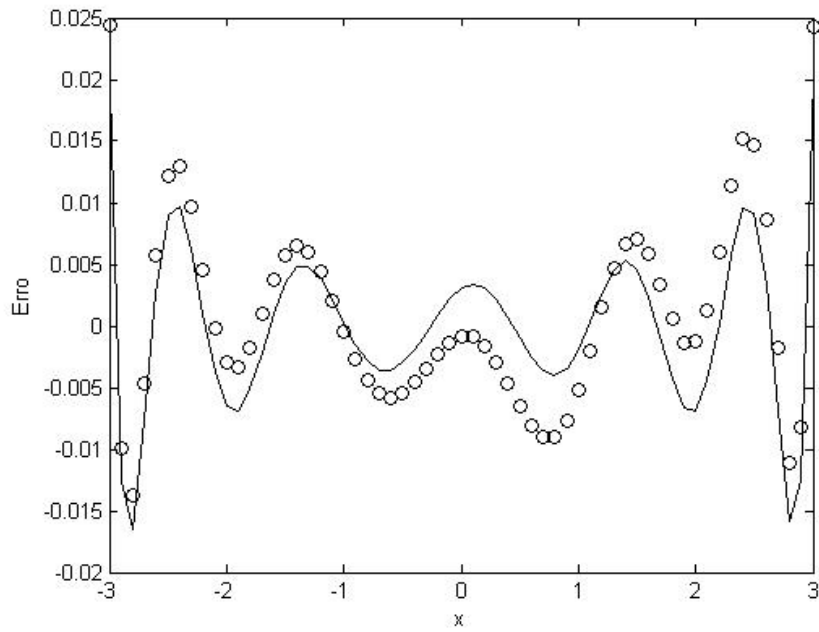


Figura 3.11: Erro entre o polinômio de Hermite H_6 e as aproximações da rede contínua (-) e da rede com parâmetros quantizados com 8 bits (o).

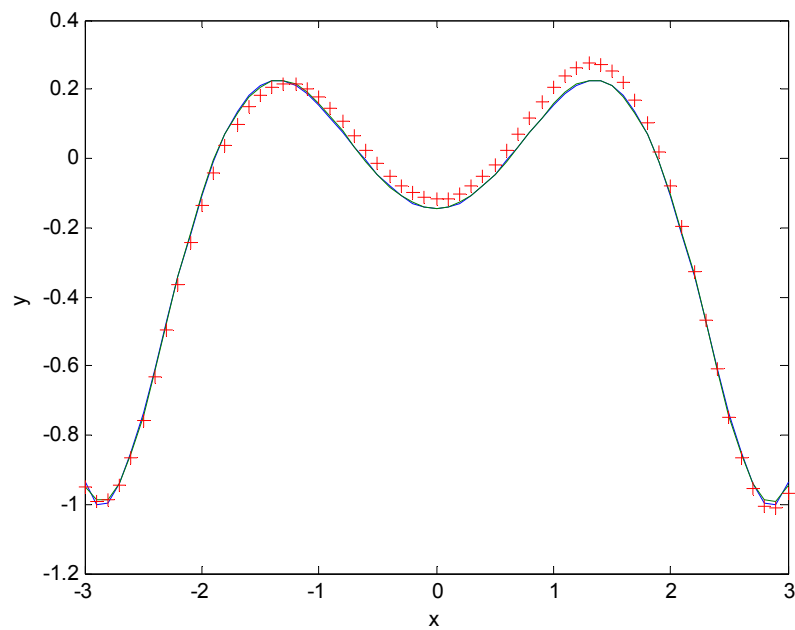


Figura 3.12: O polinômio de Hermite H_6 e a aproximação da rede contínua (-) e a da rede com parâmetros quantizados com 5 bits (+).

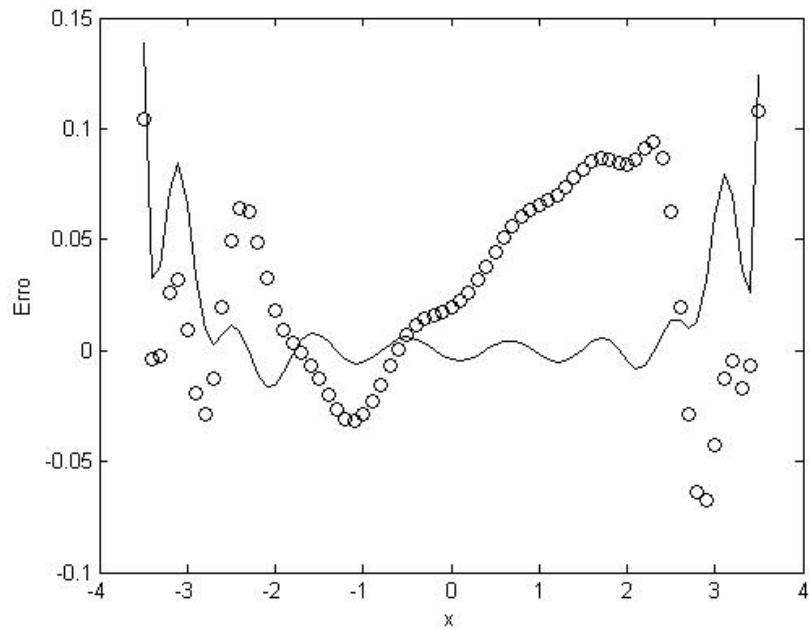


Figura 3.13: Erro entre o polinômio de Hermite H_6 e as aproximações da rede contínua (-) e da rede com parâmetros quantizados com 5 bits (o).

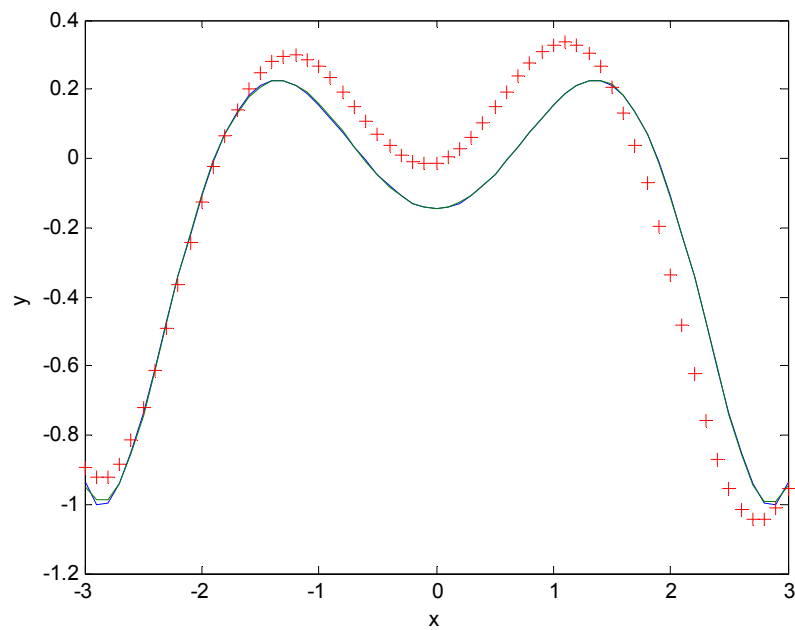


Figura 3.14: O polinômio de Hermite H_6 e a aproximação da rede contínua e a da rede com parâmetros quantizados com 3 bits.

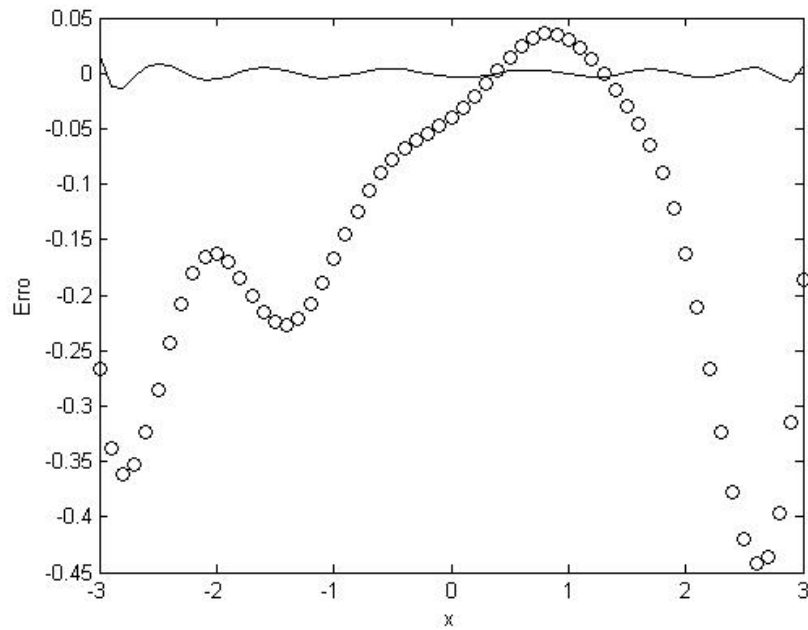


Figura 3.15: Erro entre o polinômio de Hermite H_6 e as aproximações da rede contínua (-) e da rede com parâmetros quantizados com 3 bits (o).

Tabela 3.2: Erro absoluto obtido na aproximação da função polinomial de Hermite.

		08 bits	05 bits	03 bits
RBFN	E_{\max}	0,0117	0,1389	0,0177
	E_{med}	0,0033	0,0169	0,0036
QP-RBFN	E_{\max}	0,0195	0,1077	0,4418
	E_{med}	0,0048	0,0406	0,1620

3.4.3 Função Trigonométrica Bidimensional

A função $z(x_1, x_2) = \text{sen}(x_1) \cdot \text{sen}(x_2)$ foi definida no intervalo de $-\pi$ a $+\pi$. A saída $z(x_1, x_2)$ é definida na faixa de -1 a +1. A RBFN utilizada possui seis bases.

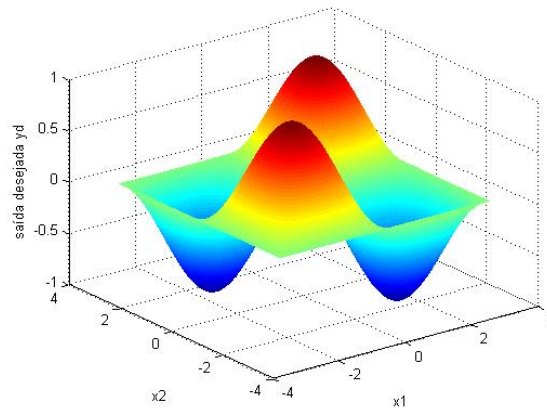


Figura 3.16: Saída desejada.

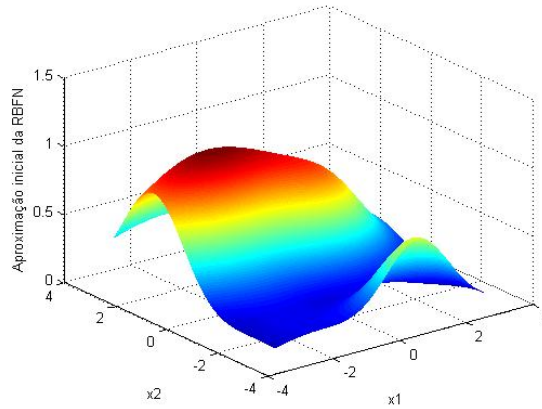


Figura 3.17: Saída inicial da RBFN.

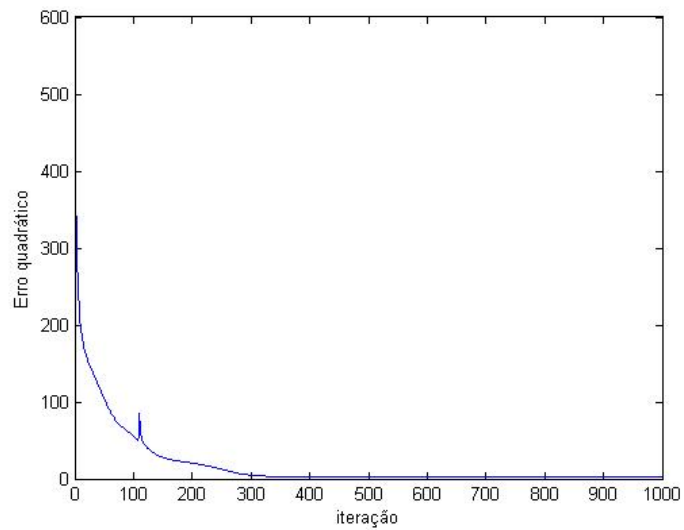


Figura 3.18: Variação do erro quadrático da RBFN durante a fase de aprendizado.

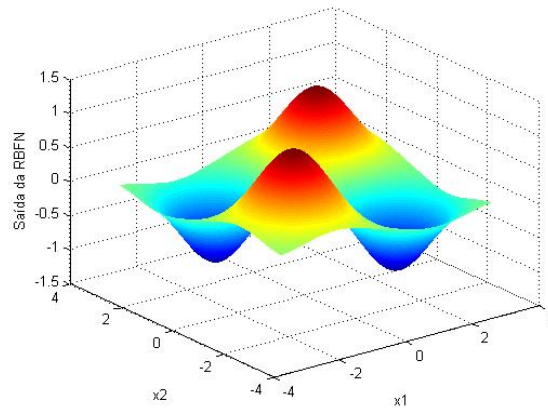


Figura 3.19: Aproximação da RBFN após o aprendizado.

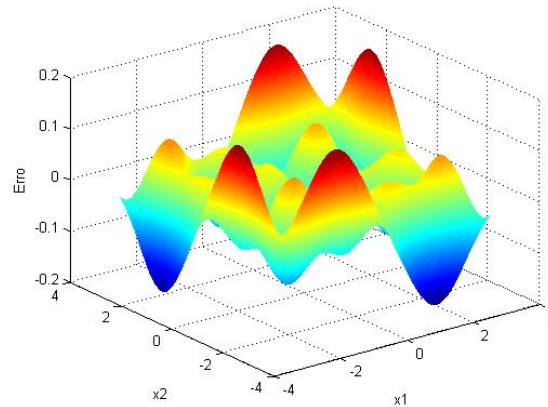


Figura 3.20: Erro entre a função desejada e a aproximação da RBFN.

Os parâmetros da rede foram quantizados com 3 bits entre os valores contínuos mínimo e máximo. Os parâmetros da rede quantizada após a fase de aprendizado e o processo de quantização são mostrados na Tabela 3.1.

Tabela 3.3: Parâmetros da rede quantizada obtidos após o treinamento.

Parâmetro	RBF					
	1	2	3	4	5	6
w	1.0839	-1.0725	1.0839	-0.4564	-0.1484	-0.7645
b _{i1}	0.6619	0.6619	0.6619	0.6619	0.1510	0.7470
b _{i2}	0.6748	0.6748	0.6748	0.7377	0.2977	0.6748
c _{i1}	-1.5977	1.5286	1.5286	-1.5977	0.1888	-1.5977
c _{i2}	-1.5248	-1.5248	1.2899	1.7590	0.3517	1.2899

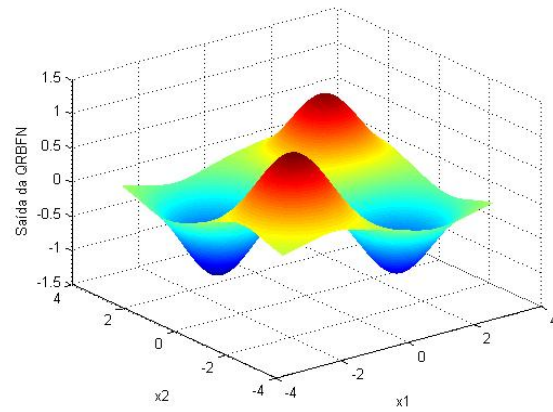


Figura 3.21: Aproximação da QP-RBFN.

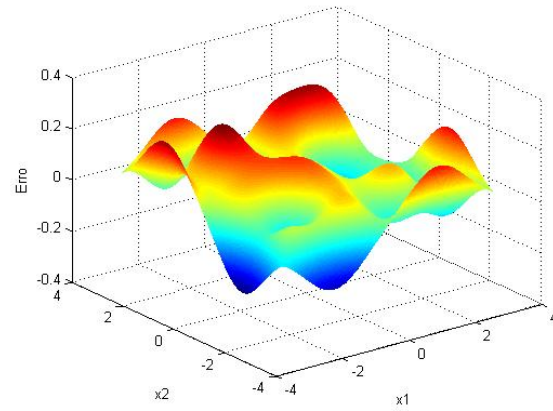


Figura 3.22: Erro entre a função desejada e a aproximação da QP-RBFN.

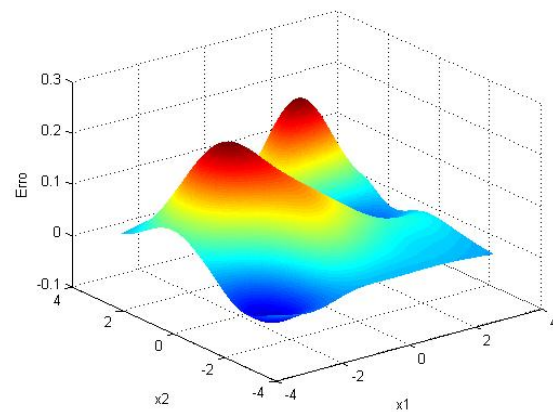


Figura 3.23: Diferença entre as aproximações RBFN e QP-RBFN.

Tabela 3.4: Erro absoluto obtido na aproximação da função trigonométrica bidimensional.

		03 bits
RBFN	E_{\max}	0,1787
	E_{med}	0,0367
QP-RBFN	E_{\max}	0,2555
	E_{med}	0,0752

3.5 Conclusões

Neste capítulo, foram apresentados alguns resultados obtidos para tarefas de aproximação de funções utilizando parâmetros quantizados. Mostrou-se que mesmo com uma quantização severa a rede pode obter uma boa aproximação. O erro dessa aproximação depende, dentre outros fatores, do grau de quantização utilizado. Verificou-se que esse erro aumenta quando se utilizam menos níveis no armazenamento dos parâmetros e diminui para mais níveis. No entanto, a partir de certo nível de quantização, nota-se que a diminuição do erro é praticamente desprezível. Esse fato se deve ao limite que é possível obter com a rede contínua, o qual depende da capacidade da rede e do algoritmo de aprendizado utilizado.

Parte II:
Implementação Eletrônica de Redes de Base Radial

Capítulo 4

Propostas de Implementação Eletrônica de Redes de Base Radial

Neste capítulo serão apresentados os circuitos propostos para a implementação das funções de base radial, bem como uma visão geral de algumas propostas encontradas na literatura. Serão comentadas também as principais características das implementações práticas das redes e, ainda, os desafios a serem enfrentados.

4.1 Tipos de Implementação

As RNAs podem ser classificadas com relação a diversos fatores, como por exemplo: arquitetura, programabilidade, tipo de aprendizado, número de neurônios, precisão, custo, etc. Inicialmente, podem-se dividir as implementações eletrônicas em digital, analógica ou mista.

4.1.1 Implementação Analógica

Em uma implementação analógica, os sinais de informação apresentam uma amplitude contínua. A grande vantagem da implementação analógica é o aproveitamento do paralelismo natural da rede, que pode aumentar a velocidade de trabalho do circuito. Entretanto, há vários problemas a enfrentar, sendo que o principal deles é a necessidade de uma memória analógica para armazenar os parâmetros ajustáveis da rede. Há várias propostas encontradas na literatura para implementação dessas memórias. As mais usuais utilizam

resistores, capacitores ou transistores MOS de porta flutuante (FGMOS). Outro problema importante é o descasamento entre componentes, que gera diferenças indesejadas de tensão ou corrente (“offset”, em inglês). Esta desvantagem é compensada pela eficiência computacional baseada em princípios de teoria de circuitos e nas leis físicas (HAYKIN, 1994).

As operações analógicas se baseiam em funções de tempo, espaço, tensão, corrente ou carga, que estão diretamente relacionadas à física dos componentes. Essas funções são descritas nos níveis de dispositivo, circuito e arquitetura. O nível de dispositivo consiste na combinação das equações características dos transistores MOS em inversão fraca e inversão forte, as quais definem os tipos de operações analógicas que podem ser executadas. Em geral, os transistores MOS operam segundo uma função quadrática (inversão forte), função exponencial (inversão fraca), como chaves e como resistência controlada por tensão. A região de inversão fraca apresenta alta transcondutância e dissipação de potência extremamente baixa, da ordem de pW a μ W, enquanto a região de inversão forte permite a operação em velocidades maiores, com um aumento correspondente no consumo de potência. O nível de circuito é governado pela lei da conservação de carga e pela lei da conservação da energia as quais geram, respectivamente, as leis de Kirchoff das correntes e das tensões. No nível de arquitetura, equações matemáticas algébricas, diferenciais e integrais são utilizadas para implementar funções úteis, dependendo da aplicação de interesse.

4.1.2 Implementação Digital

Em uma implementação digital, os sinais são quantizados em um conjunto finito de amplitudes discretas. A abordagem digital apresenta várias vantagens sobre a analógica. A primeira é a facilidade de projeto e fabricação, incluindo a possibilidade de rápida prototipagem com a utilização de FPGA ou EPLD. Além disso, processos digitais são mais disponíveis que os analógicos. Outra vantagem é a maior flexibilidade, permitindo o uso de algoritmos mais complexos e expandindo a faixa de aplicações possível. A tecnologia digital oferece ainda as vantagens de alta precisão e facilidade de armazenamento dos parâmetros das redes. Entretanto, uma desvantagem importante é que a realização da operação multiplicação consome muita área e potência (HAYKIN, 1994).

4.1.3 Implementação Mista

O uso de computação analógica é atrativo para redes neurais pela compacidade, velocidade e ausência de quantização. O uso de técnicas digitais, por outro lado, é preferido para comunicação a longa distância porque os sinais digitais são mais robustos e facilmente transmitidos e armazenados. Esses fatos encorajam o uso de abordagens híbridas, que aproveitam as vantagens das duas técnicas. No caso da rede mista, em geral, a implementação das funções é feita analogicamente e o armazenamento dos parâmetros e o controle da rede são feitos digitalmente.

4.2 Tipos de Aprendizado

Os tipos de aprendizado na implementação em hardware podem ser divididos em:

- *off-line*: é feita uma simulação no computador e no final os parâmetros encontrados são transferidos para o circuito;
- *on-chip*: o algoritmo de aprendizado é implementado eletronicamente dentro do circuito;
- *chip-in-the-loop*: em cada iteração o erro na saída é calculado apresentando os padrões exemplos ao circuito e medindo-se a sua saída. Os valores atualizados dos parâmetros são calculados no computador e transferidos para o circuito.

4.3 Revisão Bibliográfica

Neste item, será feita uma retrospectiva de algumas propostas encontradas na literatura sobre implementação de RBFNs em *hardware*.

Delbrück (1991, 1993) apresenta um circuito CMOS para computar similaridade e dissimilaridade entre sinais. O circuito utiliza um bloco denominado correlacionador de corrente. Nessa proposta podem ser ajustados os centros e as amplitudes da função. Uma das características do circuito é que ele usa transistores em inversão fraca. Isto implica níveis de corrente da ordem de nanoamperes tornando o circuito evidentemente mais sensível a ruídos. Além disso, o uso da inversão fraca limita a velocidade de resposta do circuito.

Watkins e Chau (1992) apresentam um neurocomputador híbrido, no qual algumas funções são implementadas analogicamente para aproveitar o paralelismo e o tamanho reduzido dos circuitos analógicos, e outra parte é implementada em um DSP Motorola 56000. São apresentados dois circuitos para implementação de funções gaussianas. O primeiro utiliza um multiplicador de Gilbert para realizar a função quadrática e um transistor MOS em inversão fraca conectado como diodo para a função exponencial. Este circuito permite a variação do centro e da largura com várias ordens de magnitude, tendo sido escolhido para a implementação do neurocomputador. O outro circuito apresentado é uma variação do circuito de Delbrück (1991, 1993) e permite a variação de todos os parâmetros incluindo a amplitude, mas a largura fica restrita a apenas uma ordem de magnitude. Não são apresentados resultados de aplicações práticas.

Churcher (1993) propõe um circuito para redes gaussianas utilizando quadradores na entrada, os quais implementam a norma euclidiana (distância entre o vetor de entrada e o centro da gaussiana). A função exponencial é realizada por meio de um integrador com capacitor e a saída da gaussiana é em forma de pulsos retangulares sendo a amplitude da função dada pela largura dos pulsos. A variância (largura da função) pode variar duas ordens de magnitude.

Em Turchetti (1993) é proposta a implementação de funções não-lineares utilizando séries de Dirac as quais, nesse caso, são compostas por diferenças de tangentes hiperbólicas. O artigo não trata esse processo como RBFN, mas o resultado é similar. Também não é feita uma análise do caso multivariável.

Em Choi (1994) é proposto um circuito para implementar a função gaussiana usando aproximação por partes. Isso é feito por meio de um par diferencial e espelhos de corrente, os quais realizam as operações aritméticas necessárias (soma ou diferença). Os parâmetros de amplitude e centro podem ser ajustados eletronicamente, mas a largura das funções depende das dimensões dos transistores do par diferencial, o que torna a programação mais difícil e limitada. O artigo não trata da escalabilidade.

Marshall e Collins (1995) apresentam um circuito VLSI que implementa uma RBFN baseado em um circuito compacto para a distância euclidiana utilizando transistores *floating-gate*. A saída do circuito euclidiano é aplicada a um circuito *bump* proposto. Por fim, para a obtenção da saída são utilizados multiplicadores baseados em um multiplicador de Gilbert modificado. Um protótipo com três entradas, três bases e duas saídas foi fabricado e foram apresentados resultados práticos. Também são discutidos o impacto das variações nos dispositivos e o escalonamento para um sistema RBFN completo.

Em Madrenas et al. (1996) é proposto um circuito para implementação de função gaussiana. Ele utiliza um transistor MOS na saturação atuando como quadrador e um espelho de Widlar com MOS em inversão fraca para realizar a exponencial. O circuito implementa apenas metade da função. Para conseguir uma função completa é necessário utilizar um circuito quadrático na entrada. A amplitude pode ser variada em uma ampla faixa, mas a largura é limitada a uma pequena faixa. Para tornar o circuito realmente viável é imprescindível o uso de quadradores na entrada.

Theogarajan e Kers (1997) apresentam um circuito que implementa uma função de base radial aproximadamente gaussiana. O funcionamento do circuito é baseado na característica de corrente de um par inversor CMOS. Apesar da simplicidade e tamanho reduzido, o circuito apresenta certos inconvenientes, como assimetria da função e o fato da amplitude depender das dimensões dos transistores que formam o par complementar. É descrita, também, uma versão multivariável do circuito utilizando o correlacionador de correntes para multiplicar as funções entre si.

Em Lin (1998) é apresentado um circuito capaz de aproximar funções gaussianas, quando opera em inversão fraca, e funções quadráticas, em inversão forte. O circuito permite a programação de amplitude, centro e largura das funções. Os autores não apresentam nenhuma extensão para operação multidimensional, e também nenhuma aplicação em uma rede completa.

Çevikbaş et al. (2000) apresentam o projeto de uma Rede de Base Radial Gaussiana (GRBF), em tecnologia BiCMOS, a qual utiliza multiplicadores de Gilbert, circuitos somador e exponencial. Os parâmetros da rede são armazenados digitalmente com a utilização de conversores D/A. São apresentados resultados de simulação para aproximação de funções de uma variável mostrando a funcionalidade da arquitetura proposta.

Ogawa e Shibata (2001) apresentam uma memória associativa analógica baseada em circuitos gaussianos utilizando quatro transistores MOS em conexão cruzada, sendo dois pares *floating-gate*. A arquitetura proposta utiliza conversores D/A e técnicas de capacitor chaveado. O circuito foi implementado e algumas medidas experimentais são apresentadas.

Peng, Minch e Hasler (2005) propõem um circuito *bump* programável utilizando transistores MOS de porta flutuante (*floating-gate*). O circuito permite a programação de centro, largura e amplitude das funções. Os mesmos autores, em 2007, apresentam uma versão aprimorada do circuito que mostra melhorias em relação à programação e uma melhor aproximação da função gaussiana. É mostrada a possibilidade de estender o circuito para o caso multidimensional, com resultados experimentais apresentados para o caso bidimensional. Os autores apresentam a possibilidade de aplicação do circuito em um quantizador vetorial.

Abuelma'ati e Shwehneh (2006) apresentam um circuito que pode ser configurado para implementar funções sigmóide, triangular e gaussiana. O circuito triangular é baseado em chaveamento de correntes via espelhos de corrente. Os circuitos para gaussiana e sigmóide utilizam o circuito triangular e fontes de corrente *peaking*, as quais acrescentam ou subtraem correntes não-lineares do circuito triangular. São apresentados resultados de

simulação das funções realizadas, mostrando um erro pequeno para a função gaussiana e um erro um pouco maior para a função sigmóide. O resultado final é um circuito reconfigurável, porém não otimizado para nenhuma função específica.

Lucks (1998) apresenta um circuito que implementa as funções gaussiana e secante hiperbólica quadrática baseado na curva de transcondutância de pares diferenciais bipolares. A curva de transcondutância é obtida utilizando um método de aproximação de derivadas baseado em diferenças finitas. O circuito pode ser programado em todos os parâmetros de forma. Uma rede de base radial unidimensional foi implementada e aplicada a vários problemas de aproximação de funções, linearização de transdutores e de osciladores controlados por tensão com sucesso. Foi utilizado aprendizado *off-line* e o método do gradiente com *simulated annealing* para a fase de aprendizado. Esse circuito foi posteriormente integrado em tecnologia BiCMOS conforme Oliveira (2001) e aplicado por Nagashima (2006) na resolução de alguns problemas unidimensionais, utilizando algoritmos genéticos para a implementação do aprendizado *chip-in-the-loop*. Os circuitos propostos nesta tese dão continuidade a esse desenvolvimento, enfocando agora o caso multidimensional, importante para a implementação de redes de maior capacidade. Além disso, procurou-se utilizar a tecnologia CMOS, que é economicamente mais interessante do que a tecnologia BiCMOS.

4.4 Circuitos Propostos

Nesta seção serão apresentadas novas propostas para a implementação eletrônica de funções de base radial, tais como funções gaussianas e similares (também chamadas de funções sino), as quais podem ser utilizadas na implementação de redes de base radial. Os circuitos têm seu funcionamento com base nas curvas características de transcondutância de pares diferenciais MOS e bipolar, obtidas com a técnica da aproximação da derivada, ou seja, estas RBFs serão obtidas pela subtração de duas funções sigmóides dadas pelas correntes de saída de dois pares diferenciais. O circuito RBF₁ utiliza a topologia cascode dobrado para efetuar a operação de subtração, enquanto o circuito RBF₂ utiliza apenas espelhos de corrente para efetuar essa operação. Também nesta seção são propostas

versões multidimensionais baseadas nas versões unidimensionais dos circuitos. O par diferencial é um dos blocos mais versáteis e mais utilizados na implementação de redes neurais e constitui-se no núcleo das células propostas. Portanto, é conveniente fazer uma breve revisão sobre esse circuito e suas curvas características principais.

4.4.1 Par diferencial bipolar

O par diferencial bipolar é apresentado na Figura 4.1. Como o próprio nome sugere, a saída do circuito é função da diferença entre as tensões nas entradas.

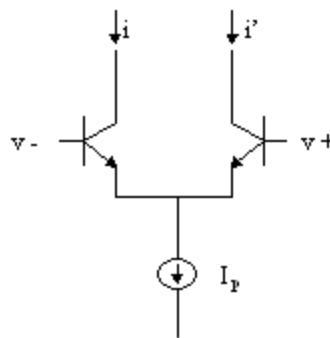


Figura 4.1: Esquema simplificado do par diferencial bipolar.

A equação que rege o par diferencial bipolar é dada por:

$$I_{dif} = \alpha_F I_p \tanh\left(\frac{V_{dif}}{2V_t}\right) \quad (4.1)$$

sendo :

- V_{dif} a diferença de tensão nas entradas;
- I_{dif} a diferença entre as correntes de coletor;
- I_p a corrente de polarização;
- α_F a relação entre as correntes de coletor e emissor em um transistor bipolar;
- V_t a tensão térmica (V).

A Figura 4.2 mostra a curva característica de saída do par diferencial bipolar.

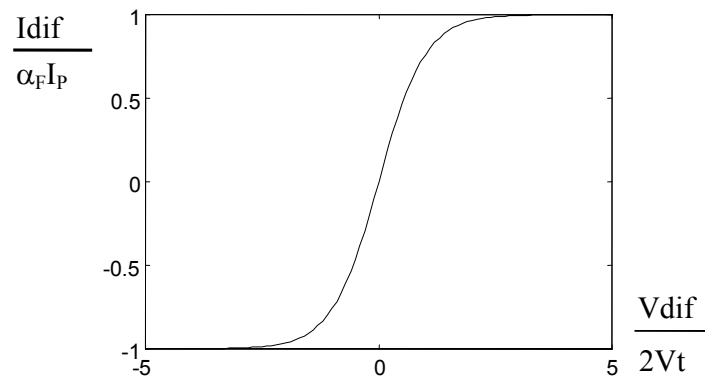


Figura 4.2: Curva característica do par diferencial bipolar.

A equação de transcondutância do par diferencial bipolar é dada por:

$$g_m = \frac{\partial I_{dif}}{\partial V_{dif}} = \frac{\alpha_F I_P}{2V_t} \operatorname{sech}^2\left(\frac{V_{dif}}{2V_t}\right) \quad (4.2)$$

sendo : g_m a transcondutância do par diferencial.

A Figura 4.3 apresenta a curva característica de transcondutância típica de um par diferencial bipolar.

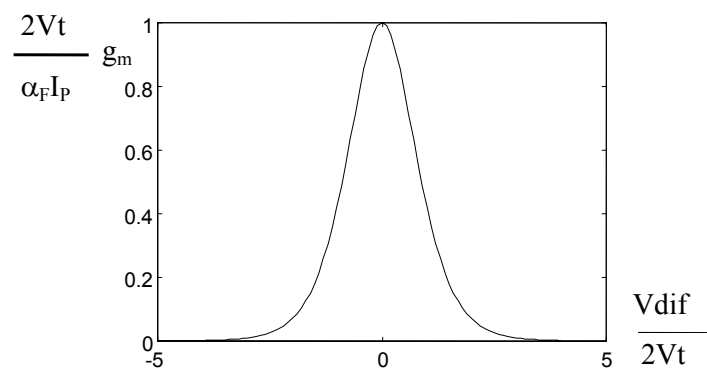


Figura 4.3: Curva de transcondutância do par diferencial bipolar.

A corrente diferencial de saída do par diferencial bipolar é uma função tangente hiperbólica e a curva de transcondutância é uma função secante hiperbólica ao quadrado, derivada da primeira.

4.4.2 Par diferencial MOS

O par diferencial NMOS é apresentado na Figura 4.4. Seu funcionamento é similar ao do par diferencial bipolar. A corrente de saída do circuito é função da tensão diferencial entre as entradas.

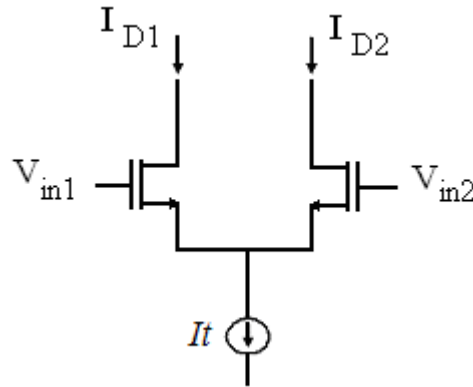


Figura 4.4: Esquema simplificado do par diferencial NMOS.

A equação que rege o par diferencial NMOS é dada por:

$$I_{D1} - I_{D2} = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{in1} - V_{in2}) \sqrt{\frac{4It}{\mu_n C_{ox} \frac{W}{L}} - (V_{in1} - V_{in2})^2} \quad (4.3)$$

$$I_{dif} = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{dif}) \sqrt{\frac{4It}{\mu_n C_{ox} \frac{W}{L}} - (V_{dif})^2} \quad \text{se } |V_{dif}| \leq \sqrt{\frac{2It}{\mu_n C_{ox} \frac{W}{L}}}$$

$$I_{dif} = -It \quad \text{se } V_{dif} \leq -\sqrt{\frac{2It}{\mu_n C_{ox} \frac{W}{L}}}$$

$$I_{dif} = It \quad \text{se } V_{dif} \geq \sqrt{\frac{2It}{\mu_n C_{ox} \frac{W}{L}}}$$

sendo : V_{dif} a diferença de tensão nas entradas;
 I_{dif} a diferença entre as correntes de dreno;

- I_t a corrente de polarização;
 W a largura do canal do transistor MOS;
 L o comprimento do canal do transistor MOS;
 μ_n a mobilidade de elétrons;
 C_{ox} a capacitância do óxido;

A Figura 4.5 mostra uma curva característica de saída para um par diferencial NMOS.

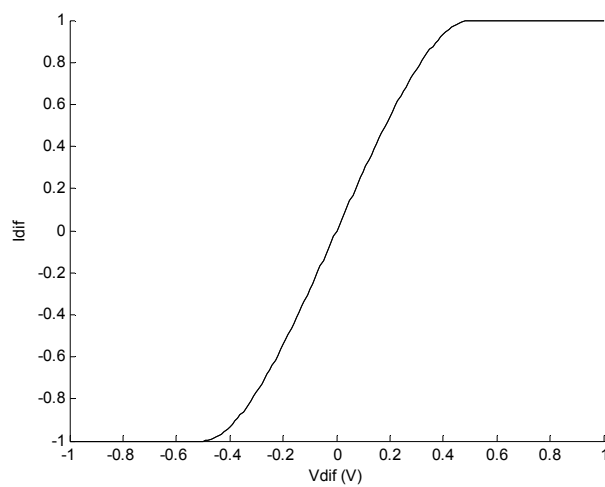


Figura 4.5: Curva característica do par diferencial NMOS.

A equação de transcondutância do par diferencial NMOS é dada por:

$$g_m = \frac{\partial I_{dif}}{\partial V_{dif}} = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} \frac{\frac{4I_t}{\mu_n C_{ox} \frac{W}{L}} - 2V_{dif}^2}{\sqrt{\frac{4I_t}{\mu_n C_{ox} \frac{W}{L}} - (V_{dif})^2}} \quad \text{se } |V_{dif}| \leq \sqrt{\frac{2I_t}{\mu_n C_{ox} \frac{W}{L}}} \quad (4.4)$$

sendo: g_m a transcondutância do par diferencial NMOS.

A Figura 4.6 apresenta uma curva característica de transcondutância para um par diferencial NMOS.

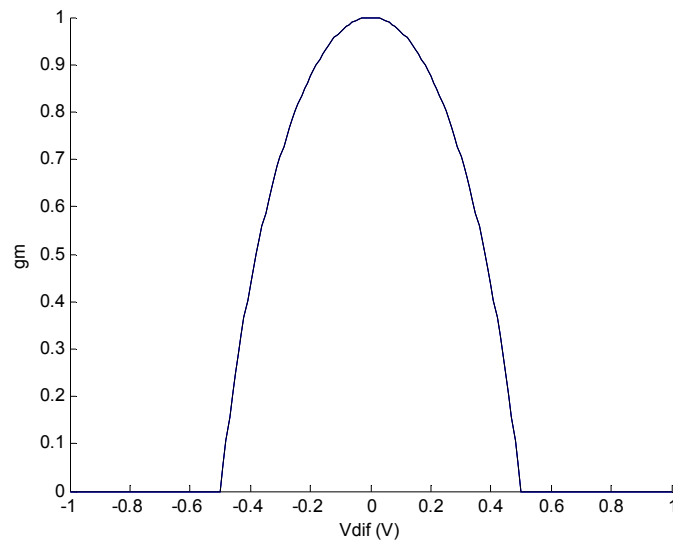


Figura 4.6: Curva de transcondutância do par diferencial NMOS.

A equação 4.5 apresenta a definição da derivada de uma função univariável:

$$f'(x) = \lim_{\Delta \rightarrow 0} \frac{f(x + \Delta) - f(x)}{\Delta} \quad \text{ou} \quad f'(x) = \lim_{\Delta \rightarrow 0} \frac{f(x + \Delta/2) - f(x - \Delta/2)}{\Delta} \quad (4.5)$$

Depreende-se que para Δ suficientemente pequeno pode-se conseguir uma aproximação suficiente da derivada. Portanto, com dois pares diferenciais e uma tensão de *offset* aplicada à entrada de um deles obtém-se uma implementação eletrônica aproximada para a derivada. Assim, tem-se que:

$$f'(x) = \frac{1}{\Delta} [f(x + \Delta) - f(x)] + \varepsilon(\Delta) \quad \text{ou} \quad f'(x) = \frac{1}{\Delta} [f(x + \Delta/2) - f(x - \Delta/2)] + \varepsilon(\Delta) \quad (4.6)$$

O fator $1/\Delta$ é tão-somente um ganho e sua importância depende da aplicação.

Em outras palavras, concluímos que a expressão:

$$g(x) = [f(x + \Delta) - f(x)] \cong \Delta \cdot f'(x) \quad (4.7)$$

Portanto,

$$I_{d5} = I_{b3} - I_{d2} - I_{d4} \quad (4.11)$$

$$I_{d6} = I_{b4} - I_{d1} - I_{d3} \quad (4.12)$$

No nó de saída, a corrente I_o fica:

$$I_o = I_{d8} - I_{d6} = I_{d5} - I_{d6} \quad (4.13)$$

Então:

$$I_o = I_{b3} - I_{d2} - I_{d4} - I_{b4} + I_{d1} + I_{d3} \quad (4.14)$$

Como

$$I_{b3} = I_{b4}, \quad (4.15)$$

Então

$$I_{b3} - I_{b4} = 0 \quad (4.16)$$

Portanto,

$$I_o = -I_{d2} - I_{d4} + I_{d1} + I_{d3} \quad (4.17)$$

Finalmente:

$$I_o = (I_{d1} - I_{d2}) - (I_{d4} - I_{d3}) \quad (4.18)$$

$$I_o = \Delta I_{d1-2} - \Delta I_{d4-3} \quad (4.19)$$

As correntes diferenciais enquanto um dos transistores de cada par diferencial não está na região de corte são dadas por:

$$\Delta I_{d1-2} = \frac{k'Wv_{id1-2}}{2L} \sqrt{\frac{4I_{b1}}{k'(W/L)} - v_{id1-2}^2} \quad (4.20)$$

$$\Delta I_{d4-3} = \frac{k'Wv_{id4-3}}{2L} \sqrt{\frac{4I_{b2}}{k'(W/L)} - v_{id4-3}^2} \quad (4.21)$$

$$\text{Sendo:} \quad v_{id1-2} = v_i - v_{b1} \quad \text{e} \quad v_{id4-3} = v_i - v_{b2} \quad (4.22)$$

Na Figura 4.8, é mostrada a corrente de saída obtida via simulação SPICE, utilizando os modelos para os MOSFETs NMOS e PMOS presentes nos CIs ALD1105, ALD1106 e ALD1107. A diferença de tensão foi obtida fazendo $V_{b1} = 50 \text{ mV}$ e $V_{b2} = -50 \text{ mV}$ e as correntes de polarização são as mostradas na Figura 4.7, sendo $V_{DD} = +2\text{V}$ e $V_{SS} = -2\text{V}$. Na Figura 4.9 é mostrada a variação na corrente de saída para uma variação simétrica na diferença de tensão ($V_{b1} - V_{b2}$).

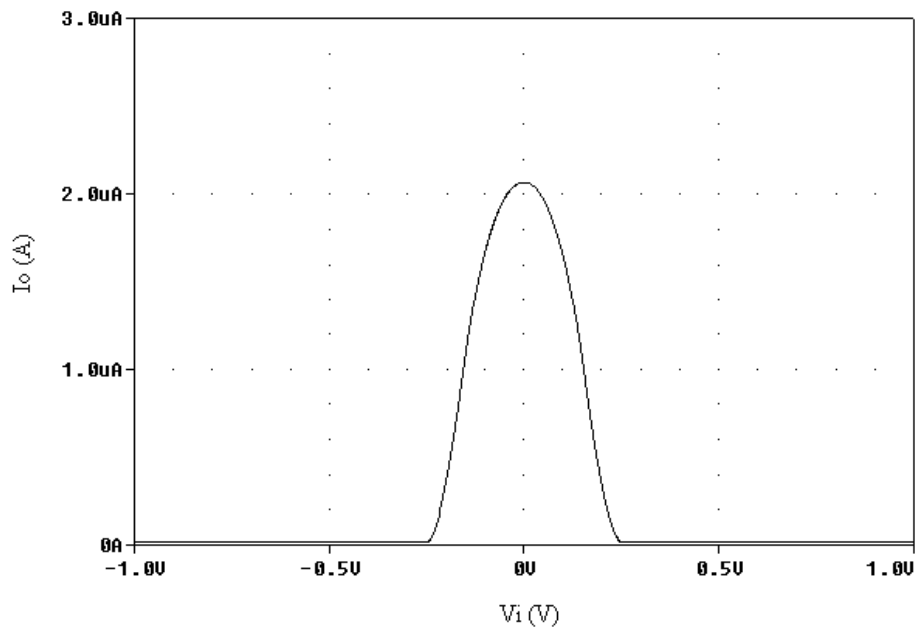


Figura 4.8: Corrente de saída do circuito CMOS RBF₁.

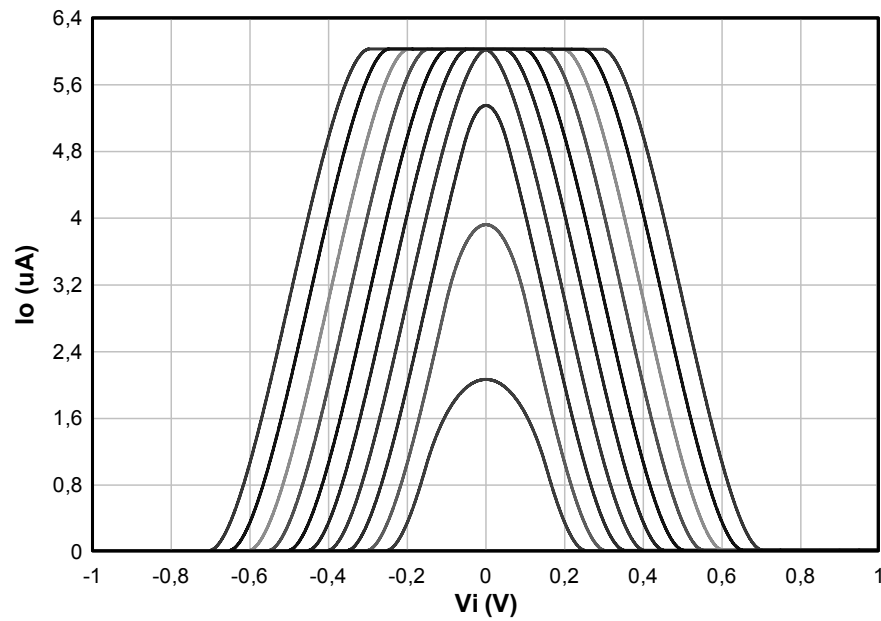


Figura 4.9: Corrente de saída em função da variação de $(V_{b1} - V_{b2})$ de 100 mV a 1 V.

Na Figura 4.10 é mostrado como a variação do centro é obtidas com a polarização adequada das tensões V_{b1} e V_{b2} . Na Figura 4.11 e 4.12, mostra-se como a variação da corrente de polarização e das tensões V_{b1} e V_{b2} permite o ajuste da amplitude e da largura da RBF.

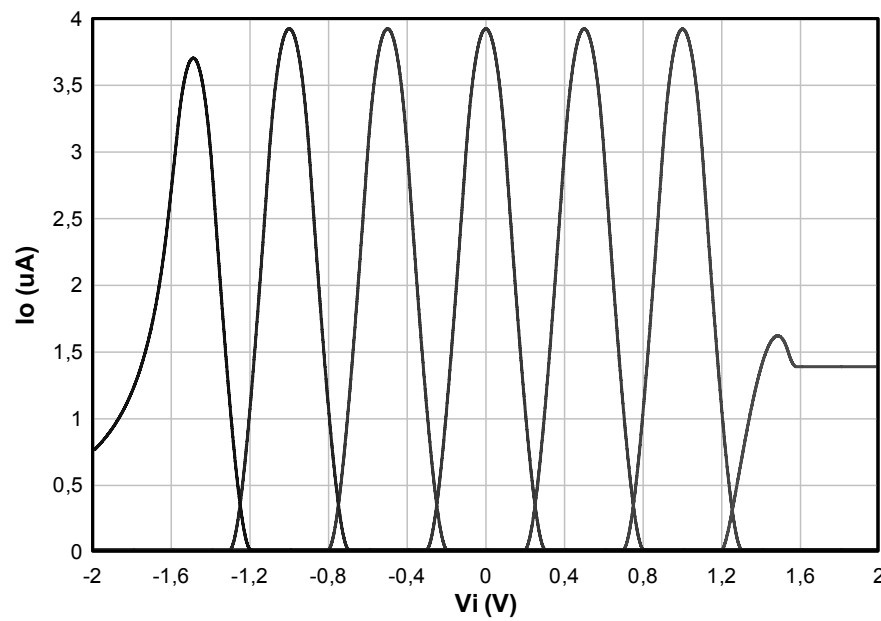


Figura 4.10: Variação no centro da RBF em função da variação de V_{b1} e V_{b2} .

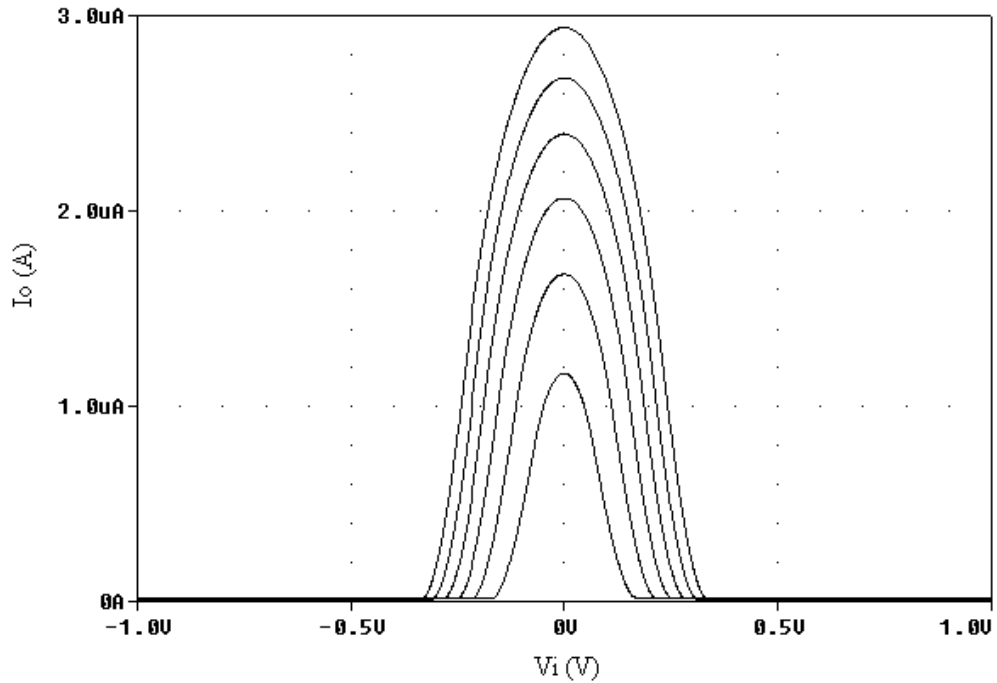


Figura 4.11: Corrente de saída em função da variação de $I_{b1}=I_{b2}$ entre 1 μ A e 6 μ A para $V_{b1}=50$ mV e $V_{b2}=-50$ mV.

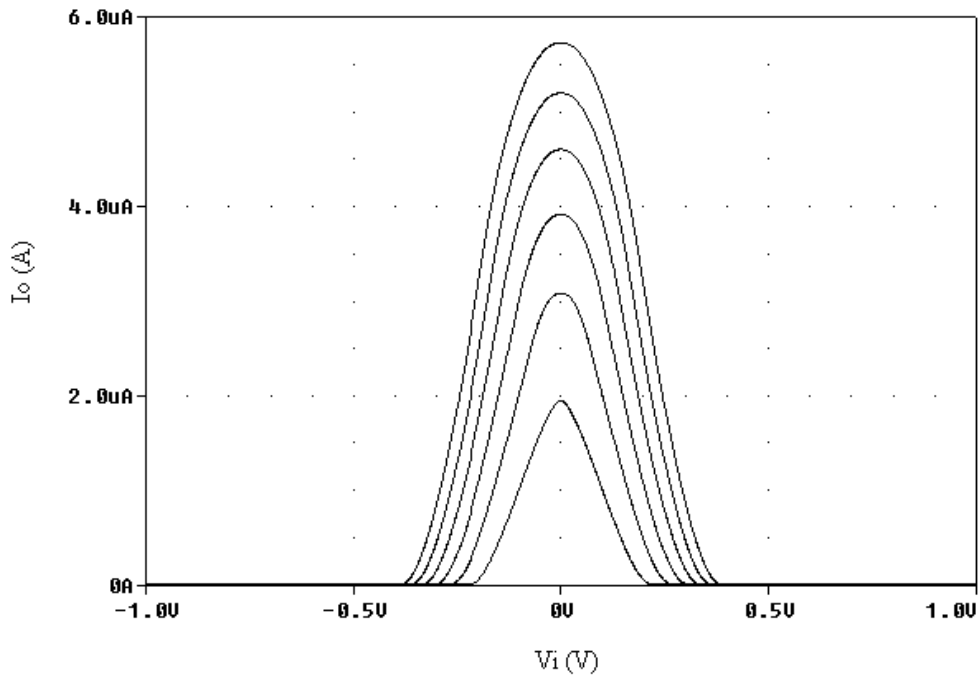


Figura 4.12: Corrente de saída em função da variação de $I_{b1}=I_{b2}$ entre 1 μ A e 6 μ A para $V_{b1}=100$ mV e $V_{b2}=-100$ mV.

4.6 Circuitos RBF₁ Multidimensionais

Há basicamente três maneiras de se obter uma função RBF multidimensional a partir de funções RBF unidimensionais. A primeira consiste em multiplicar várias funções unidimensionais. A segunda é a utilização de um circuito para calcular a norma, em geral euclidiana, na entrada. E a terceira é utilizar alguma forma de correlação entre correntes de saída. O circuito multidimensional aqui proposto utiliza a terceira opção. A corrente de saída de uma RBF₁ unidimensional é utilizada para controlar a corrente de polarização do circuito RBF₁ seguinte, obtendo assim uma correlação de correntes que permite a operação multidimensional. Como exemplo, é apresentado um circuito RBF₁ CMOS bidimensional na Figura 4.13. O circuito foi simulado usando SPICE e a corrente de saída obtida é mostrada na Figura 4.14.

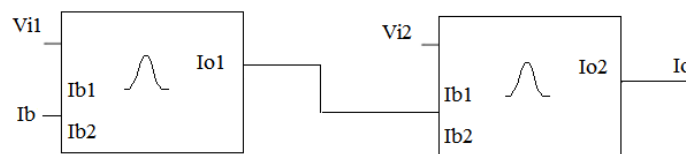


Figura 4.13: Circuito RBF₁ Bidimensional.

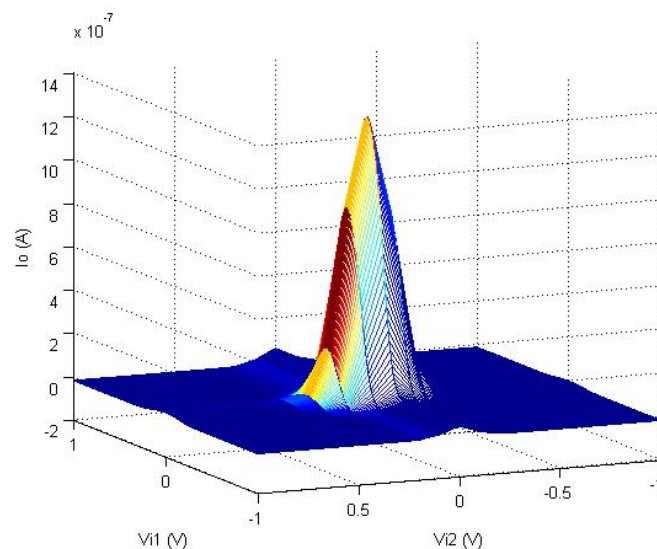


Figura 4.14: Corrente de saída do circuito RBF₁ CMOS bidimensional.

4.7 Circuito RBF₁ BiCMOS

O circuito RBF₁ BiCMOS é apresentado na Figura 4.15. Sua operação é similar ao circuito CMOS. Q1-Q2 e Q3-Q4 formam dois pares diferenciais bipolares. O sinal de saída é obtido pela subtração de duas funções sigmóides (tanh) produzindo uma função secante hiperbólica quadrática, a qual pode ser utilizada para aproximar uma função gaussiana. Da mesma forma que o circuito CMOS, a fim de obter uma boa aproximação, a diferença de tensão deve ser pequena. Neste caso, também foi usado 100 mV. A subtração é realizada pela estrutura cascode dobrado.

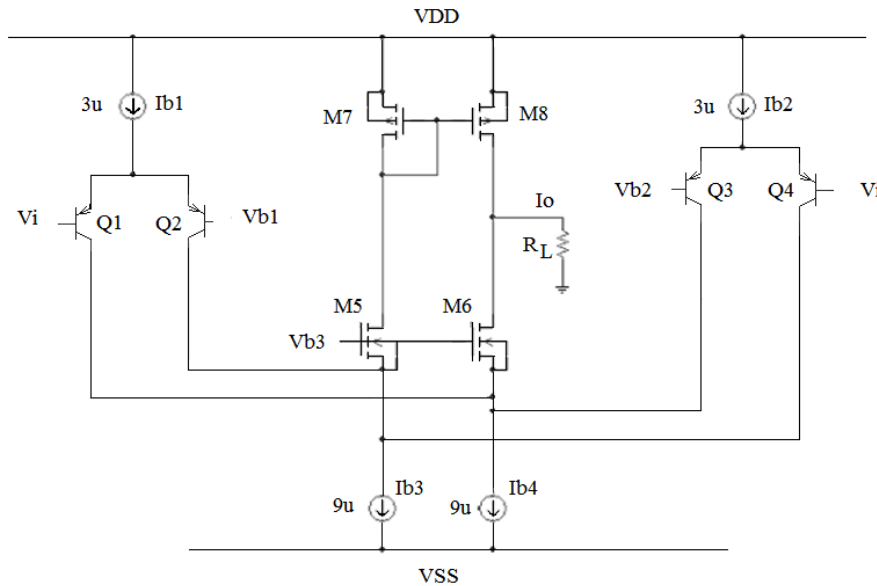


Figura 4.15: Esquema simplificado do circuito RBF₁ BiCMOS.

Vamos obter uma expressão analítica para a corrente de saída para o circuito BiCMOS. No nó de saída tem-se:

$$I_o = I_{d8} - I_{d6} = I_{d5} - I_{d6} \quad (4.23)$$

Portanto,

$$I_o = I_{b3} - I_{c2} - I_{c4} - I_{b4} + I_{c1} + I_{c3} \quad (4.24)$$

Como $I_{b3} = I_{b4}$, (4.25)

Então,

$$I_{b3} - I_{b4} = 0 \quad (4.26)$$

Desta forma:

$$I_o = -I_{c2} - I_{c4} + I_{c1} + I_{c3} \quad (4.27)$$

$$I_o = (I_{c1} - I_{c2}) - (I_{c4} - I_{c3}) \quad (4.28)$$

Finalmente:

$$I_o = \Delta I_{c1-2} - \Delta I_{c4-3} \quad (4.29)$$

As correntes diferenciais dos transistores de cada par diferencial são dadas por:

$$\Delta I_{c1-2} = \alpha_F I_{b1} \tanh(-v_{id1-2} / 2v_T) \quad (4.30)$$

$$\Delta I_{c4-3} = \alpha_F I_{b2} \tanh(-v_{id4-3} / 2v_T) \quad (4.31)$$

Sendo

$$v_{id1-2} = v_i - v_{b1} \quad \text{e} \quad v_{id4-3} = v_i - v_{b2} \quad (4.32)$$

A diferença de tensão foi obtida fazendo $V_{b1} = 50 \text{ mV}$ e $V_{b2} = -50 \text{ mV}$. Na Figura 4.16, é mostrada a função obtida via simulação SPICE.

O princípio de funcionamento é análogo ao do circuito RBF₁, exceto na maneira de obter a subtração de correntes na saída. Neste caso, isso é feito utilizando apenas espelhos de corrente. Da mesma forma que o circuito RBF₁, a fim de obter uma boa aproximação, a diferença de tensão deve ser pequena e, neste caso, o valor utilizado também foi 100 mV. A operação de subtração é realizada pelos espelhos de corrente CM₁, CM₂ e CM₃. A corrente de saída I_o é dada por:

$$I_o = (I_2 - I_1) - (I_3 - I_4) \quad (4.33)$$

As correntes diferenciais enquanto um dos transistores de cada par diferencial não esteja na região de corte são dadas por:

$$\Delta I_{d1-2} = \frac{k'Wv_{id1-2}}{2L} \sqrt{\frac{4I_{b1}}{k'(W/L)} - v_{id1-2}^2} \quad (4.34)$$

$$\Delta I_{d4-3} = \frac{k'Wv_{id4-3}}{2L} \sqrt{\frac{4I_{b2}}{k'(W/L)} - v_{id4-3}^2} \quad (4.35)$$

sendo: $v_{id1-2} = v_i - v_{b1}$ e $v_{id4-3} = v_i - v_{b2}$ (4.36)

Na Figura 4.18, é mostrada a corrente de saída obtida via simulação SPICE, utilizando os modelos para os MOSFETs NMOS e PMOS dos CIs ALD1105, ALD1106 e ALD1107. A polarização do circuito foi obtida com V_{DD} = +2 V, V_{SS} = -2 V, V_{b1} = 50 mV, V_{b2} = -50 mV e corrente de polarização I_b = 3 μA, com. Na Figura 4.19 é mostrada a variação na corrente de saída para uma variação simétrica na diferença de tensão (V_{b1}-V_{b2}). Na Figura 4.20 é mostrado como a variação do centro é obtidas com a polarização adequada das tensões V_{b1} e V_{b2}. Nas Figuras 4.21 e 4.22, mostra-se como a variação da corrente de polarização e das tensões V_{b1} e V_{b2} permite o ajuste da amplitude e da largura da RBF.

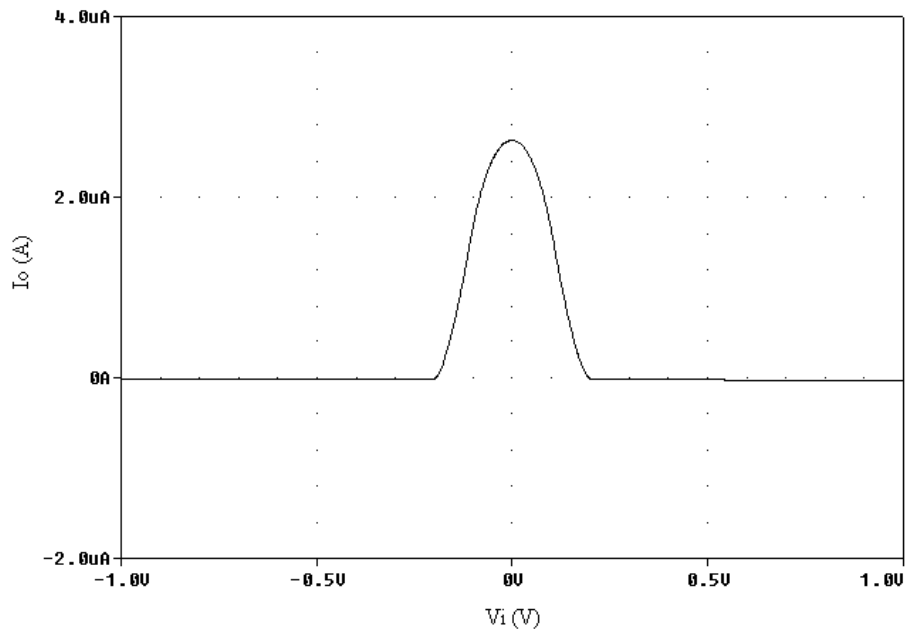


Figura 4.18: Corrente de saída do circuito CMOS RBF₂.

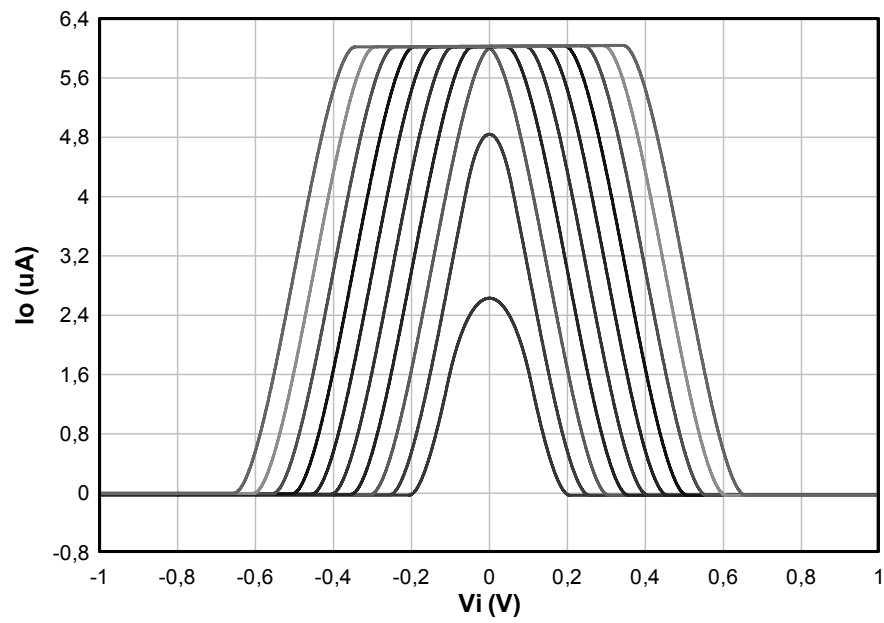


Figura 4.19: Corrente de saída em função da variação de $(V_{b1} - V_{b2})$ de 100 mV a 1 V.

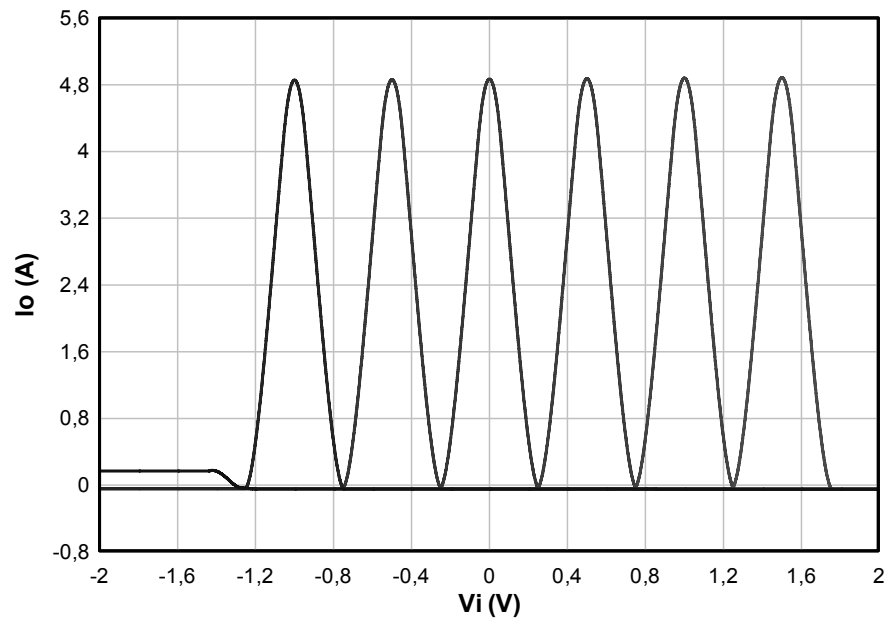


Figura 4.20: Variação no centro da RBF em função da variação de V_{b1} e V_{b2} .

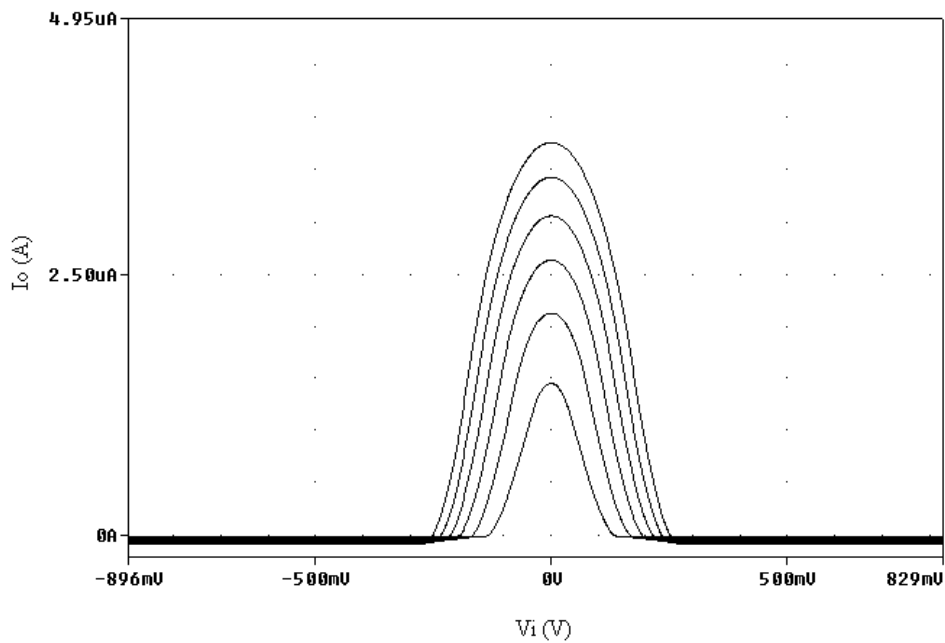


Figura 4.21: Corrente de saída em função da variação de $I_{b1}=I_{b2}$ entre $1 \mu\text{A}$ e $6 \mu\text{A}$ para $V_{b1}=50 \text{ mV}$ e $V_{b2}=-50 \text{ mV}$.

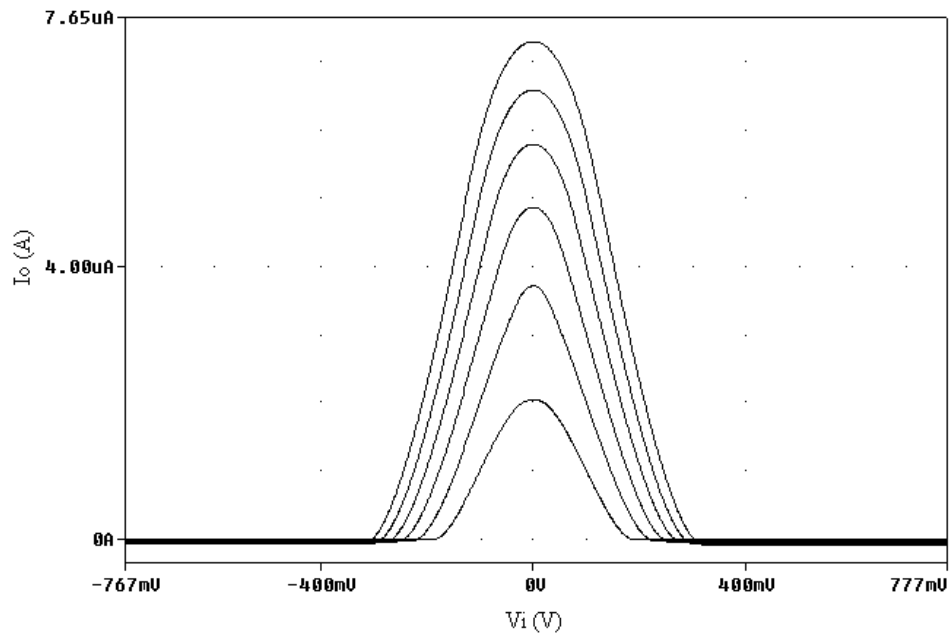


Figura 4.22: Corrente de saída em função da variação de $I_{b1}=I_{b2}$ entre $1\ \mu\text{A}$ e $6\ \mu\text{A}$ para $V_{b1}=100\ \text{mV}$ e $V_{b2}=-100\ \text{mV}$.

4.9 Circuitos RBF_2 Multidimensionais

O circuito RBF_2 multidimensional proposto utiliza a corrente de saída de uma RBF_2 unidimensional para controlar a corrente de polarização do circuito RBF_2 seguinte, obtendo assim a correlação de correntes que permite a operação multidimensional, de forma análoga ao circuito RBF_1 multidimensional. Como exemplo, é apresentado um circuito RBF_2 CMOS bidimensional na Figura 4.23. O circuito foi simulado usando SPICE e a corrente de saída obtida é mostrada na Figura 4.24.



Figura 4.23: Circuito RBF_2 Bidimensional.

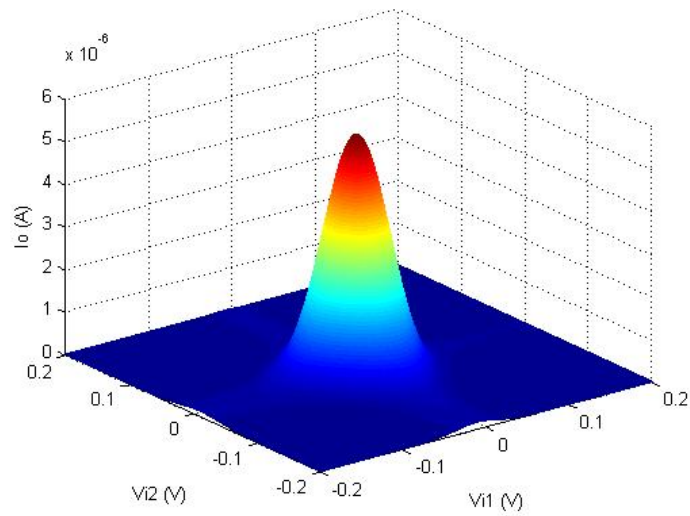


Figura 4.24: Corrente de saída do circuito RBF₂ CMOS bidimensional.

4.10 Circuito CMOS RBF₂ Completo

A fim de se obter uma variação dos parâmetros centro e largura da função, pode-se acrescentar na entrada do circuito um par diferencial seguido por um conversor I/V linear, conforme é mostrado na Figura 4.25.

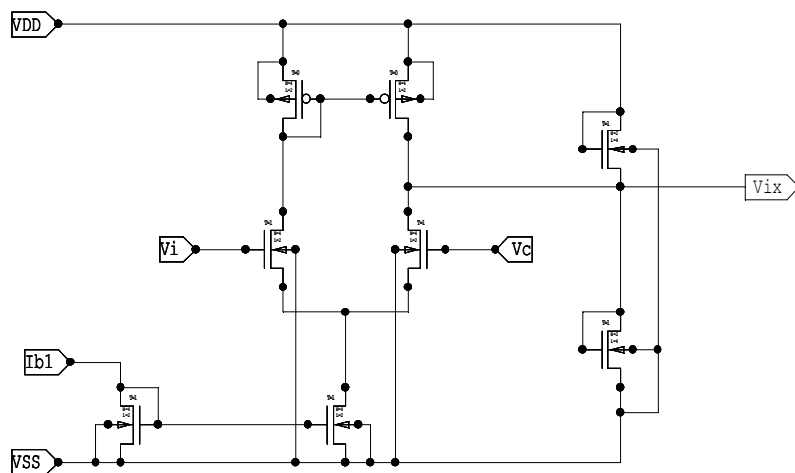


Figura 4.25: Circuito para variação de centro e largura.

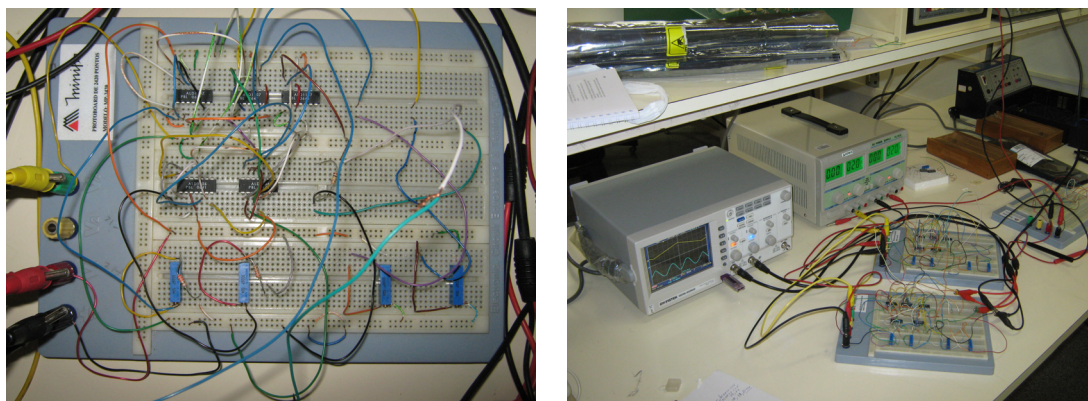
Capítulo 5

Implementação Prática e Resultados Obtidos

Neste capítulo serão apresentados os circuitos RBF_1 e RBF_2 unidimensionais implementados e os resultados experimentais obtidos. Serão apresentados também os resultados obtidos para o circuito RBF_1 bidimensional implementado.

5.1 Circuito RBF_1

Foram implementados dois circuitos RBF_1 unidimensionais, os quais foram também configurados para operação bidimensional. Os circuitos foram montados em *proto-board* com componentes comerciais ALD1105, ALD1106 e ALD1107 (*MOSFET matched-pair arrays*). Fotos da montagem experimental são apresentadas na Figura 5.1



(a)

(b)

Figura 5.1: Montagem experimental do circuito CMOS RBF_1 . (a) Detalhe do circuito (b) Implementação de dois circuitos RBF_1 .

5.1.1 Circuito RBF₁ Unidimensional

A fim de verificar a funcionalidade do circuito RBF₁ unidimensional, foi aplicada uma onda triangular na entrada do circuito, e a tensão na saída foi medida com o osciloscópio. Nas Figuras 5.2 e 5.3 são mostradas fotos do sinal de saída obtido no osciloscópio para o sinal triangular na entrada do circuito.

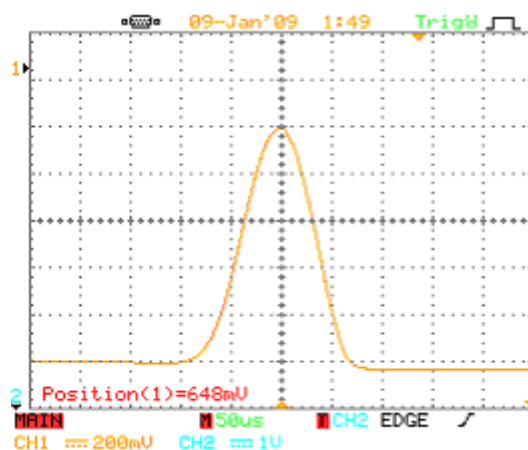


Figura 5.2: Resultados experimentais obtidos para o circuito CMOS RBF₁.

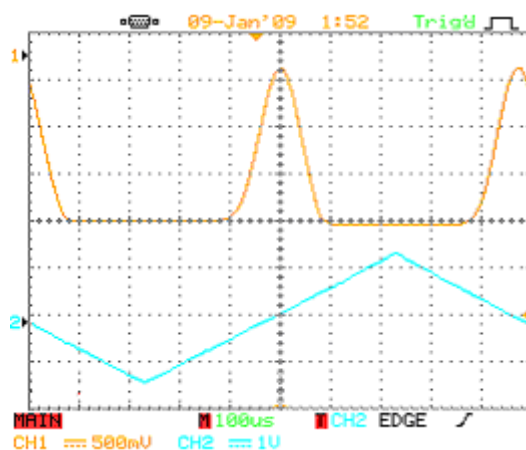


Figura 5.3: Resultados experimentais obtidos para o circuito CMOS RBF₁ mostrando a onda triangular na entrada e a função de base radial obtida.

Verificou-se que o circuito apresentou um bom funcionamento até aproximadamente uma frequência de 100 kHz para o sinal triangular na entrada.

5.1.2 Dois Circuitos RBF₁ Conectados em Paralelo

As saídas de ambos os circuitos foram conectadas à resistência de carga na saída e a tensão obtida é mostrada na Figura 5.4. São mostradas também a saída de cada circuito em separado.

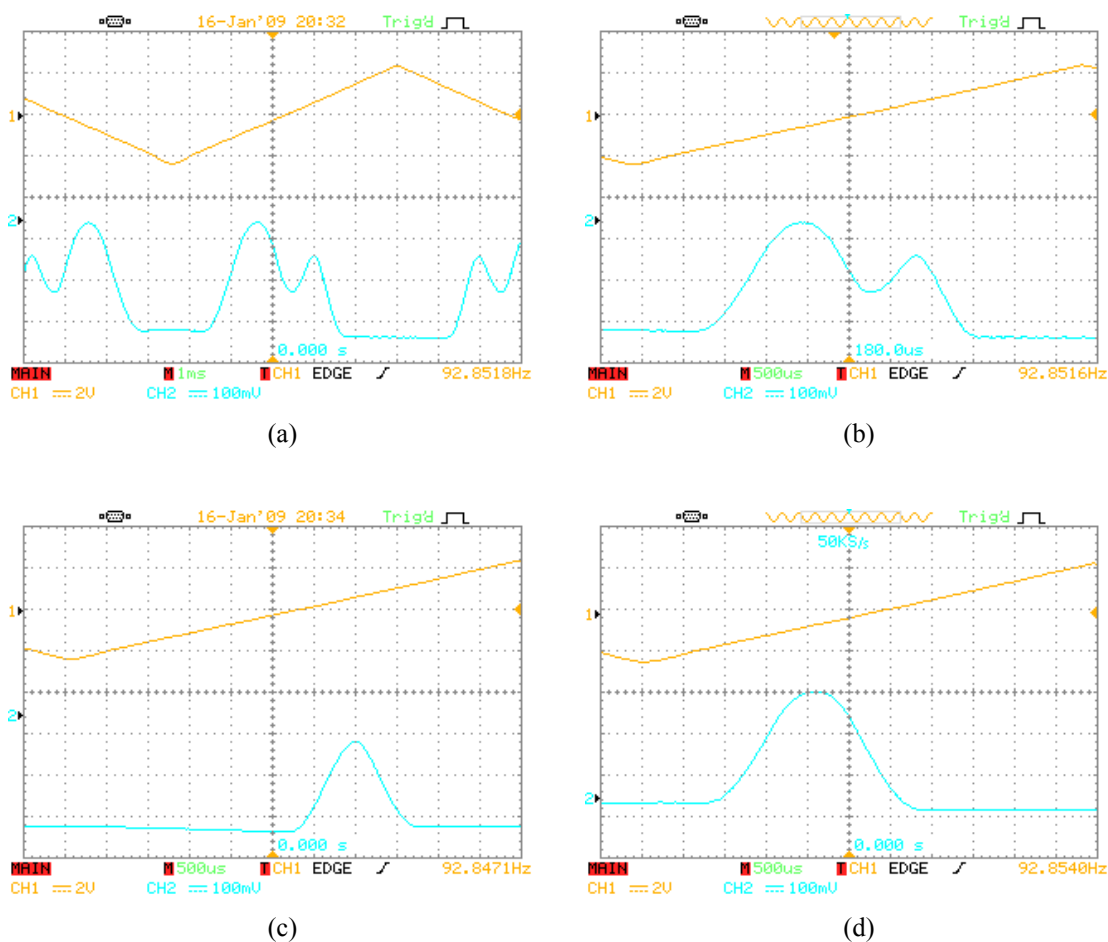


Figura 5.4: (a) Resultados experimentais obtidos para os dois circuitos RBF₁ conectados em paralelo. (b) Ampliação da escala de tempo. (c) Saída do circuito 1. (d) Saída do circuito 2.

Com as saídas das duas gaussianas somadas, retirou-se a resistência de carga, ou seja, fez-se $R_L \approx \infty$ e foi verificada a saída obtida. O resultado é apresentado na Figura 5.5.

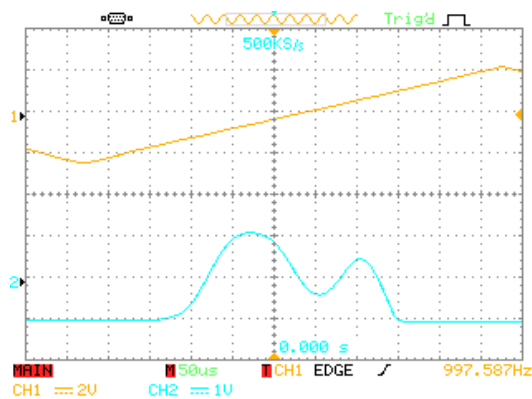


Figura 5.5: Resultados obtidos para saídas conectadas e R_L em aberto.

5.1.3 Aproximação de uma função senoidal

Os circuitos foram ajustados para aproximar uma função senoidal. Os resultados obtidos são apresentados na Figura 5.6. As duas funções são mostradas em separado nas Figuras 5.7 e 5.8. Nota-se que uma função consegue aproximar uma senóide.

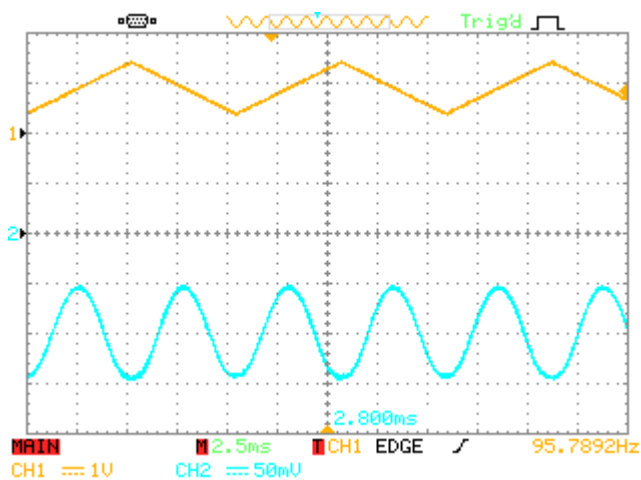


Figura 5.6: Aproximação de uma senóide.

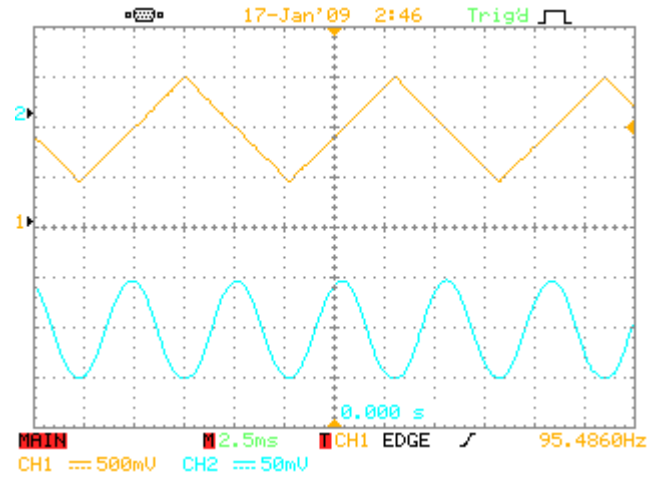


Figura 5.7: Aproximação de uma senóide: circuito 1.

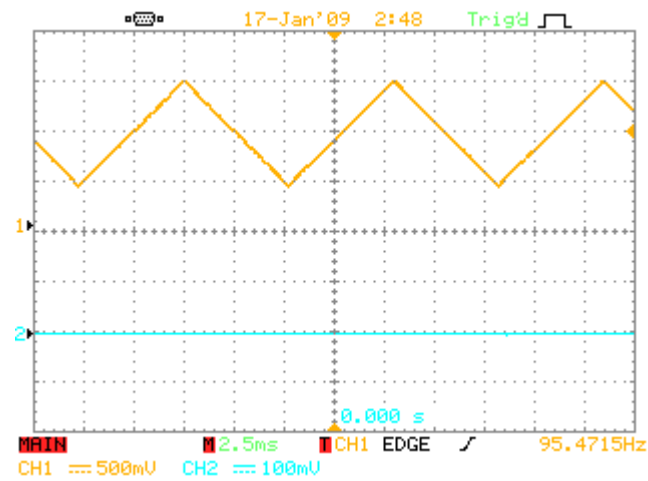
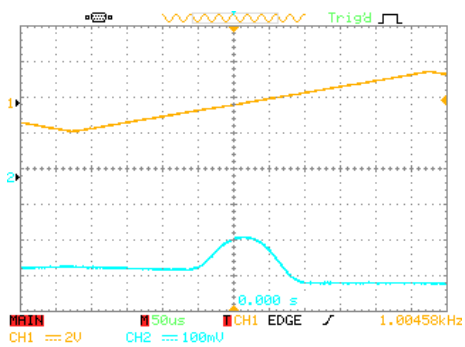


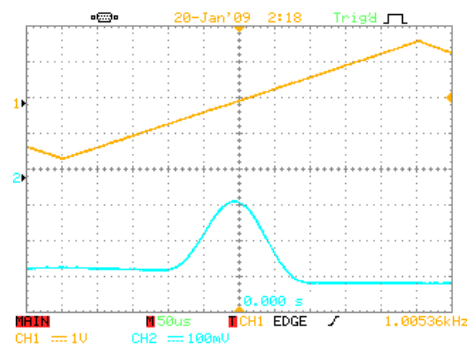
Figura 5.8: Aproximação de uma senóide: circuito 2.

5.1.4 Variação de parâmetros da RBF₁

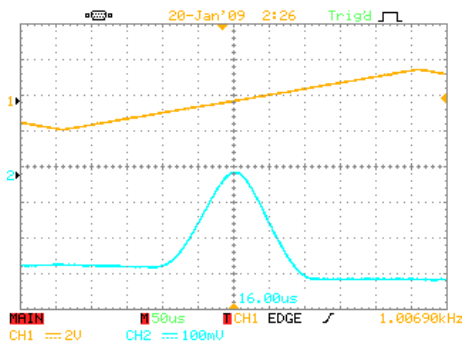
Na Figura 5.9 são apresentados os resultados obtidos para variações nas tensões de polarização V_{b1} e V_{b2} do circuito RBF₁ unidimensional e uma onda triangular na entrada do circuito. As condições de polarização utilizadas foram $I_{b1}=25,6 \mu\text{A}$ e $I_{b2}=93,3 \mu\text{A}$, e uma resistência de carga na saída $R_L = 10 \text{ k}\Omega$.



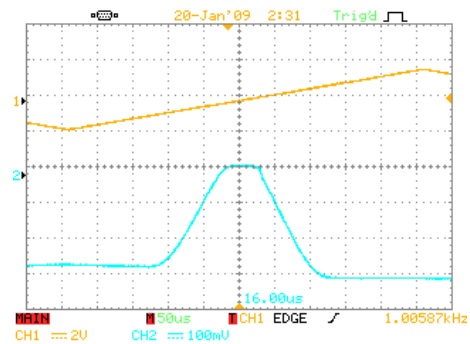
(a) $V_{b1}=0,10\text{V}$ E $V_{b2}=-0,10\text{V}$.



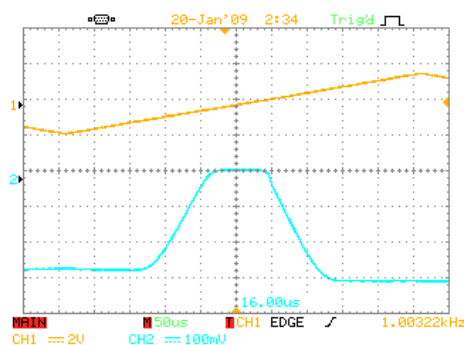
(b) $V_{b1}=0,20\text{V}$ E $V_{b2}=-0,20\text{V}$.



(c) $V_{b1}=0,30\text{V}$ E $V_{b2}=-0,30\text{V}$.



(d) $V_{b1}=0,40\text{V}$ E $V_{b2}=-0,40\text{V}$.



(e) $V_{b1}=0,50\text{V}$ E $V_{b2}=-0,50\text{V}$.

Figura 5.9: Resultados obtidos para variações nas tensões de polarização V_{b1} e V_{b2} .

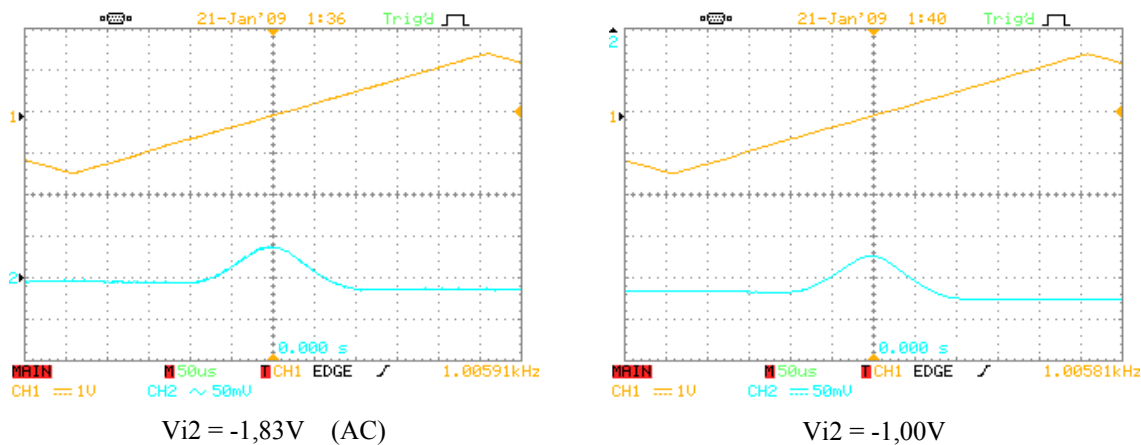
5.2 Circuito RBF₁ Bidimensional

Os dois circuitos RBF₁ foram conectados em série a fim de obter a operação bidimensional. A corrente de saída do circuito 2 foi conectada ao circuito 1 por meio de um espelho de corrente obtido com um CI ALD1105. Dessa forma, I_{b1} (circuito1) $\approx I_o$ (circuito2). A polarização do circuito é mostrada na Tabela 5.1.

Tabela5.1: Polarização do circuito RBF₁ bidimensional.

CIRC 1:	CIRC 2:
I_{b1} (circ1) $\approx I_o$ (circ2)	$I_{b1}=34,2 \mu\text{A}$
$I_{b2}=79 \mu\text{A}$	$I_{b2}=96,3 \mu\text{A}$
$V_{b1}=0,22 \text{ V}$	$V_{b1}=0,25 \text{ V}$
$V_{b2}=-0,22 \text{ V}$	$V_{b2}=-0,25 \text{ V}$
$\Delta V=V_{b1}-V_{b2}=440 \text{ mV}$	$\Delta V=V_{b1}-V_{b2}=500 \text{ mV}$

Inicialmente, a resistência de carga na saída foi definida como $R_L = 10 \text{ k}\Omega$. Foi aplicada uma onda triangular na entrada V_{i1} do circuito 1 e uma tensão ajustável por meio de um potenciômetro de precisão na entrada V_{i2} do circuito 2. A saída da função bidimensional foi verificada para vários valores de V_{i2} e os resultados são apresentados na Figura 5.10.



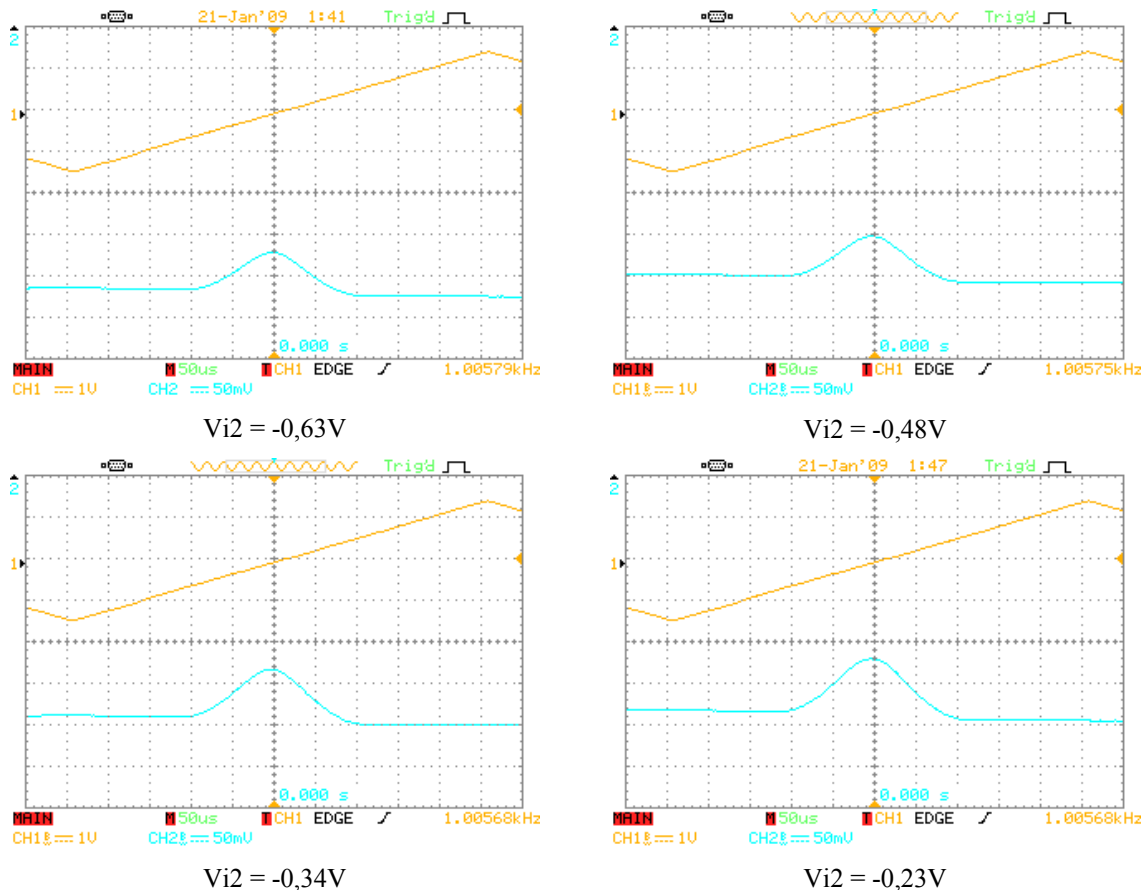
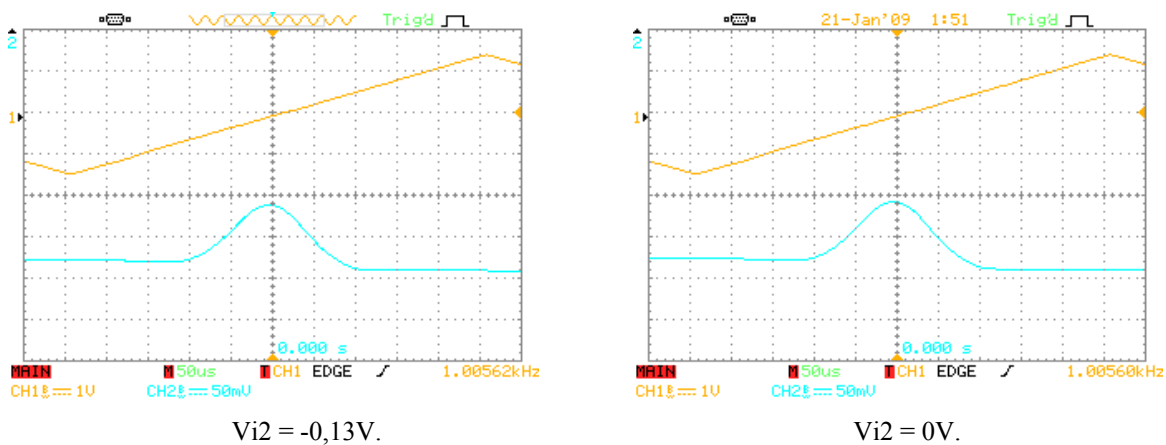
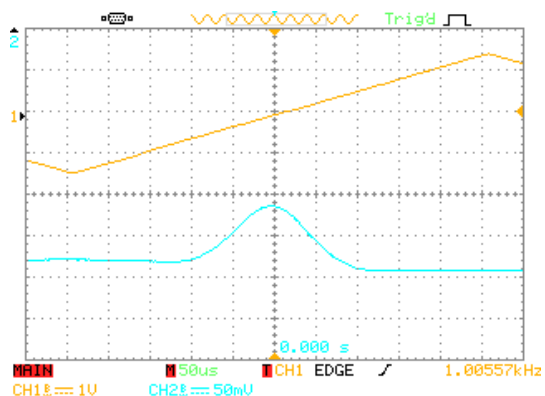
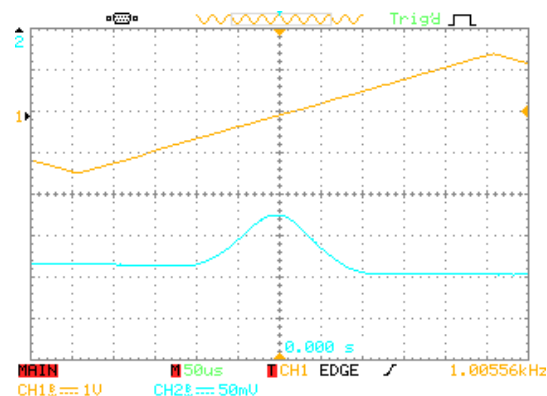


Figura 5.10a: Resultados obtidos para o circuito RBF1 bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 10 \text{ k}\Omega$.

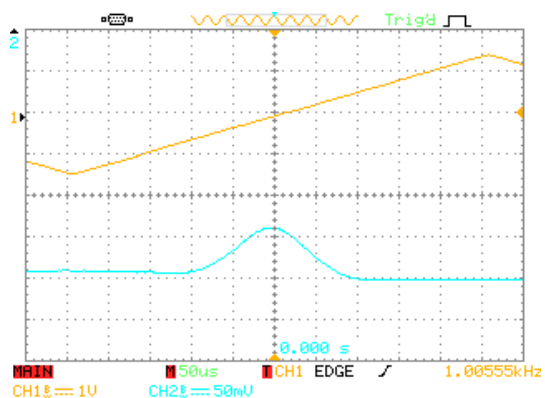




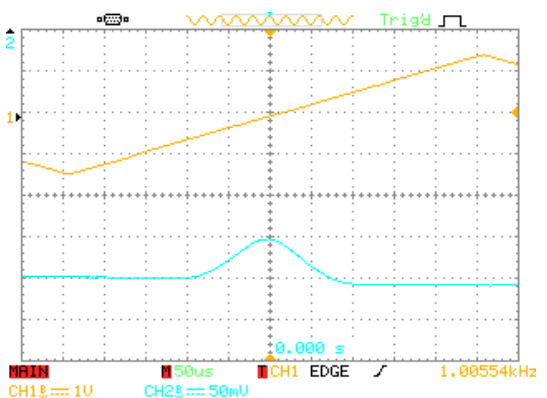
Vi2 = +0,13V.



Vi2 = +0,23V.

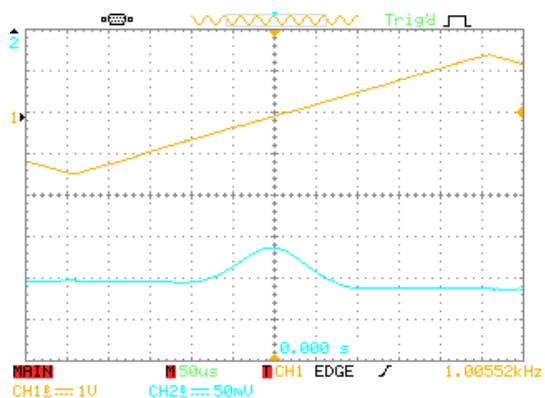


Vi2 = +0,34V.

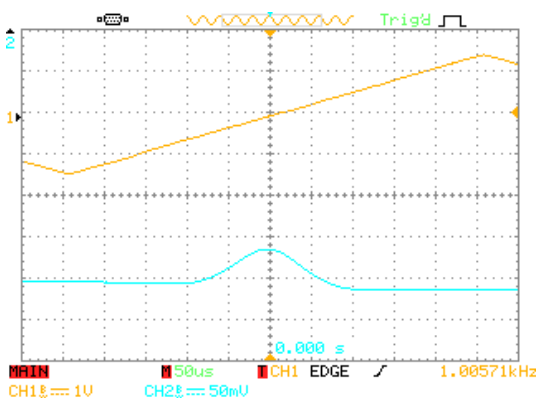


Vi2 = +0,48V.

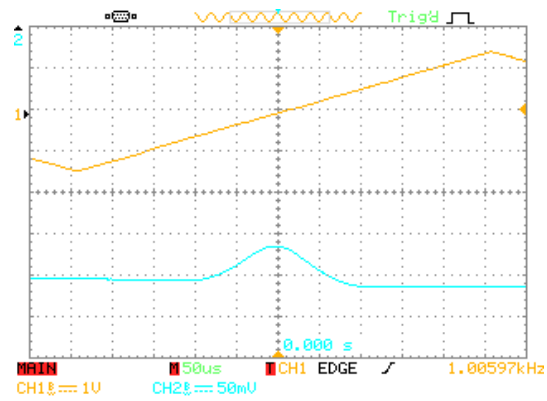
Figura 5.10b: Resultados obtidos para o circuito RBF1 bidimensional para Vi1 triangular e variações na tensão Vi2 com $R_L = 10 \text{ k}\Omega$ (continuação).



Para Vi2 = +0,63V.



Para Vi2 = +1,00V.



Para $V_{i2} = +1,83V$.

Figura 5.10c: Resultados obtidos para o circuito RBF1 bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 10\text{ k}\Omega$ (continuação).

O valor da resistência de carga na saída foi alterado para $R_L = 100\text{ k}\Omega$ e foi repetido o procedimento anterior. A saída da função bidimensional foi verificada para vários valores de V_{i2} e os resultados são apresentados na Figura 5.11.

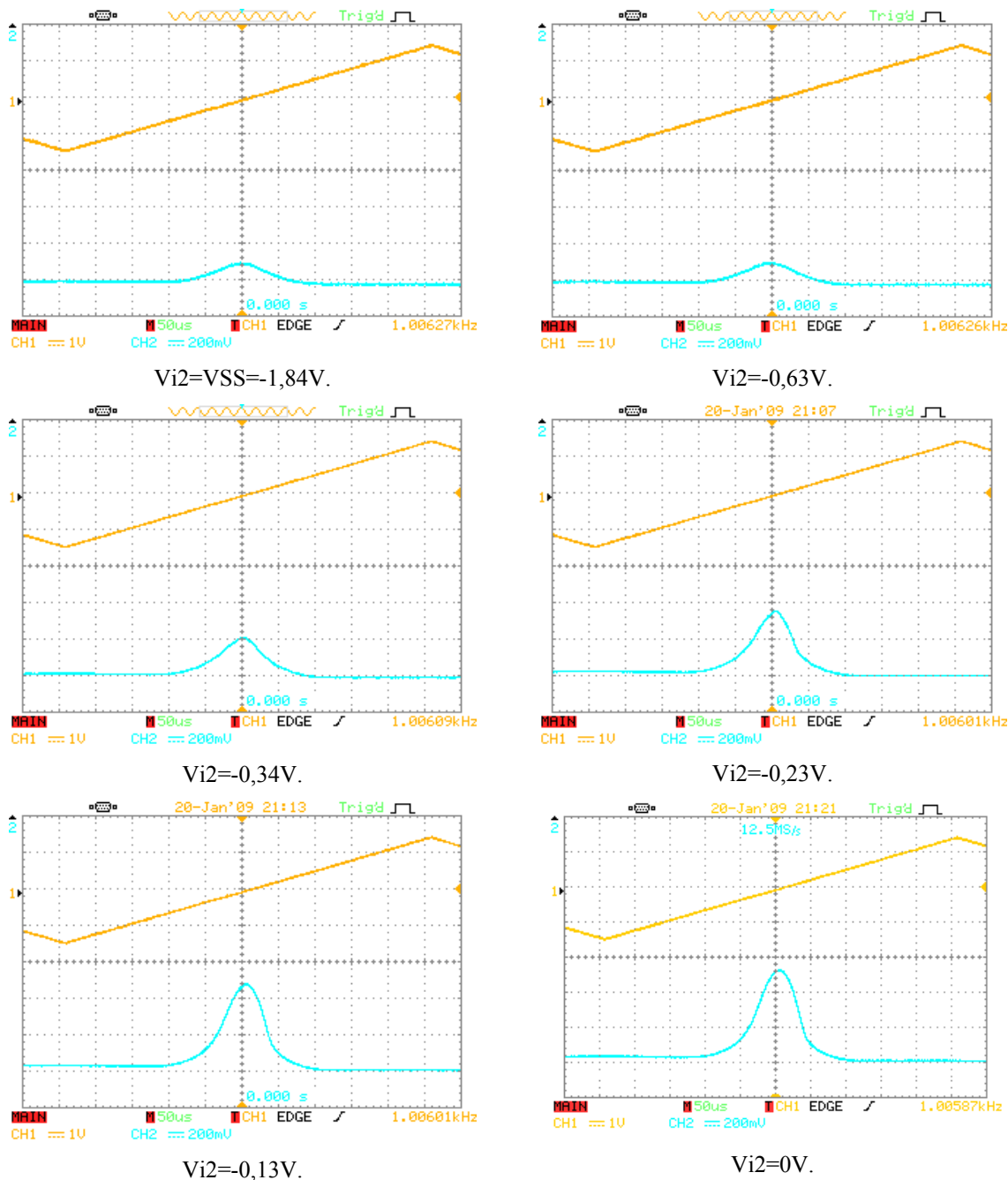


Figura 5.11a: Resultados obtidos para o circuito RBF1 bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 100\text{ k}\Omega$.

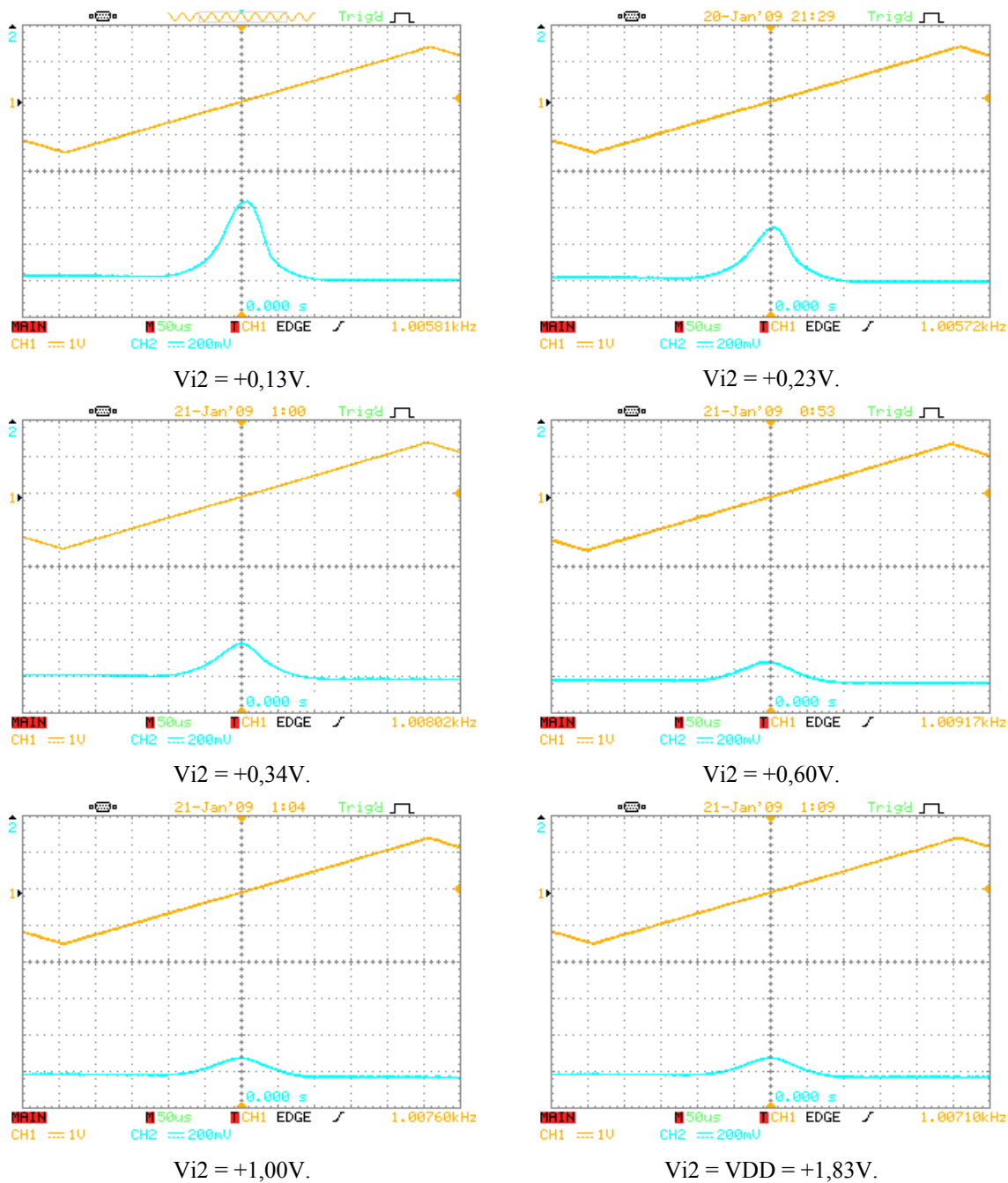


Figura 5.11b: Resultados obtidos para o circuito RBF1 bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = 100\text{ k}\Omega$ (continuação).

O mesmo procedimento foi realizado para uma resistência de carga na saída $R_L = \infty$ (aberto). A saída da função bidimensional foi verificada para vários valores de V_{i2} e os resultados são apresentados na Figura 5.12.

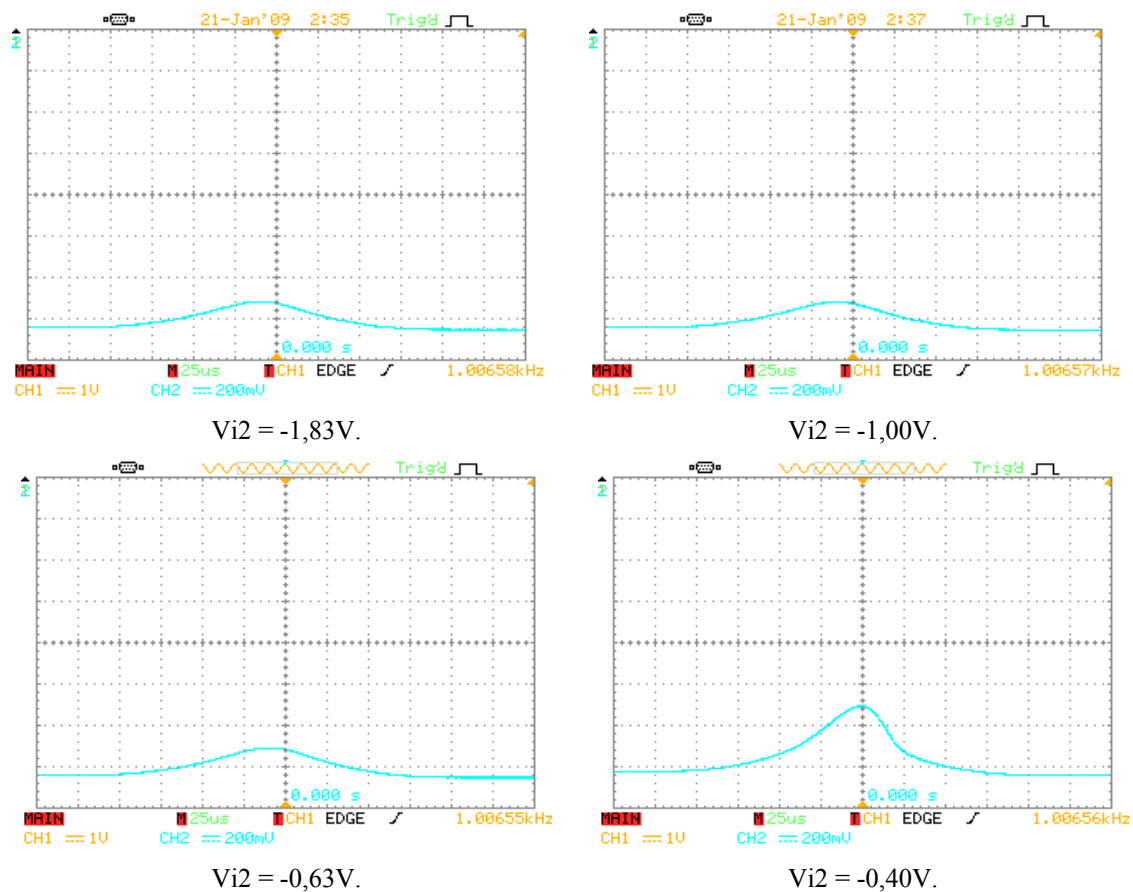


Figura 5.12a: Resultados obtidos para o circuito RBF1 bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = \infty$ (aberto).

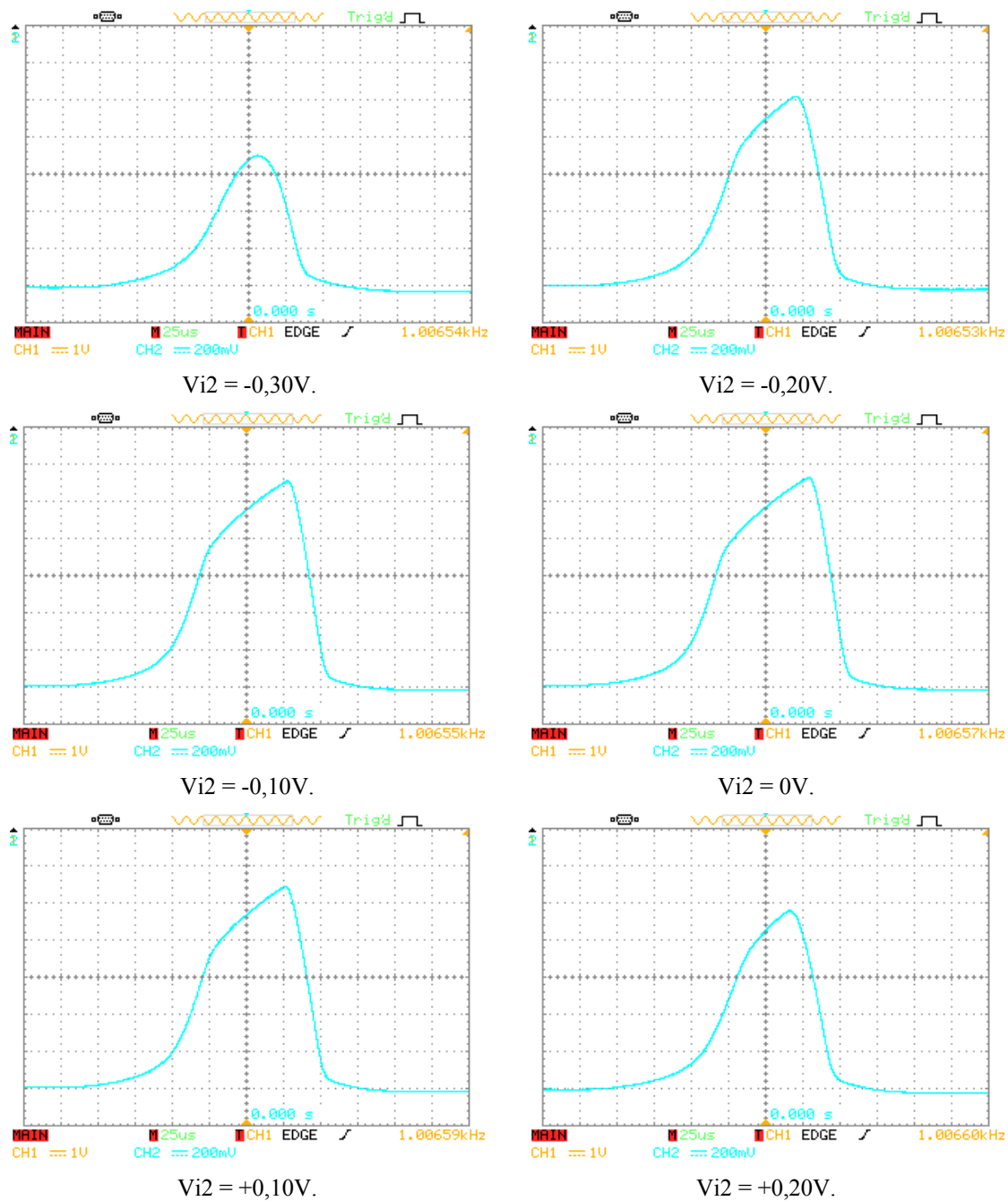


Figura 5.12b: Resultados obtidos para o circuito RBF1 bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = \infty$ (aberto) (continuação).

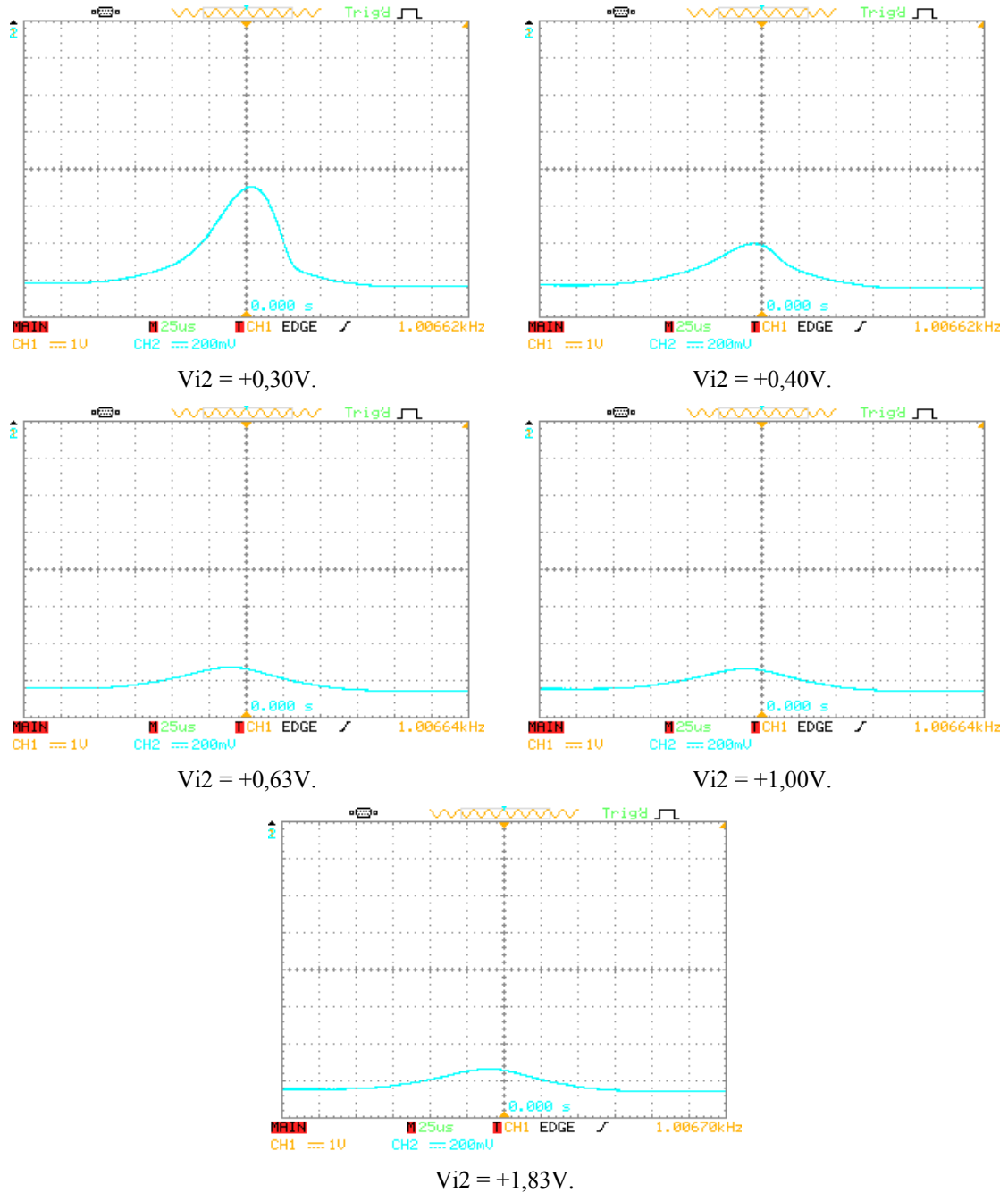


Figura 5.12c: Resultados obtidos para o circuito RBF1 bidimensional para V_{i1} triangular e variações na tensão V_{i2} com $R_L = \infty$ (aberto) (continuação).

Ainda com R_L em aberto foram intercambiadas as tensões nas entradas V_{i1} e V_{i2} . Assim, foi aplicada uma onda triangular na entrada V_{i2} e uma tensão ajustável na entrada V_{i1} . A saída da função bidimensional foi verificada para vários valores de V_{i1} e os resultados são apresentados na Figura 5.13.

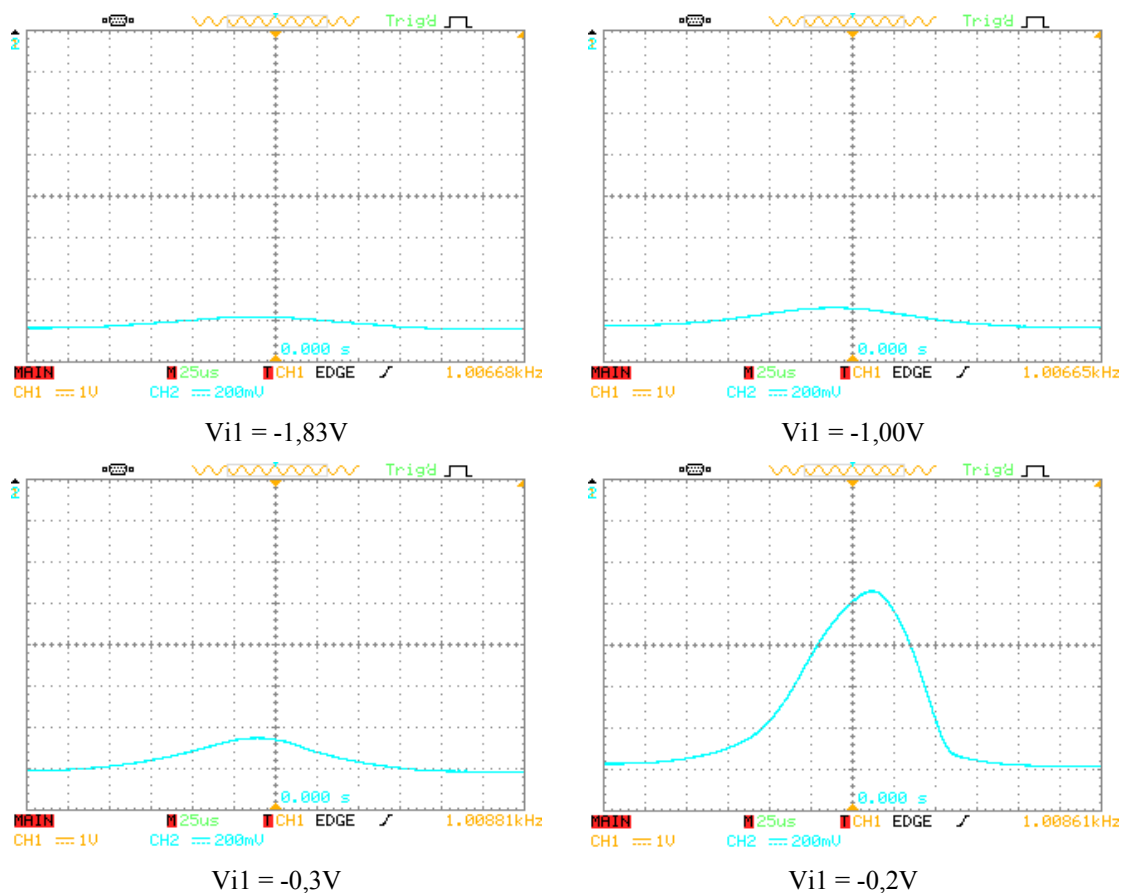


Figura 5.13a: Resultados obtidos para o circuito RBF1 bidimensional para V_{i2} triangular e variações na tensão V_{i1} com $R_L = \infty$ (aberto).

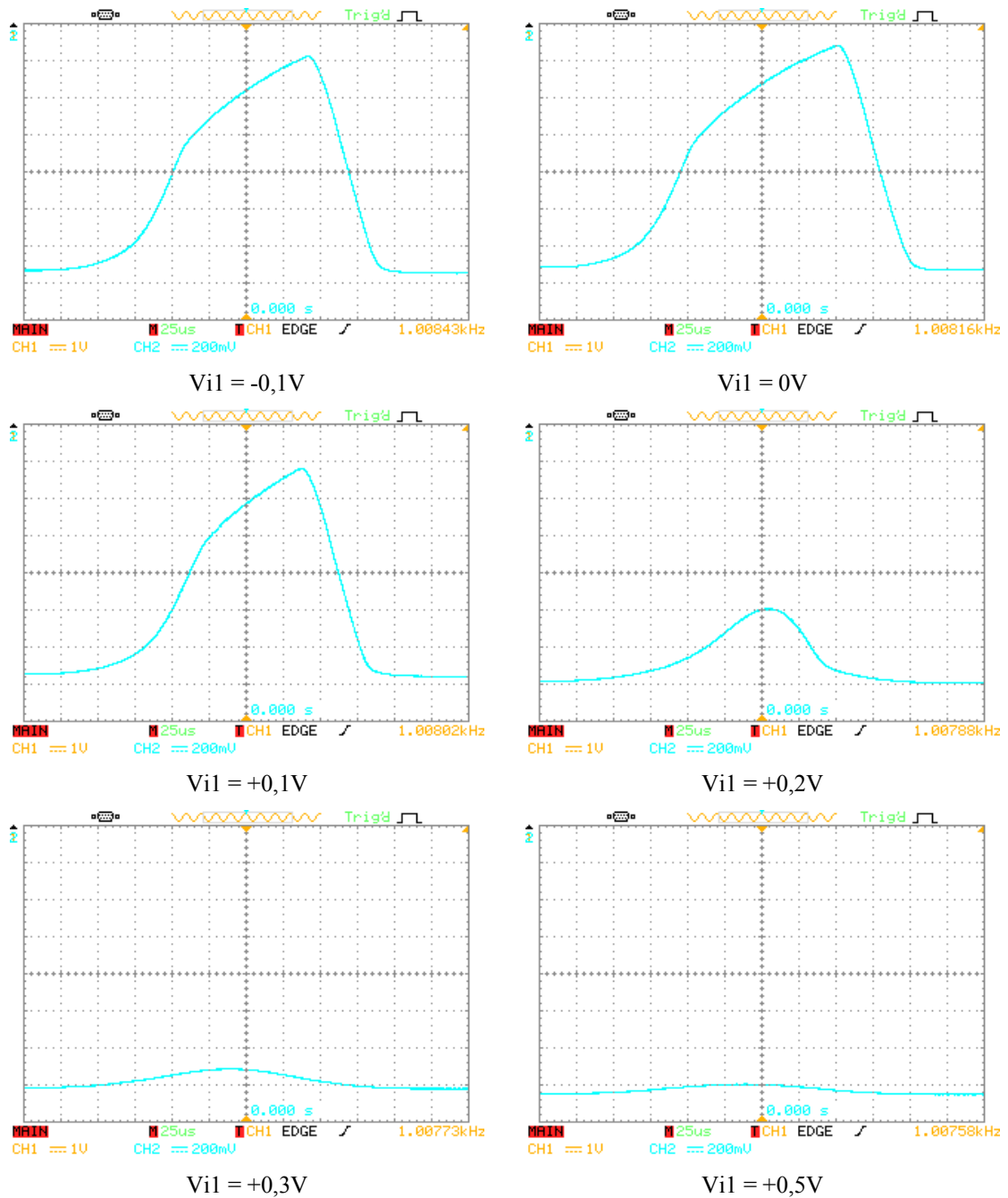


Figura 5.13b: Resultados obtidos para o circuito RBF1 bidimensional para V_{i2} triangular e variações na tensão V_{i1} com $R_L = \infty$ (aberto) (continuação).

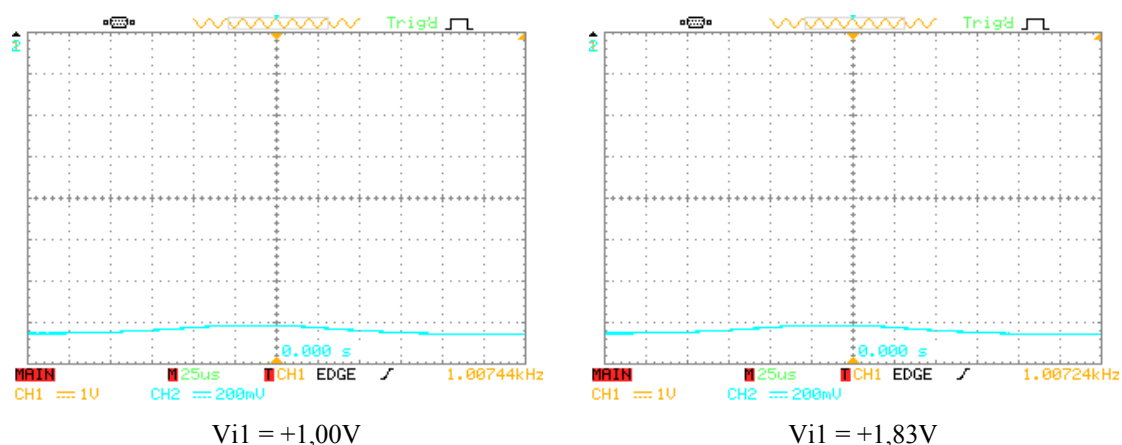


Figura 5.13c: Resultados obtidos para o circuito RBF1 bidimensional para Vi2 triangular e variações na tensão Vi1 com $R_L = \infty$ (aberto) (continuação).

5.3 Circuito RBF₂

O circuito RBF₂ CMOS unidimensional foi implementado em *protoboard* com componentes comerciais ALD1106 e ALD1107 (*MOSFET matched-pair arrays*). Fotos da montagem experimental são apresentadas nas Figuras 5.14 e 5.15. Na entrada do circuito foi aplicada uma onda dente de serra e a tensão na saída foi medida com o osciloscópio. Na Figura 5.16 é mostrada uma foto do sinal de saída obtido no osciloscópio para o sinal dente de serra na entrada.

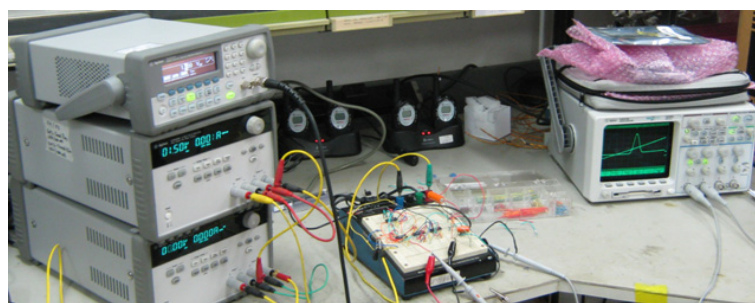


Figura 5.14: Montagem experimental para o circuito CMOS RBF₂.

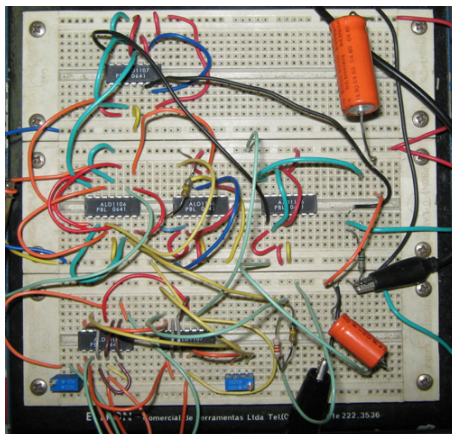


Figura 5.15: Montagem experimental do circuito CMOS RBF₂ completo.

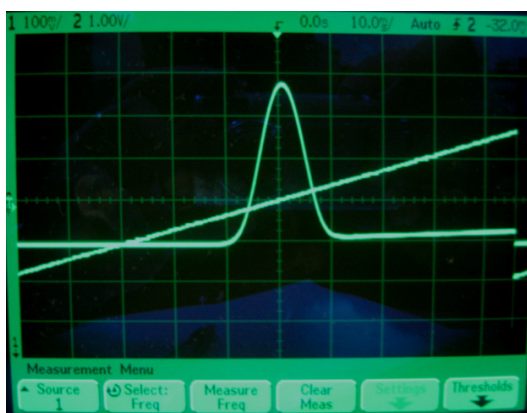
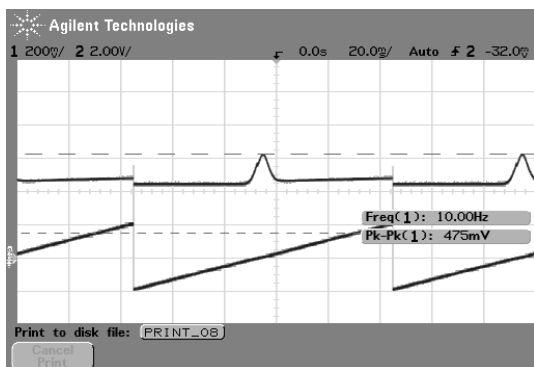


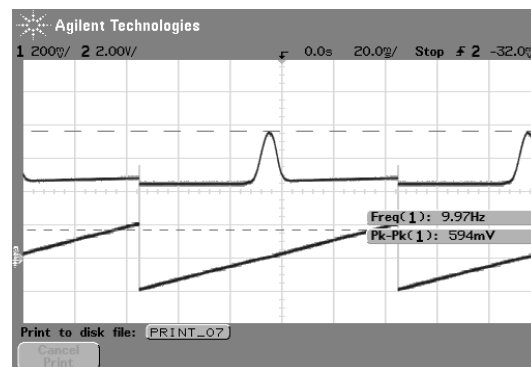
Figura 5.16: Resultados experimentais obtidos para o circuito CMOS RBF₂.

5.3.1 Variação de parâmetros da RBF₂

Na Figura 5.17 são apresentados os resultados obtidos para variações nas tensões de polarização V_{b1} e V_{b2} do circuito RBF₂ unidimensional básico.

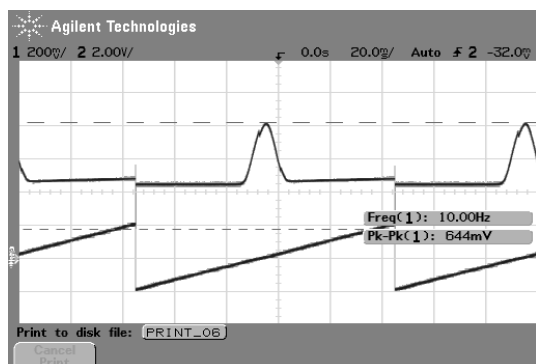


$V_{b1}=50\text{mV}$ e $V_{b2}=-50\text{mV}$.

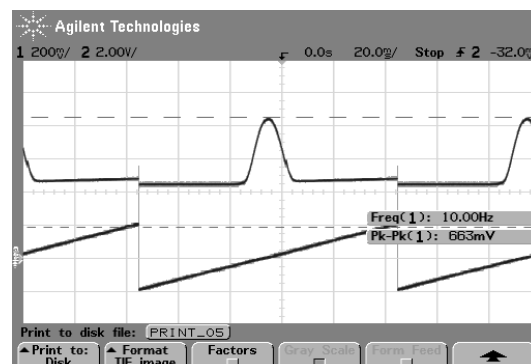


$V_{b1}=100\text{mV}$ e $V_{b2}=-100\text{mV}$.

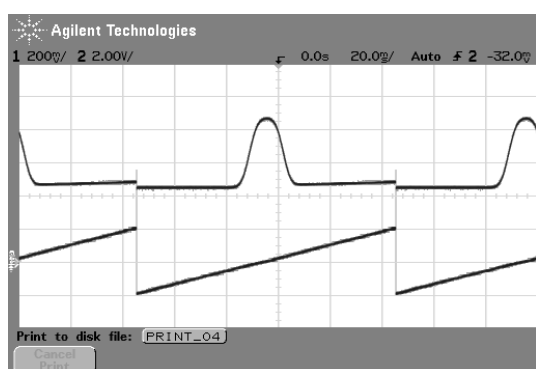
Figura 5.17a: Resultados obtidos para variações nas tensões de polarização V_{b1} e V_{b2} do circuito CMOS RBF₂ unidimensional básico.



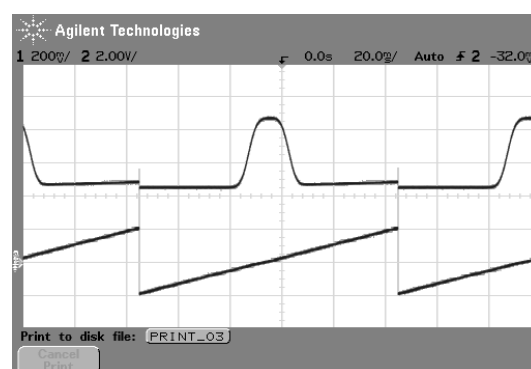
$V_{b1}=150\text{mV}$ e $V_{b2}=-150\text{mV}$.



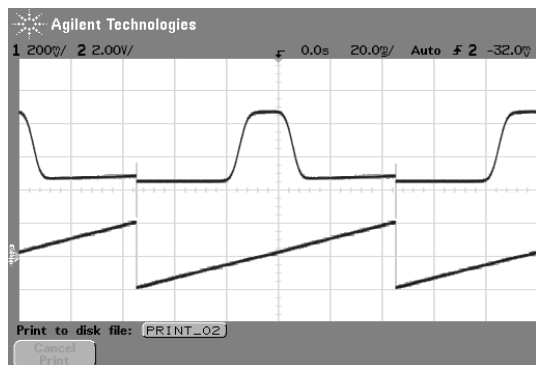
$V_{b1}=200\text{mV}$ e $V_{b2}=-200\text{mV}$.



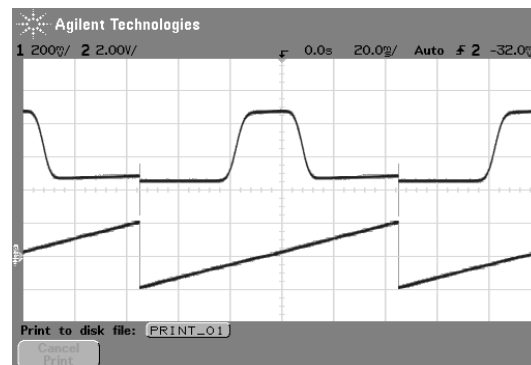
$V_{b1}=250\text{mV}$ e $V_{b2}=-250\text{mV}$.



$V_{b1}=300\text{mV}$ e $V_{b2}=-300\text{mV}$.



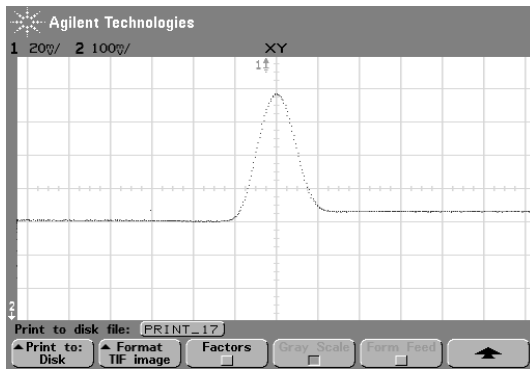
$V_{b1}=400\text{mV}$ e $V_{b2}=-400\text{mV}$.



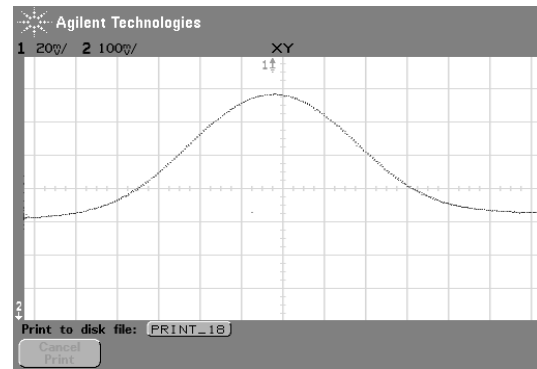
$V_{b1}=500\text{mV}$ e $V_{b2}=-500\text{mV}$.

Figura 5.17b: Resultados obtidos para variações nas tensões de polarização V_{b1} e V_{b2} do circuito CMOS RBF_2 unidimensional básico.

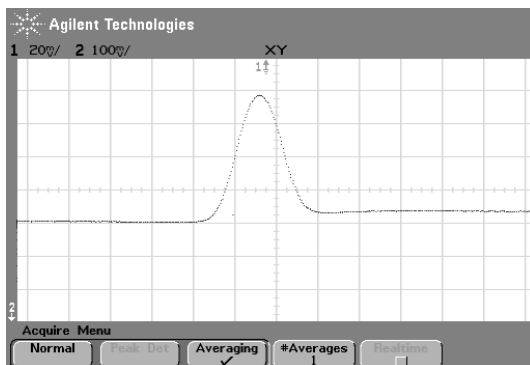
Na Figura 5.18 são apresentados os resultados obtidos para variações na corrente de polarização I_{b1} do circuito RBF_2 unidimensional completo. As condições de polarização foram $V_{b1}=150\text{ mV}$, $V_{b2}=-150\text{ mV}$, $R_{L2}=100\text{ k}\Omega$ (substituindo o conversor I/V) e $V_c = 0\text{ mV}$.



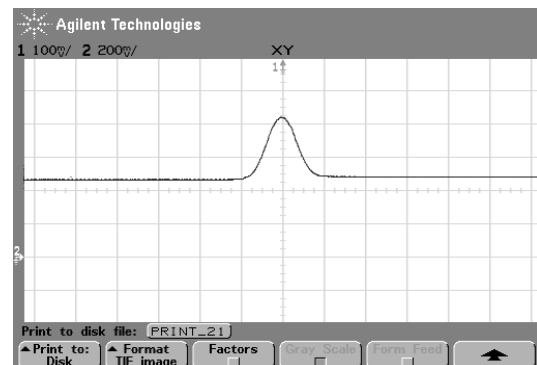
$I_{b1}=1,46 \text{ mA}$. $V_c=129,8 \text{ mV}$.



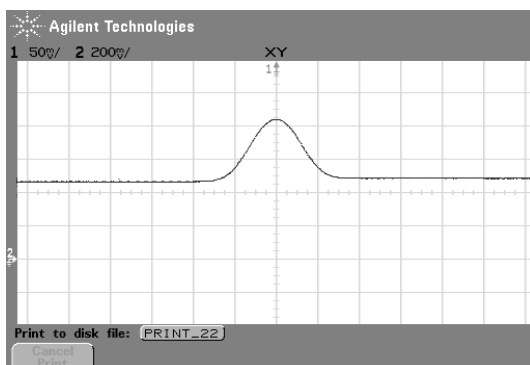
$I_{b1}=4,46 \mu\text{A}$.



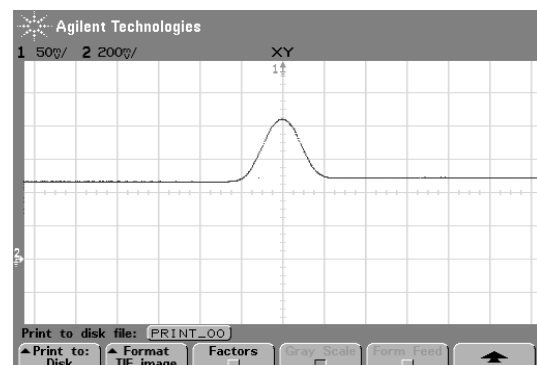
$I_{b1}=53,6 \mu\text{A}$.



$I_{b1}=5,274 \mu\text{A}$.

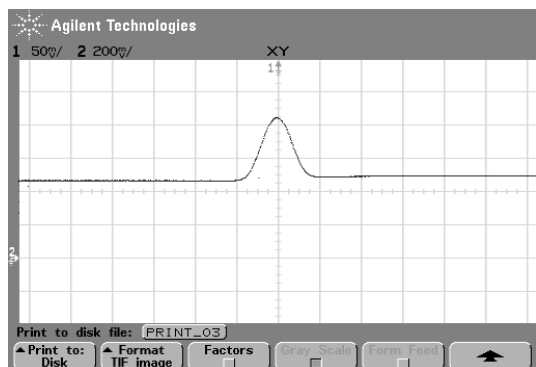


$I_{b1}=6,70 \mu\text{A}$.

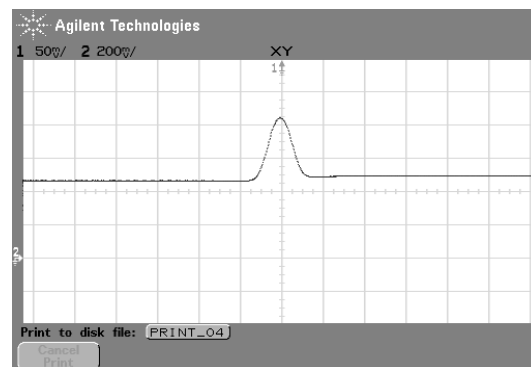


$I_{b1}=10,65 \mu\text{A}$.

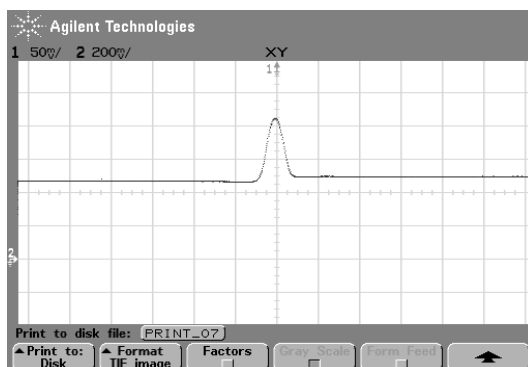
Figura 5.18a: Resultados obtidos para variações na corrente de polarização I_{b1} do circuito CMOS RBF₂ completo.



$I_{b1}=15,84 \mu\text{A}$.



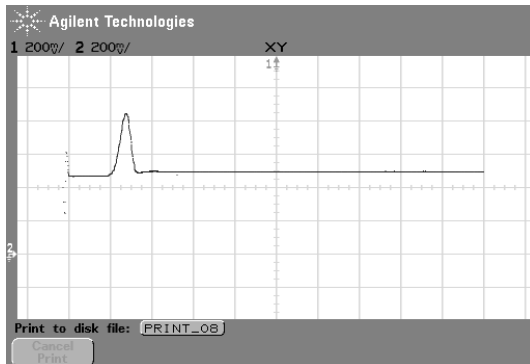
$I_{b1}=26,52 \mu\text{A}$.



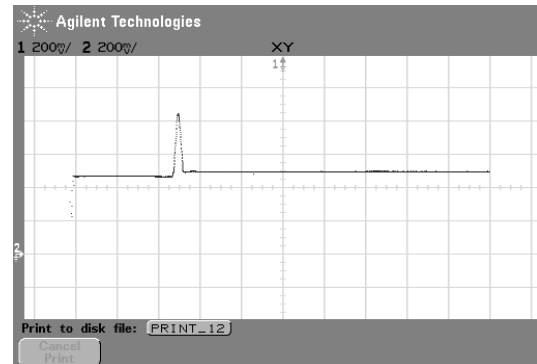
$I_{b1}=86,49 \mu\text{A}$.

Figura 5.18b: Resultados obtidos para variações na corrente de polarização I_{b1} do circuito CMOS RBF_2 completo.

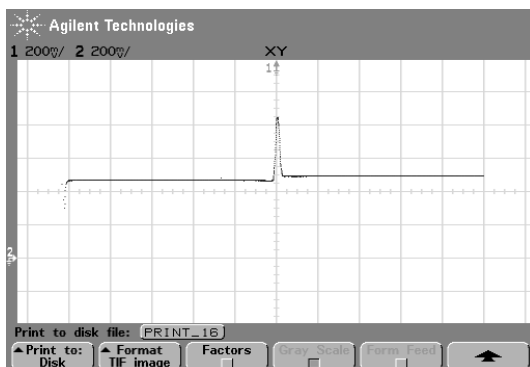
Na Figura 5.19 são apresentados os resultados obtidos para variações na tensão de centro V_c do circuito RBF_2 unidimensional completo. As condições de polarização foram $V_{b1}=150 \text{ mV}$, $V_{b2}=-150 \text{ mV}$, $R_{L2}=100 \text{ k}\Omega$ e $I_{b1}=1,47 \text{ mA}$.



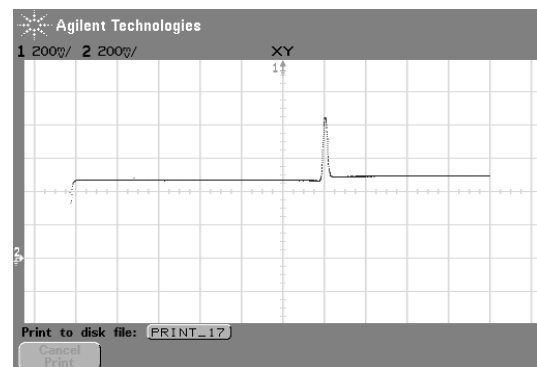
$V_c = -708 \text{ mV}$.



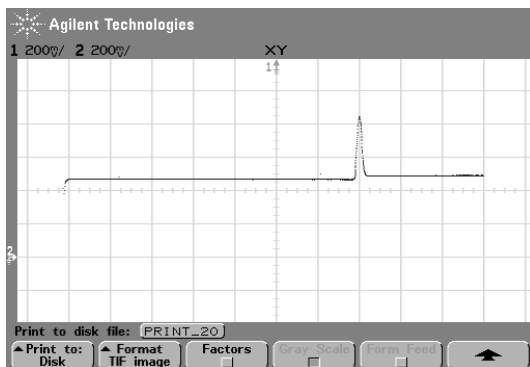
$V_c = -490 \text{ mV}$.



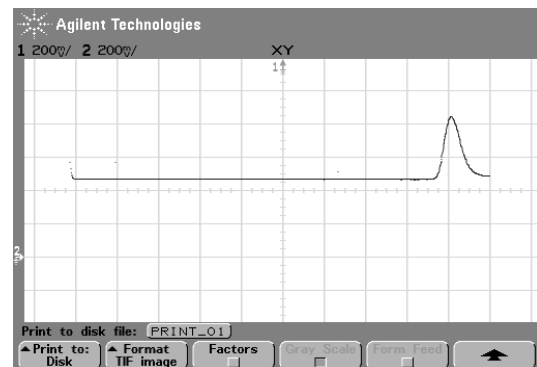
$V_c = 0,01 \text{ mV}$.



$V_c = 188,8 \text{ mV}$.



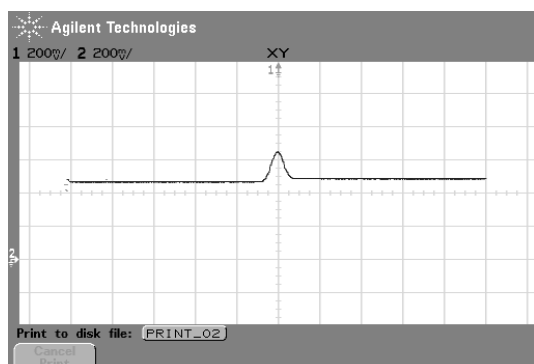
$V_c = 367 \text{ mV}$.



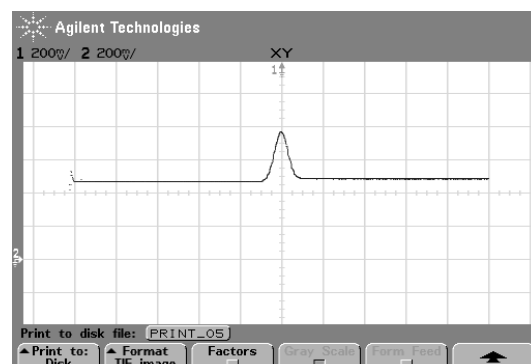
$V_c = 646 \text{ mV}$.

Figura 5.19: Resultados obtidos para variações na tensão de centro V_c do circuito CMOS RBF₂ completo.

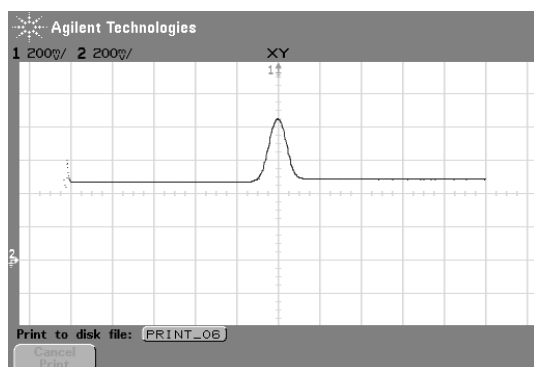
Na Figura 5.20 são apresentados os resultados obtidos para variações nas tensões de polarização V_{b1} e V_{b2} do circuito RBF_2 unidimensional completo. As condições de polarização são $R_{L2}=100\text{ k}\Omega$. $I_{b1}=4,48\text{ }\mu\text{A}$. $V_c=0\text{ mV}$.



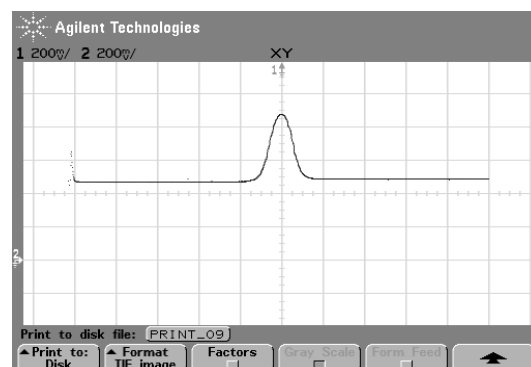
$V_{b1}=50\text{ mV}$ e $V_{b2}=-50\text{ mV}$.



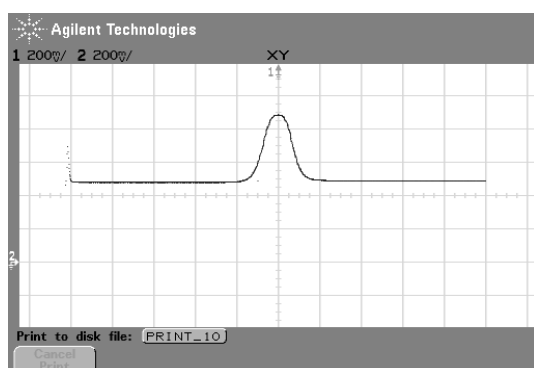
$V_{b1}=100\text{ mV}$ e $V_{b2}=-100\text{ mV}$.



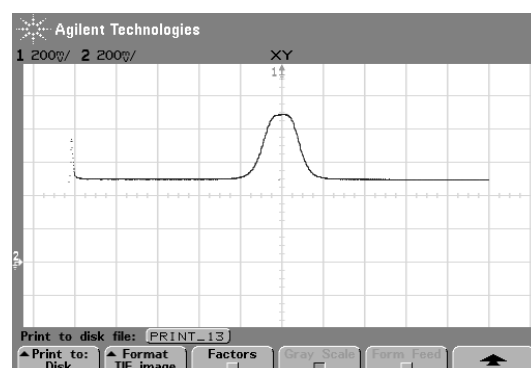
$V_{b1}=150\text{ mV}$ e $V_{b2}=-150\text{ mV}$.



$V_{b1}=200\text{ mV}$ e $V_{b2}=-200\text{ mV}$.



$V_{b1}=250\text{ mV}$ e $V_{b2}=-250\text{ mV}$.



$V_{b1}=300\text{ mV}$ e $V_{b2}=-300\text{ mV}$.

Figura 5.20: Resultados obtidos para variações nas tensões de polarização V_{b1} e V_{b2} do circuito CMOS RBF_2 unidimensional completo.

5.4 Conclusões

Neste capítulo foram apresentados resultados experimentais para os circuitos RBF_1 e RBF_2 propostos para a implementação de funções de base radial univariáveis e multivariáveis. Os resultados de simulação SPICE e os resultados experimentais mostraram uma boa funcionalidade para os circuitos. Foi constatado que o circuito RBF_1 implementado em *protoboard* apresentou um bom funcionamento até uma frequência de aproximadamente 100 kHz para as condições de polarização utilizadas.

Parte III:
Proposta de Implementação de uma Rede de Base
Radial em Tecnologia CMOS

Capítulo 6

Redes de Base Radial em Processo CMOS

Neste capítulo serão apresentados dois projetos de circuitos analógicos em tecnologia CMOS AMS 0.35 μm que permitem a implementação de uma RBFN bidimensional. Esses projetos serão realizados por meio de simulações computacionais utilizando os modelos de transistores fornecidos no *design kit* AMS 0.35 e o pacote de simulação e projeto de circuitos integrados TANNER. Essa rede bidimensional permitirá a avaliação dos circuitos propostos para as funções de base radial. Nos itens seguintes serão apresentados a arquitetura da rede, os blocos básicos utilizados e as simulações efetuadas para verificar a funcionalidade dos circuitos.

6.1 Arquitetura da Rede

A arquitetura proposta para a rede de base radial consiste em duas entradas, algumas funções de base na camada intermediária e apenas uma saída, conforme mostrado na Figura 6.1. O circuito da rede bidimensional baseia-se na ligação em cascata de funções de base radial unidimensionais, conforme mostrado no Capítulo 4. A análise dos circuitos RBF₁ e RBF₂ mostrou que eles permitem o ajuste de todos os parâmetros de forma da função básica, ou seja, a amplitude, o centro e a largura da RBF. Assim sendo, decidiu-se utilizar apenas os circuitos básicos propostos, sem a necessidade de implementar circuitos adicionais para obter essa capacidade de ajuste dos parâmetros. A saída da rede é obtida pela “soma pelo fio” das saídas de cada base, ou seja, simplesmente pela conexão de todas as saídas individuais à resistência de carga na saída. Nesta tese, tratar-se-á apenas do circuito analógico que implementa o núcleo computacional da rede. Futuramente, deverão ser desenvolvidos blocos digitais que permitirão o treinamento da rede de forma *off-line* ou

chip-in-the-loop e o armazenamento dos parâmetros da rede e a sua programação por meio de circuitos registradores.

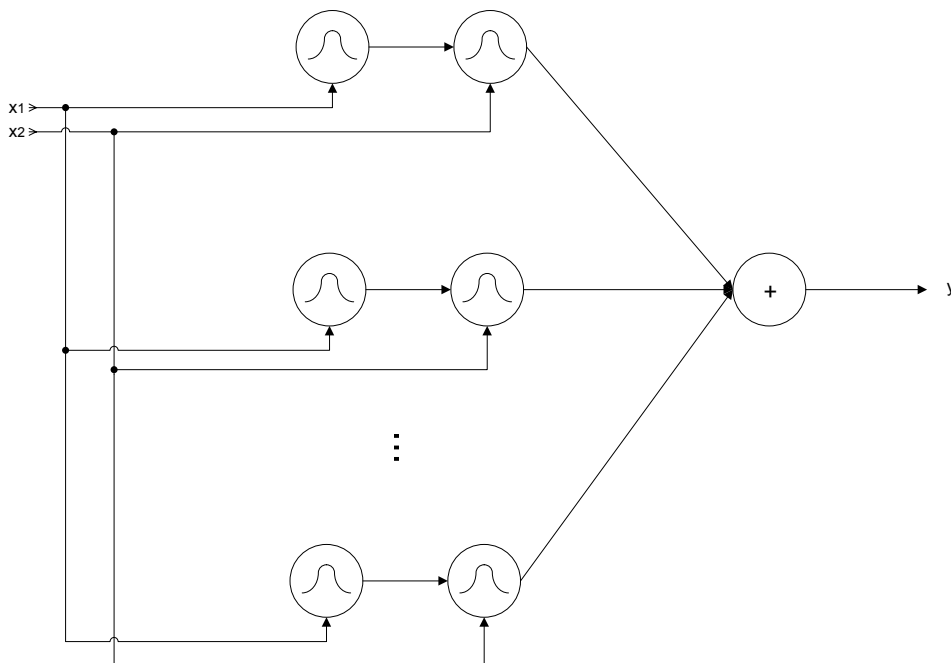


Figura 6.1: Arquitetura da rede bidimensional proposta.

6.2 Circuito RBF₁ básico

O circuito RBF₁ básico é mostrado na Figura 6.2. Esse circuito foi simulado no software TANNER e os resultados são mostrados nas Figuras 6.3 a 6.6.

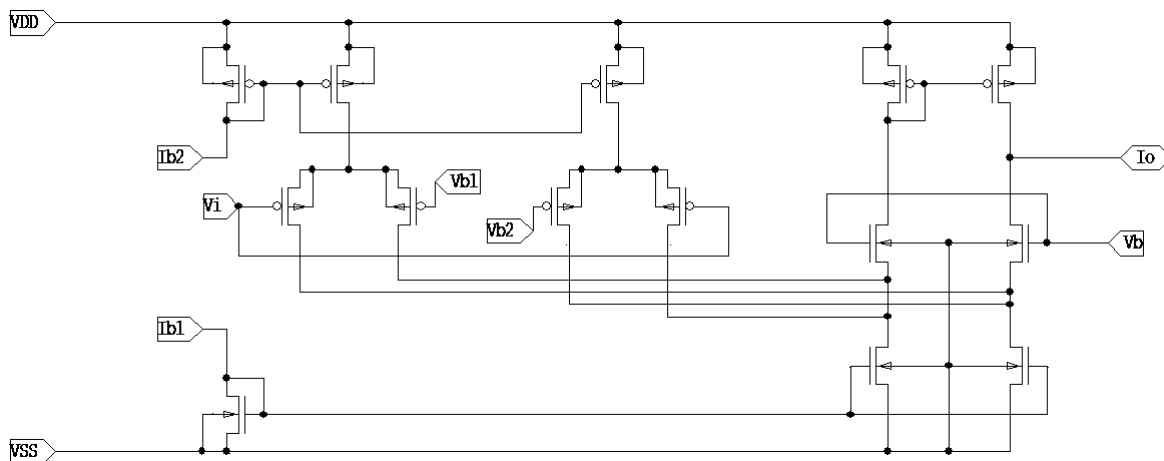


Figura 6.2: Circuito RBF1 básico.

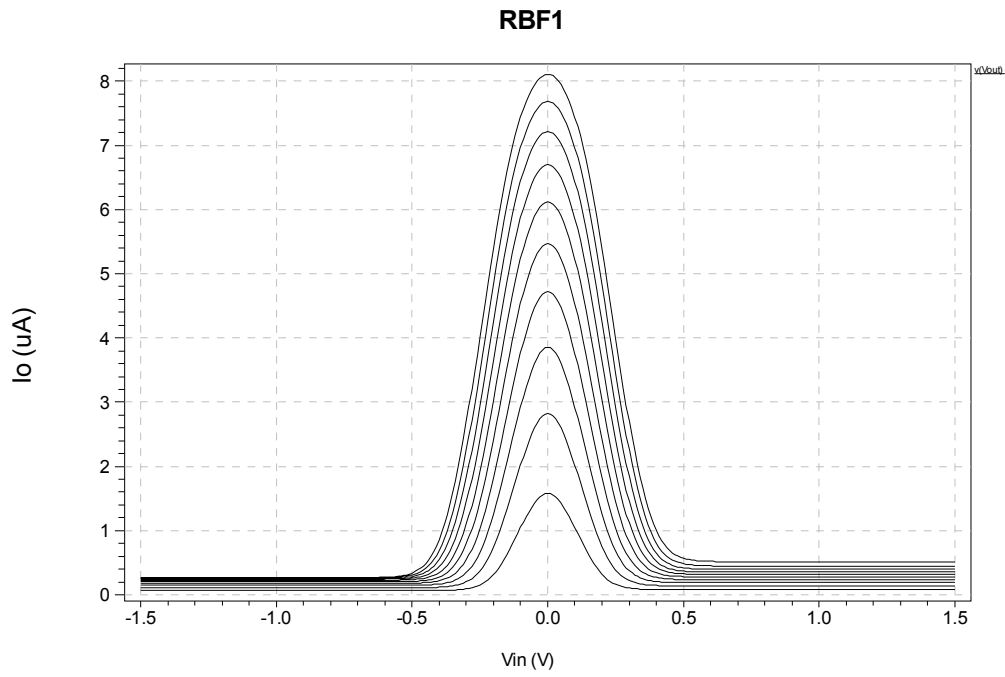


Figura 6.3: I_{b2} variando de $1\mu\text{A}$ a $10\mu\text{A}$ passo= $1\mu\text{A}$. $R_L=1\Omega$.
 $V_{b1}=+100\text{mV}$ e $V_{b2}=-100\text{mV}$.

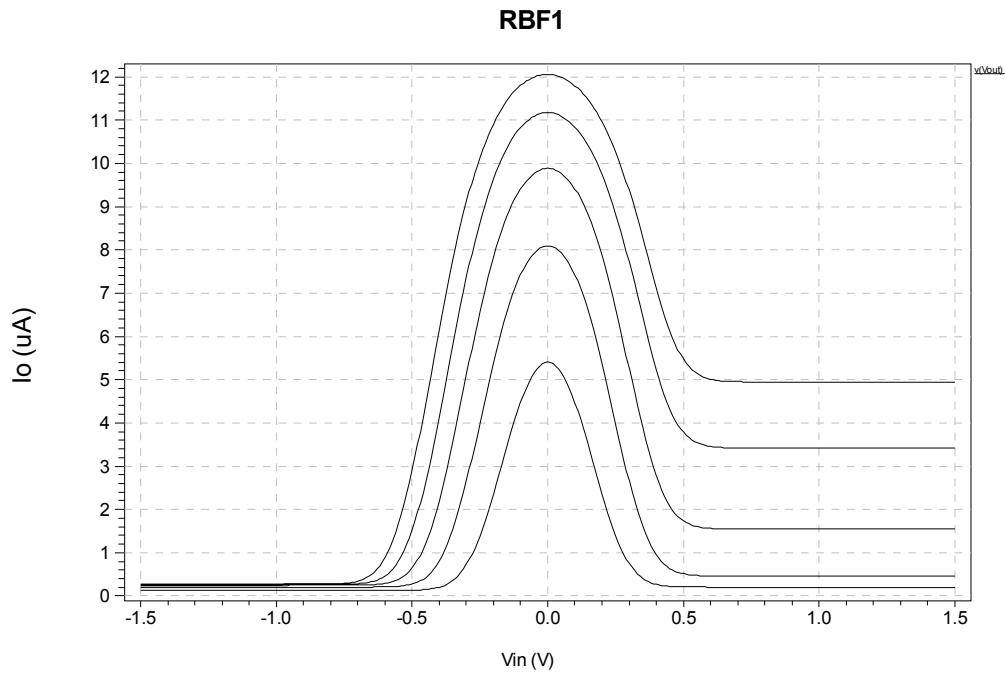


Figura 6.4: I_{b2} variando de $5\mu\text{A}$ a $25\mu\text{A}$ passo= $5\mu\text{A}$. $R_L=1\Omega$.
 $V_{b1}=+100\text{mV}$ e $V_{b2}=-100\text{mV}$.

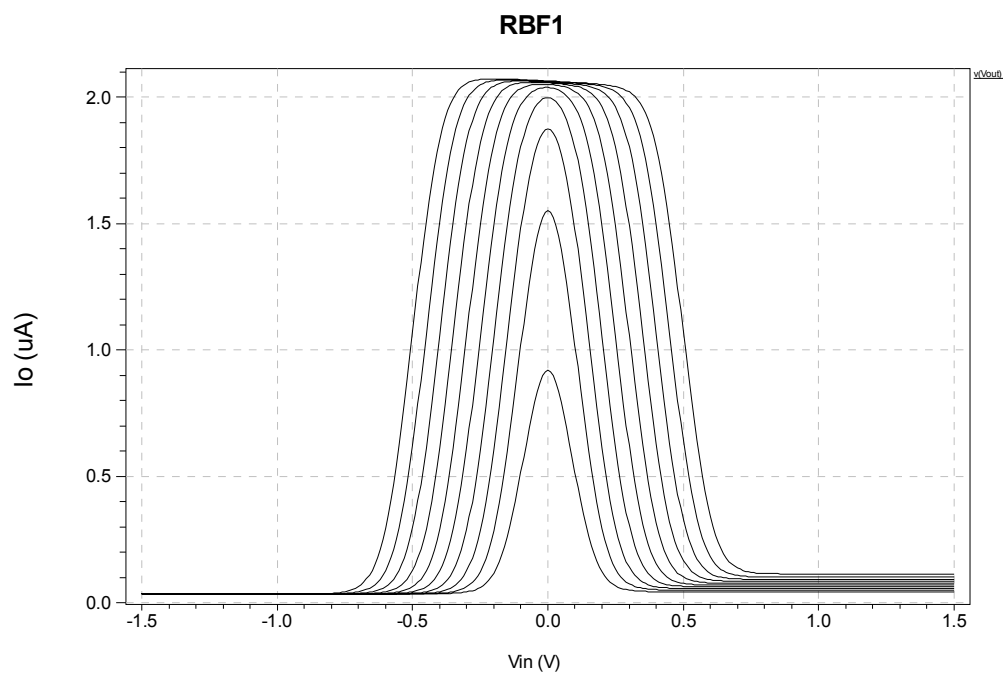


Figura 6.5: Variação de $(V_{b1}-V_{b2})$ de 100 mV a 1 V. $I_{b2}=1\mu\text{A}$.

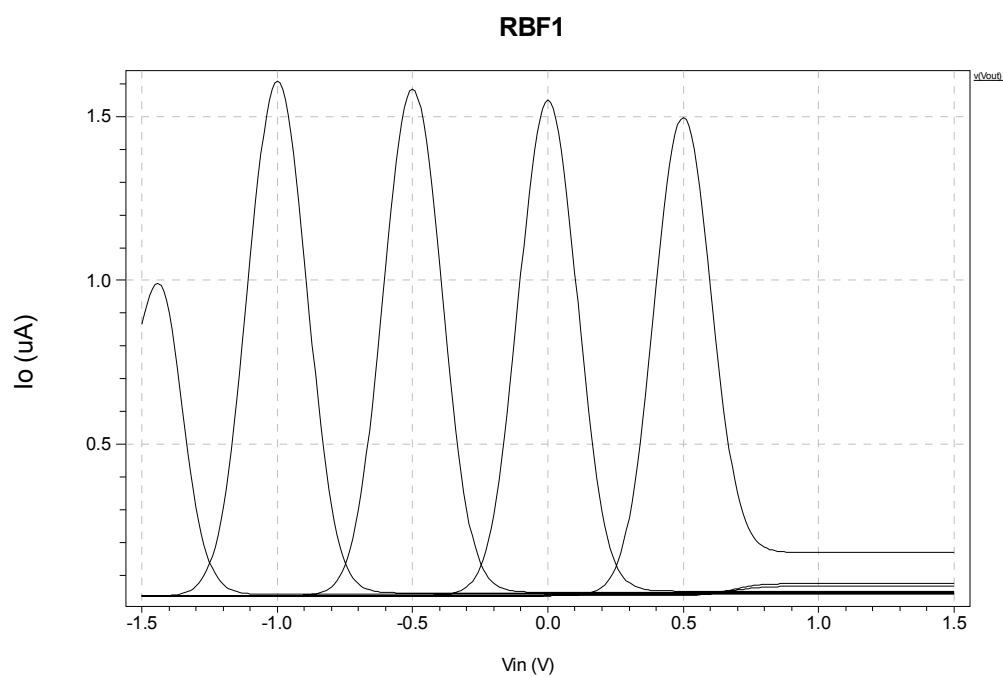


Figura 6.6: Variação de centro da RBF_1 básica AMS035.
 $I_{b2}=1\mu\text{A}$. $(V_{b1}-V_{b2})$ de 100 mV.

6.3 Análise do circuito RBF₁ com relação a variações de processo e temperatura

Foram realizadas uma série de simulações no intuito de avaliar o comportamento do circuito RBF₁ com relação a variações de processo, de temperatura e modelos de pior caso. A configuração testada é apresentada na Figura 6.7.

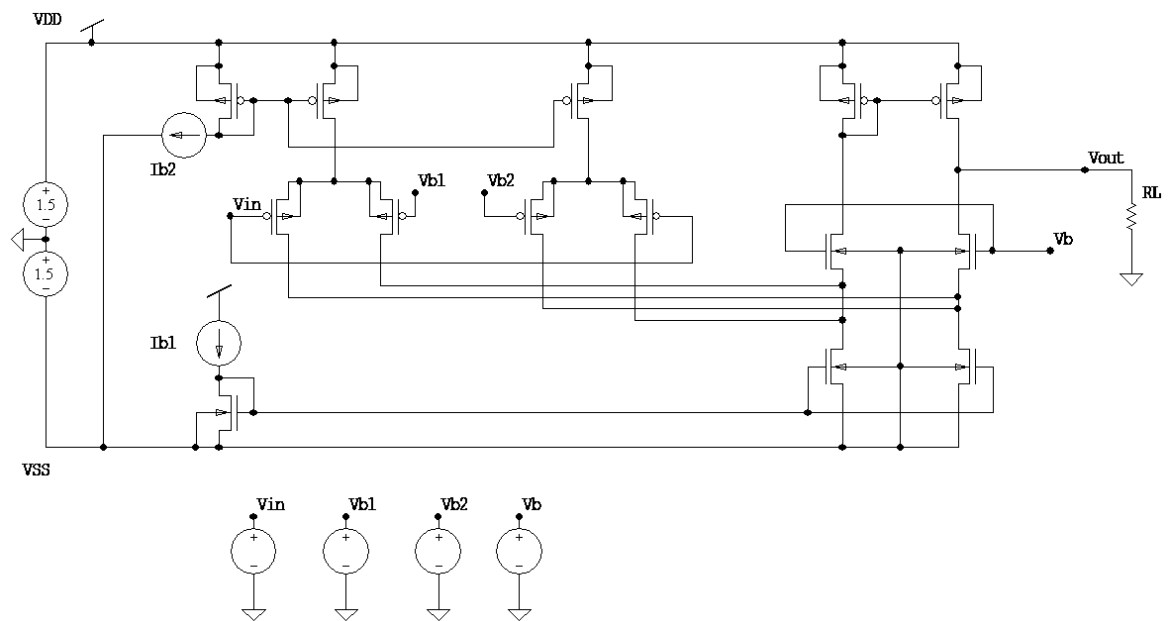


Figura 6.7: Configuração utilizada nas análises de temperatura, Monte Carlo e pior caso.

6.3.1 Análise de temperatura

A seguir serão apresentados resultados obtidos por meio de simulação para valores de temperatura variando entre -50 e +125 °C a fim de analisar o efeito da temperatura no sinal de saída da RBF₁.

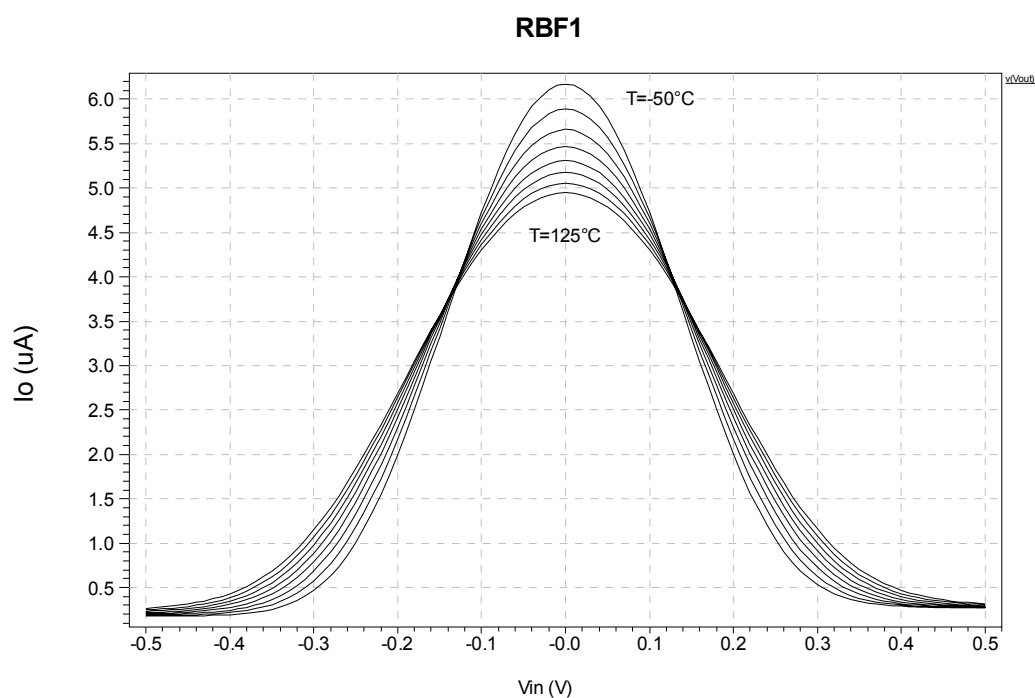


Figura 6.8: Saída de RBF₁ obtida para temperatura variando de -50 a +125 °C.

6.3.2 Análise de Pior Caso

A seguir serão apresentados resultados obtidos por meio de simulação de modelos de pior caso a fim de analisar os efeitos que variações de processo poderão provocar no sinal de saída da RBF₁. Os modelos utilizados foram: típico (tm), pior caso de potência (wp) e pior caso de velocidade (ws).

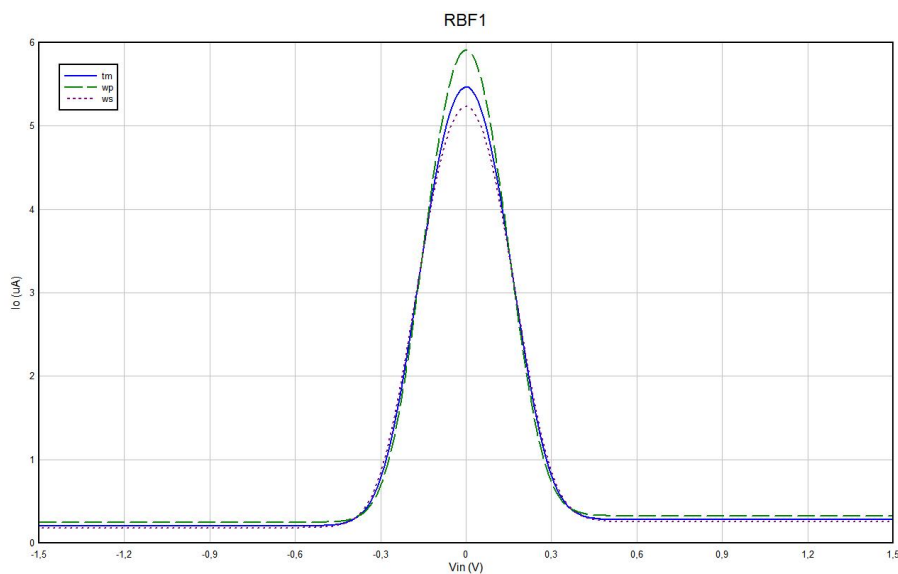


Figura 6.9: RBF₁ - Análise de pior caso.

6.3.3 Análise Monte Carlo

A seguir serão apresentados resultados obtidos por meio de simulação de Monte Carlo, a fim de analisar os efeitos que eventuais descasamentos entre componentes e variações aleatórias de processo poderão provocar no sinal de saída da RBF₁. Foram realizados estudos considerando distribuições de probabilidade uniformes e variações máximas de $\pm 1\%$ e $\pm 5\%$ para os parâmetros de projeto W e L, e também para os parâmetros de processo VTH0, TOX e U0.

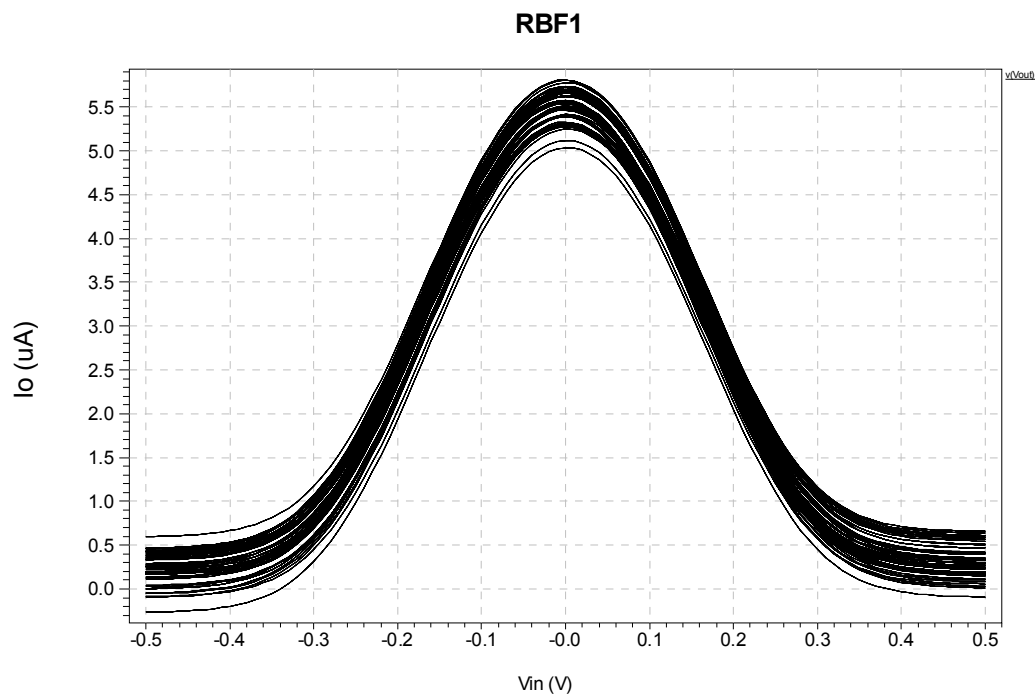


Figura 6.10: Análise Monte Carlo com variação de $\pm 1\%$ em L e $\pm 1\%$ em W.

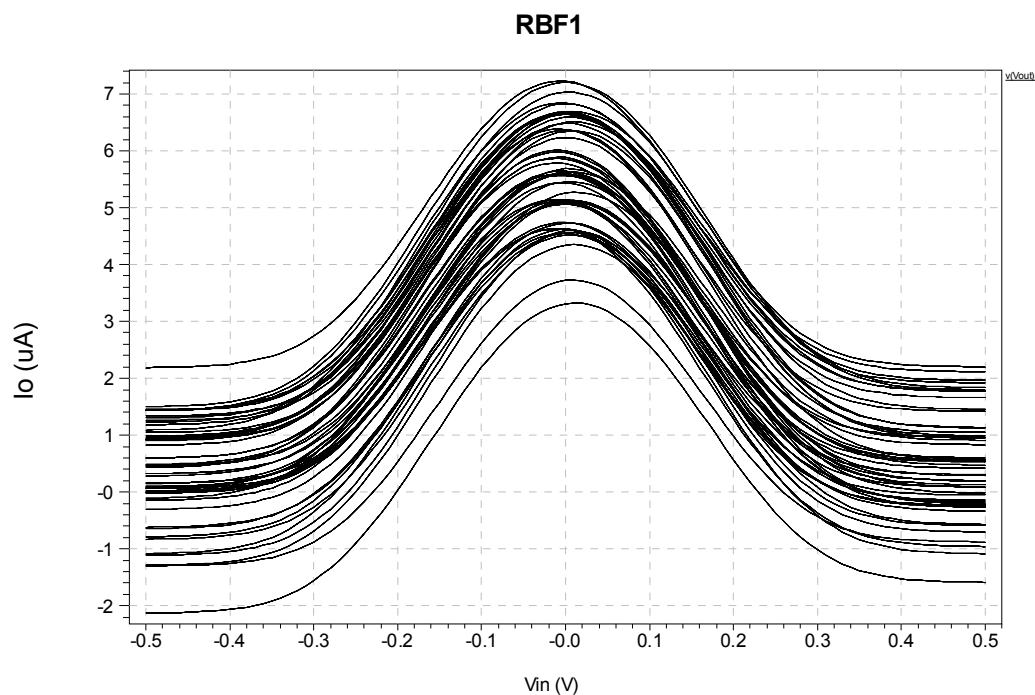


Figura 6.11: Análise Monte Carlo com variação de $\pm 5\%$ em L e $\pm 5\%$ em W.

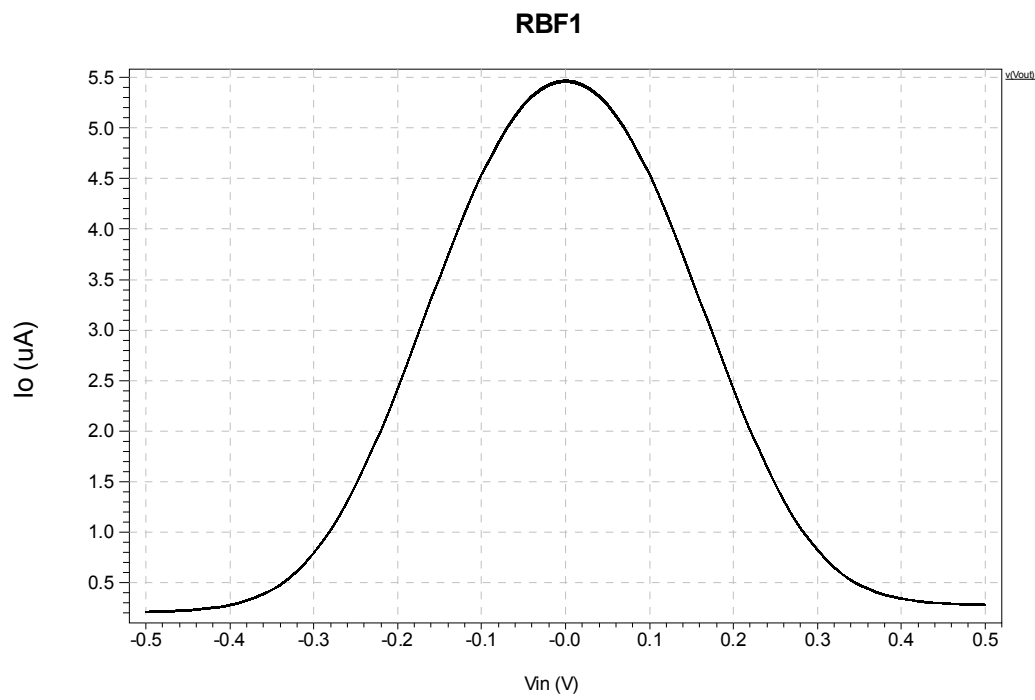


Figura 6.12: Análise Monte Carlo com variação de $\pm 1\%$ em V_{TH0} .

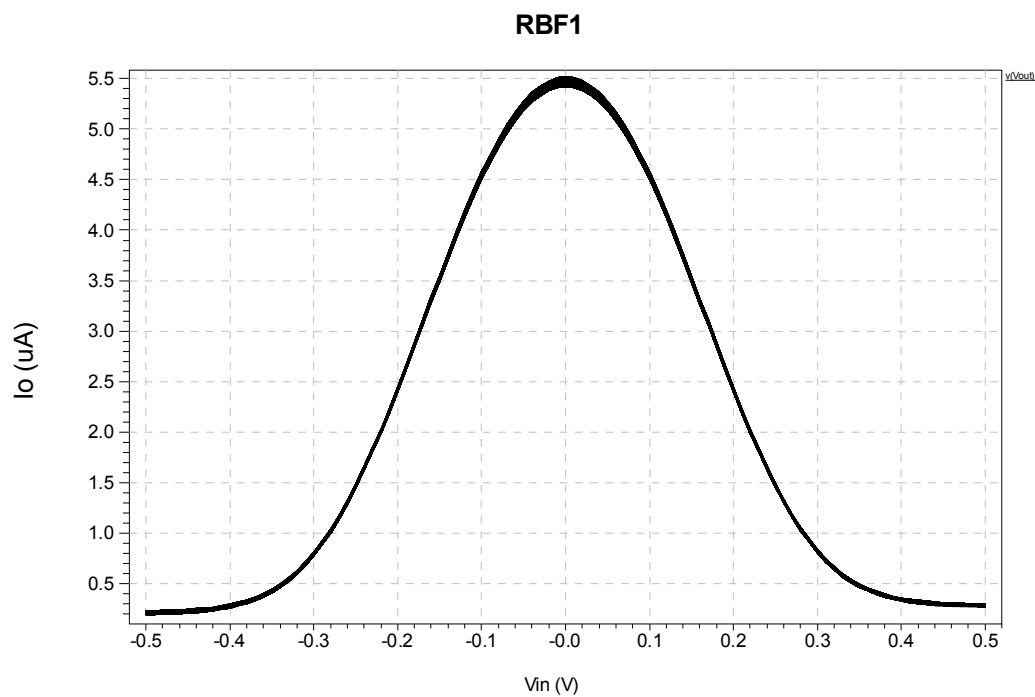


Figura 6.13: Análise Monte Carlo com variação de $\pm 5\%$ em V_{TH0} .

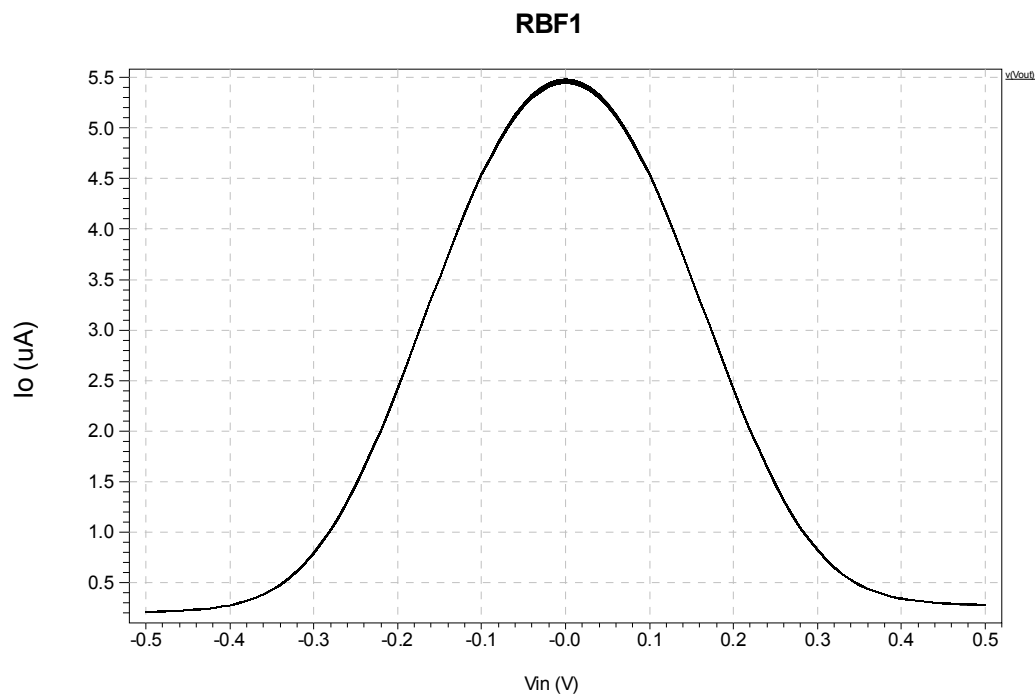


Figura 6.14: Análise Monte Carlo com variação de $\pm 1\%$ em U_0 .

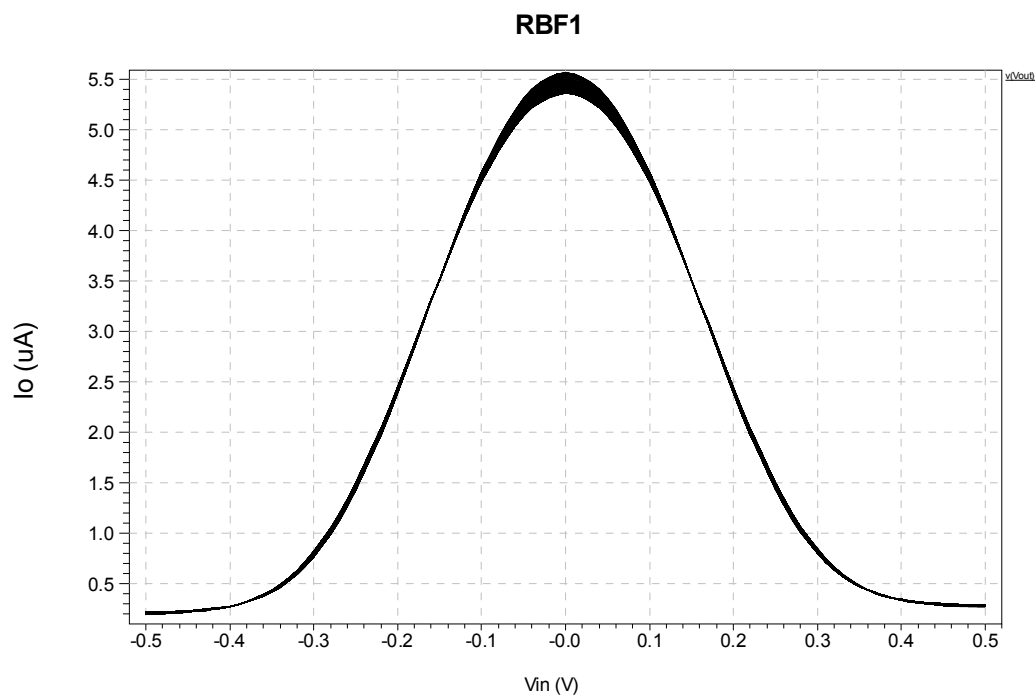


Figura 6.15: Análise Monte Carlo com variação de $\pm 5\%$ em U_0 .

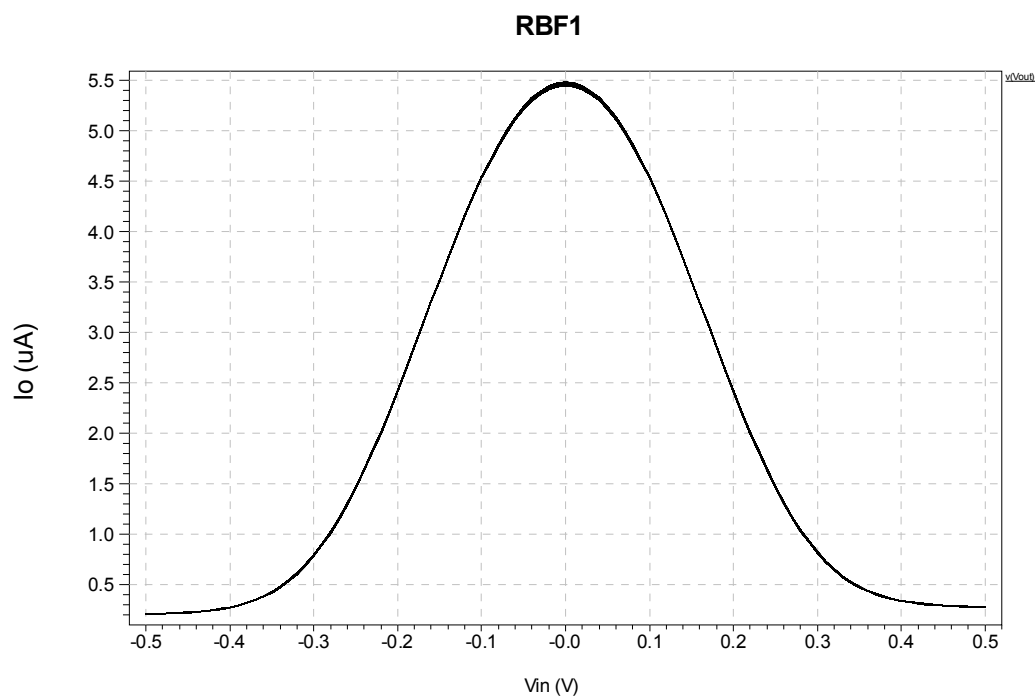


Figura 6.16: Análise Monte Carlo com variação de $\pm 1\%$ em TOX.

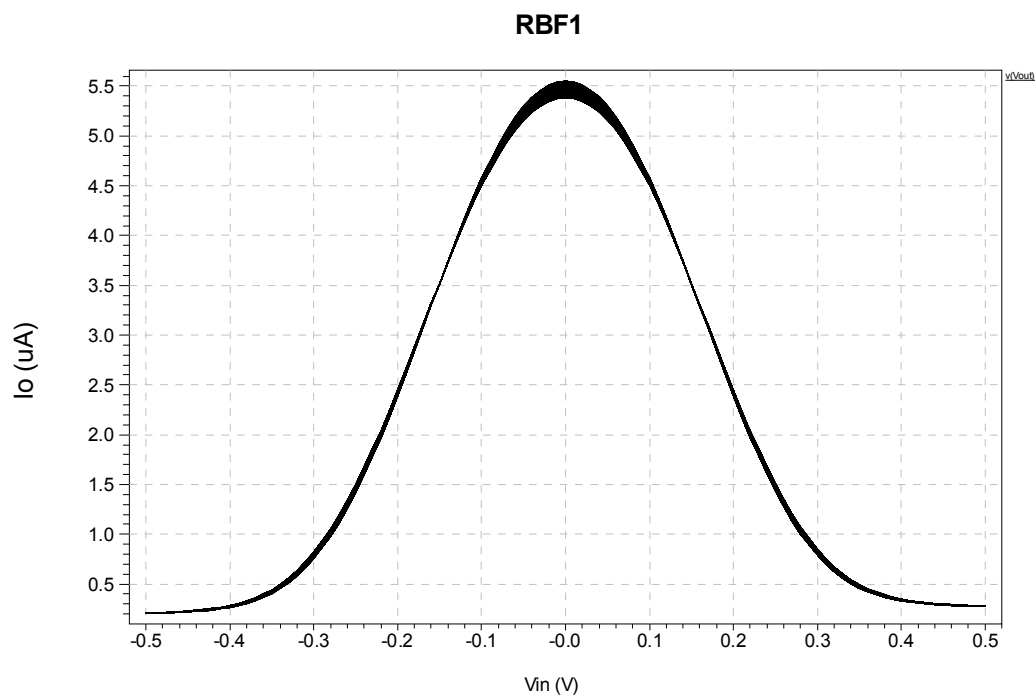


Figura 6.17: Análise Monte Carlo com variação de $\pm 5\%$ em TOX.

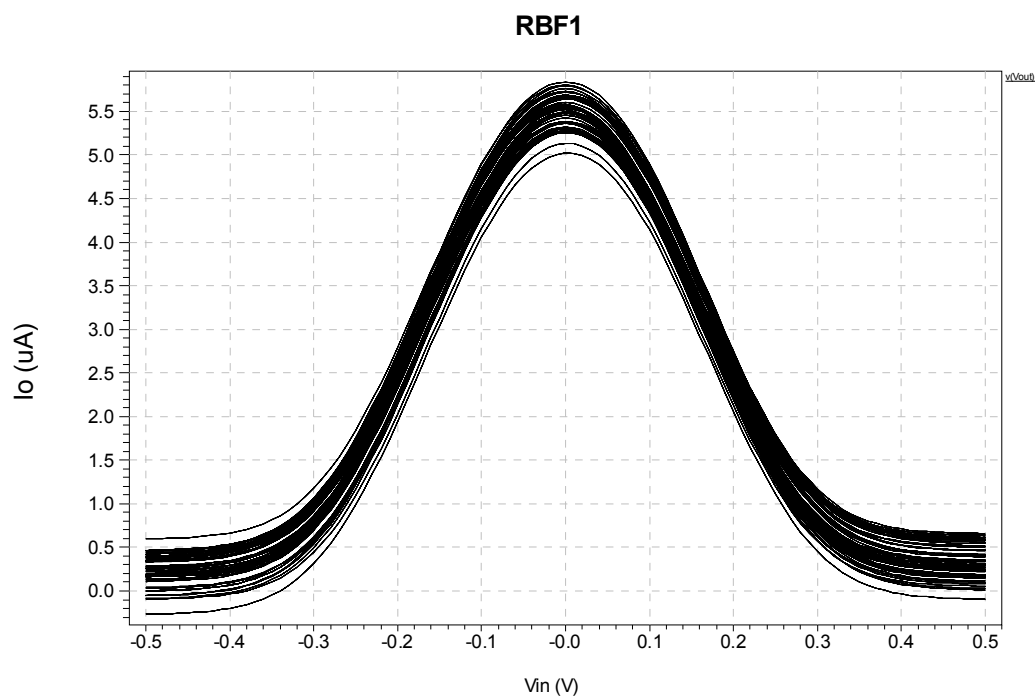


Figura 6.18: Análise Monte Carlo com variação combinada de V_{TH0} , U_0 , TOX , W e L em $\pm 1\%$.

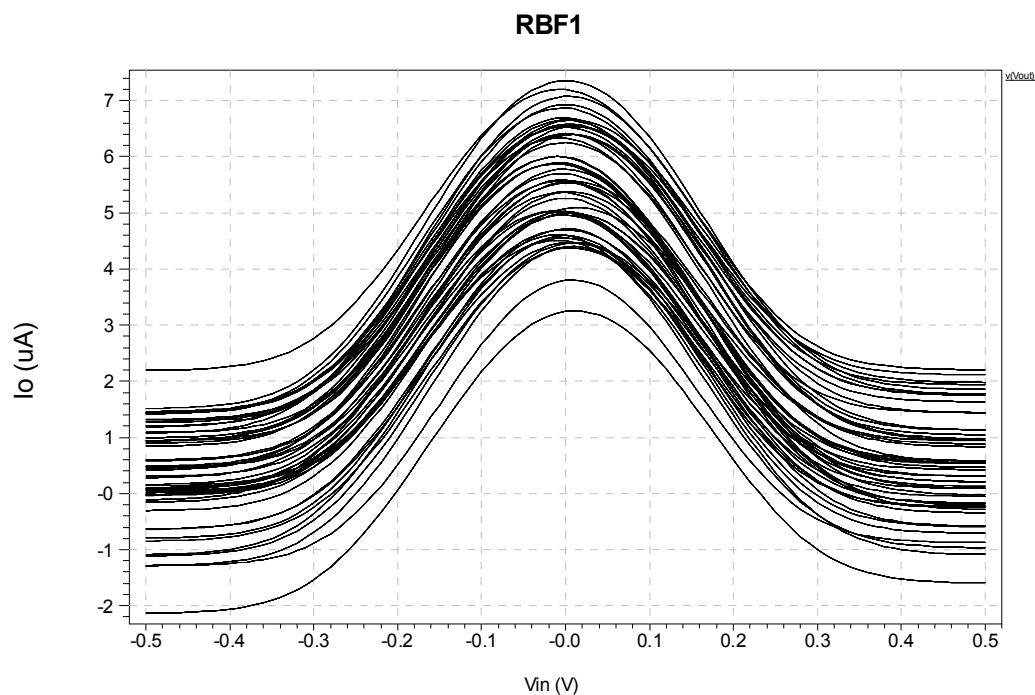


Figura 6.19: Análise Monte Carlo com variação combinada de V_{TH0} , U_0 , TOX , W e L em $\pm 5\%$.

Os resultados obtidos com a análise Monte Carlo foram compilados na Tabela 6.1, que apresenta a variação ocorrida na amplitude da RBF em função da variação dos parâmetros de processo e de projeto.

Tabela 6.1: Variação no máximo da RBF_1 em função da variação de $\pm 1\%$ e $\pm 5\%$ dos parâmetros V_{TH0} , U_0 , TOX , W e L .

Parâmetro	Variação do parâmetro	
	$\pm 1\%$	$\pm 5\%$
V_{TH0}	20,78 nA	113,5 nA
U_0	34,11 nA	204,66 nA
TOX	33,90 nA	195,67 nA
W e L	785 nA	3,91 μ A
Todos	801,83 nA	4,08 μ A

6.4 Circuito RBF_2 básico

O circuito RBF_2 básico é mostrado na Figura 6.20. A configuração de polarização do circuito é mostrada na Figura 6.21. Esse circuito foi simulado no software TANNER e os resultados são mostrados nas Figuras 6.22 a 6.25.

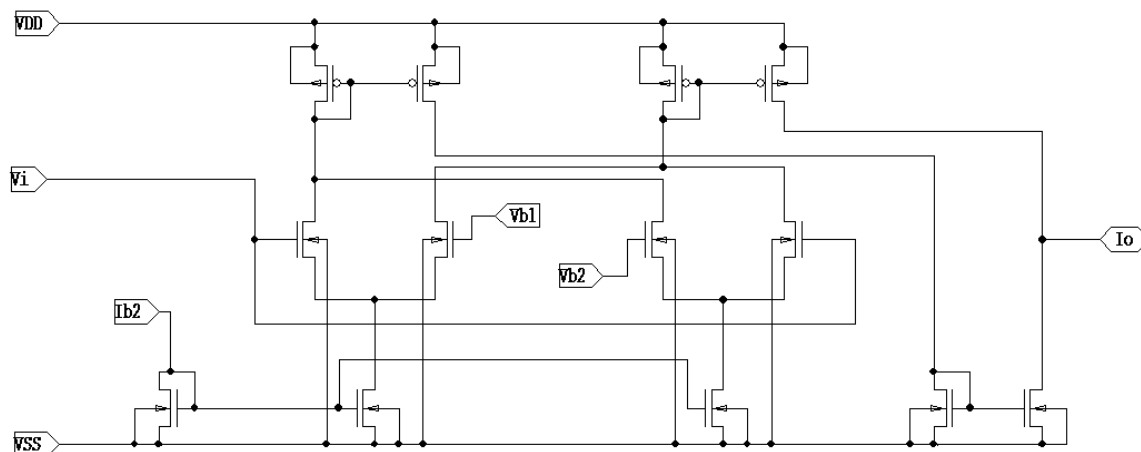


Figura 6.20: Circuito RBF_2 básico.

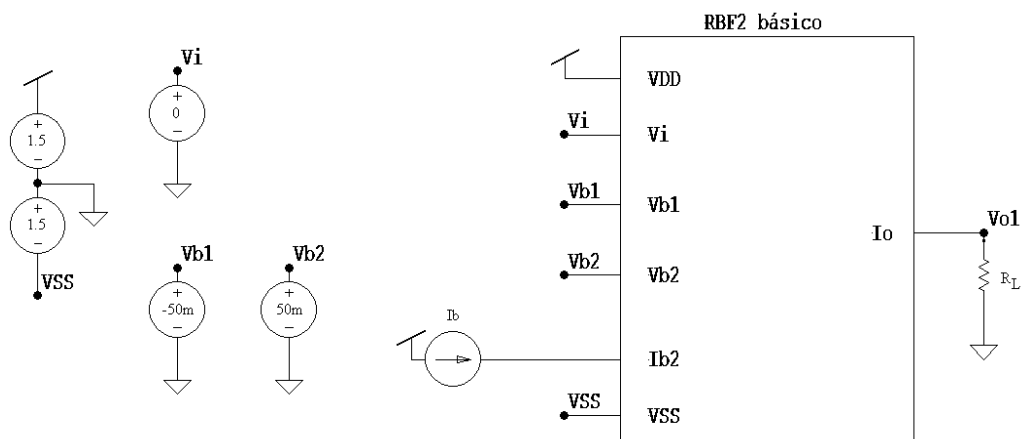


Figura 6.21: *Test bench* para o Circuito RBF₂ básico.

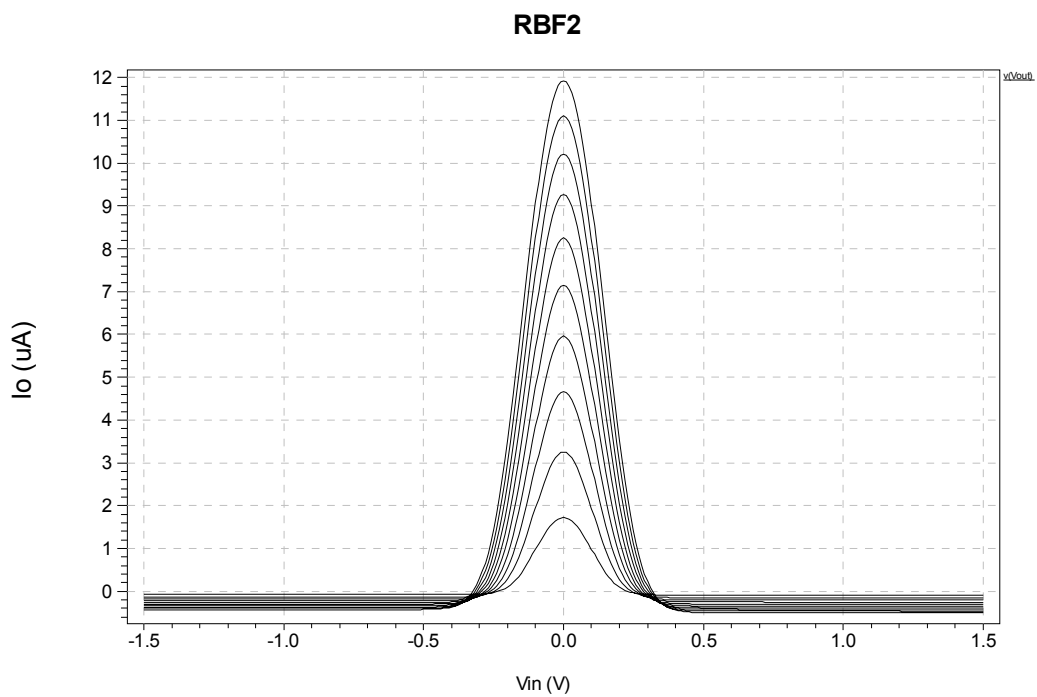


Figura 6.22: I_{b2} variando de 1 μA a 10 μA passo=1 μA . $R_L=1\Omega$.

$V_{b1}=+100\text{mV}$ e $V_{b2}=-100\text{mV}$.

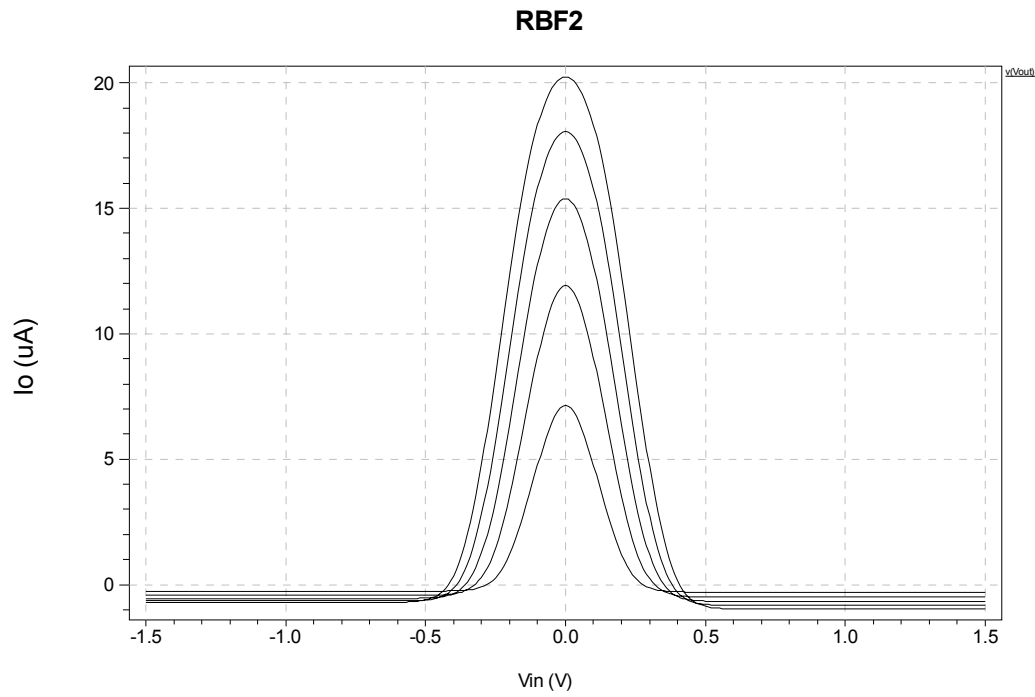


Figura 6.23: I_{b2} variando de $5\mu A$ a $50\mu A$ passo= $5\mu A$. $R_L=1\Omega$.
 $V_{b1}=+100$ mV e $V_{b2}=-100$ mV.

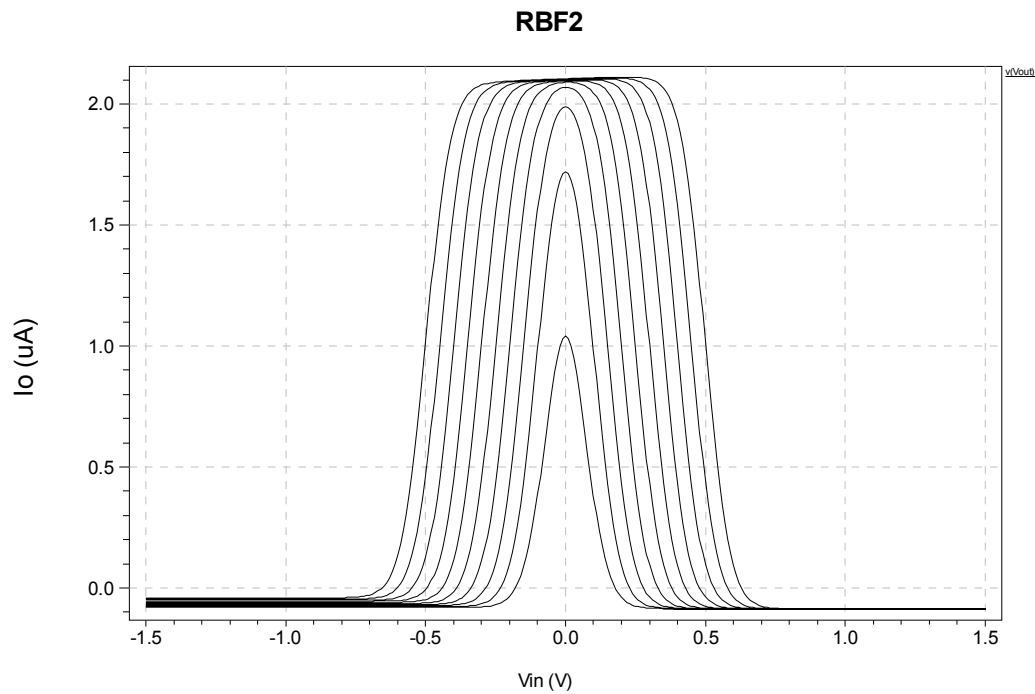


Figura 6.24: Variação de $(V_{b1}-V_{b2})$ de 100 mV a 1 V.

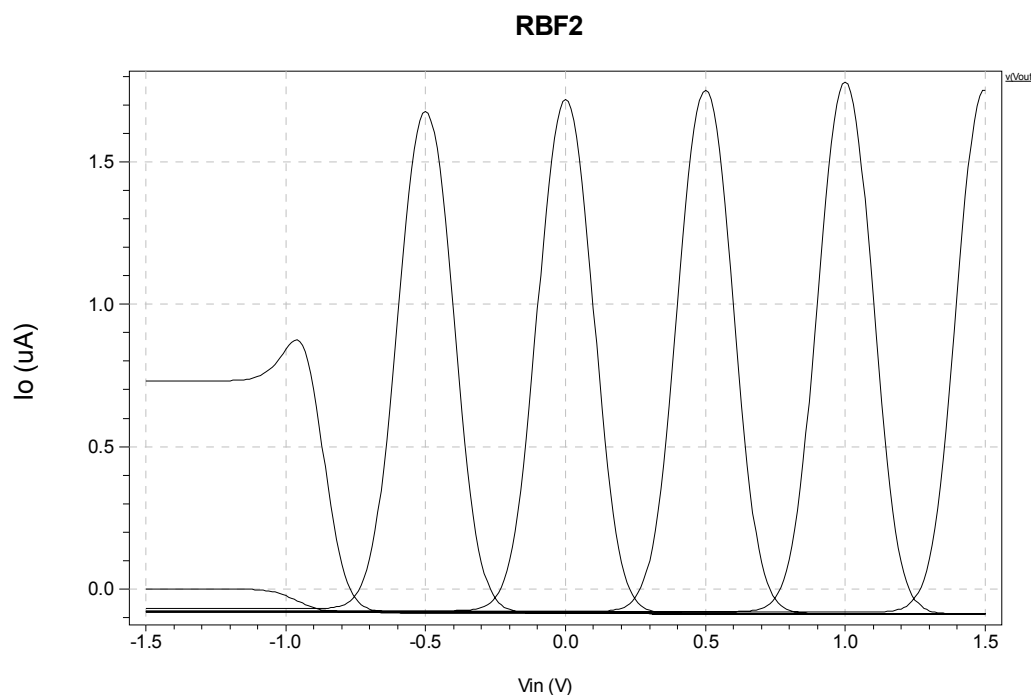


Figura 6.25: Variação de centro da RBF₂ básica AMS035.

6.5 Análise do circuito RBF₁ com relação a variações de processo e temperatura

As mesmas simulações realizadas no intuito de avaliar o comportamento do circuito RBF₁ com relação a variações de processo, de temperatura e modelos de pior caso foram repetidas para o circuito RBF₂.

6.5.1 Análise de temperatura

A seguir serão apresentados resultados obtidos por meio de simulação para valores de temperatura variando entre -50 e +125 °C a fim de analisar o efeito da temperatura no sinal de saída da RBF₂.

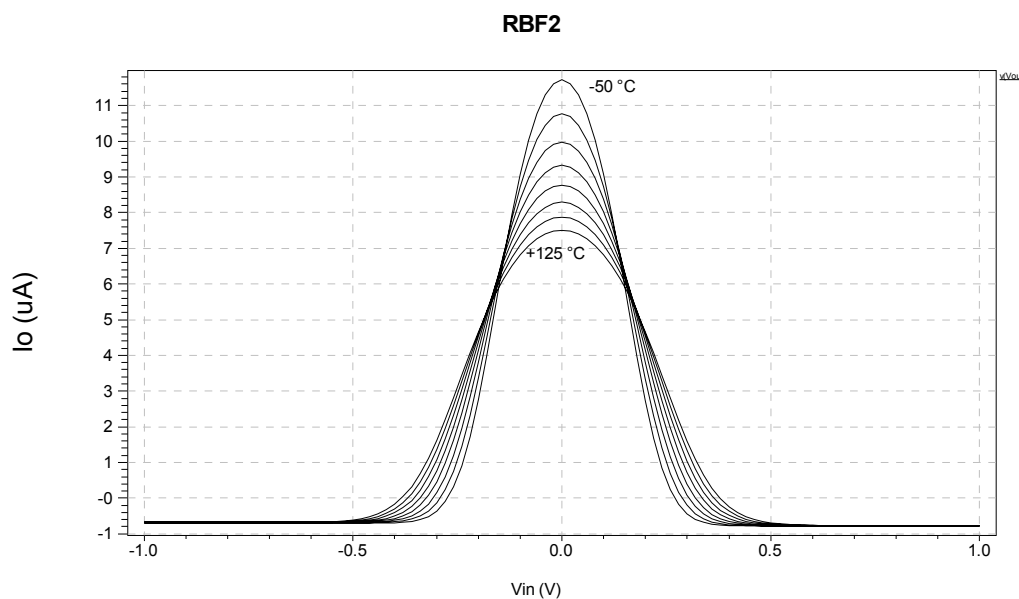


Figura 6.26: Saída de RBF₂ obtida para temperatura variando de -50 a +125 °C.

6.5.2 Análise de Pior Caso

A seguir serão apresentados resultados obtidos por meio de simulação de modelos de pior caso a fim de analisar os efeitos que variações de processo poderão provocar no sinal de saída da RBF₂. Os modelos utilizados foram: típico (tm), pior caso de potência (wp) e pior caso de velocidade (ws).

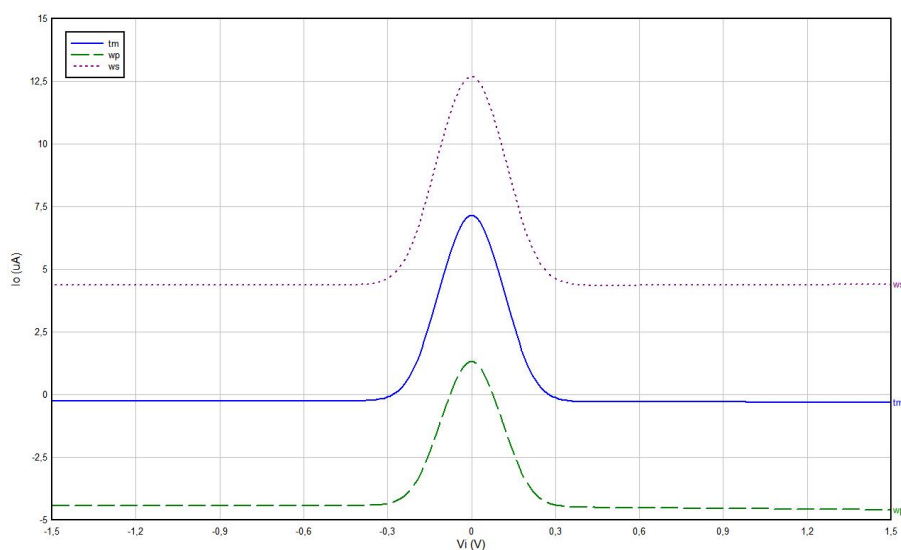


Figura 6.27: Análise de pior caso.

6.5.3 Análise Monte Carlo

A seguir serão apresentados resultados obtidos por meio de simulação de Monte Carlo para o circuito RBF₂. Analogamente ao circuito RBF₁, foram realizados estudos considerando distribuições de probabilidade uniformes e variações máximas de $\pm 1\%$ e $\pm 5\%$ para os parâmetros de projeto W e L, e também para os parâmetros de processo VTH0, TOX e U0.

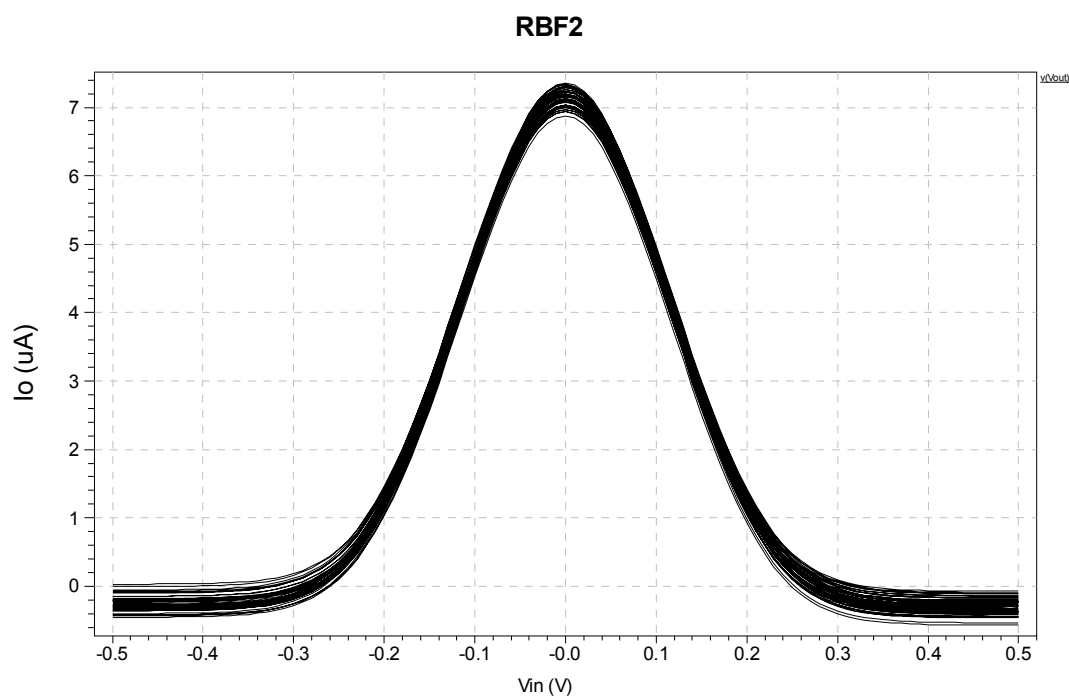


Figura 6.28: Análise Monte Carlo com variação de $\pm 1\%$ em L e $\pm 1\%$ em W.

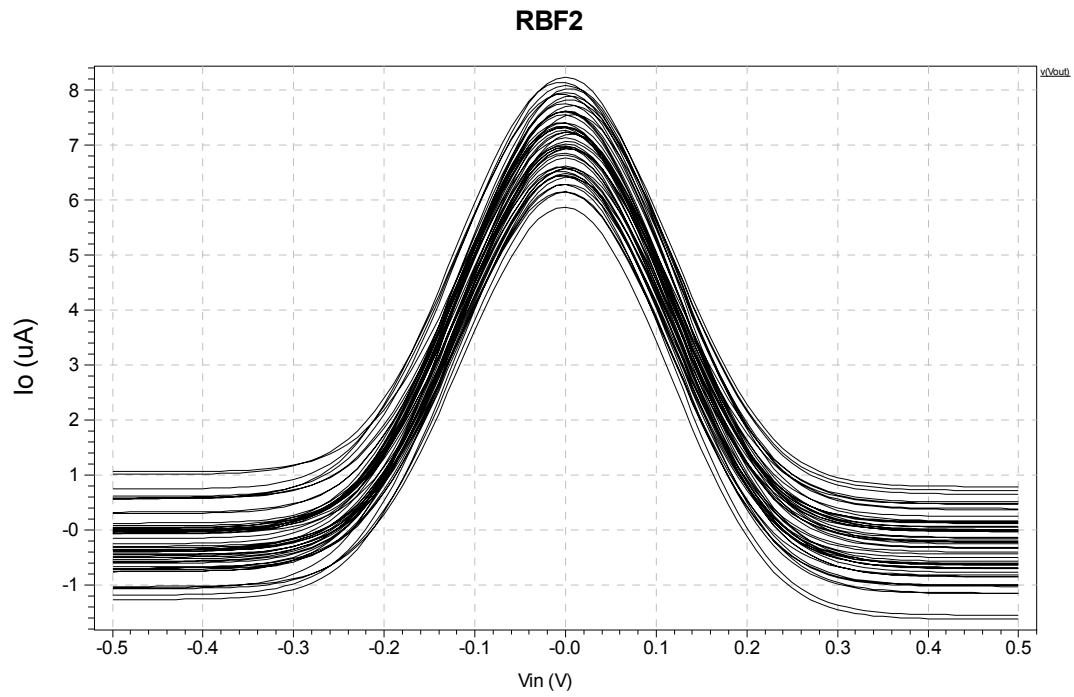


Figura 6.29: Análise Monte Carlo com variação de $\pm 5\%$ em L e $\pm 5\%$ em W.

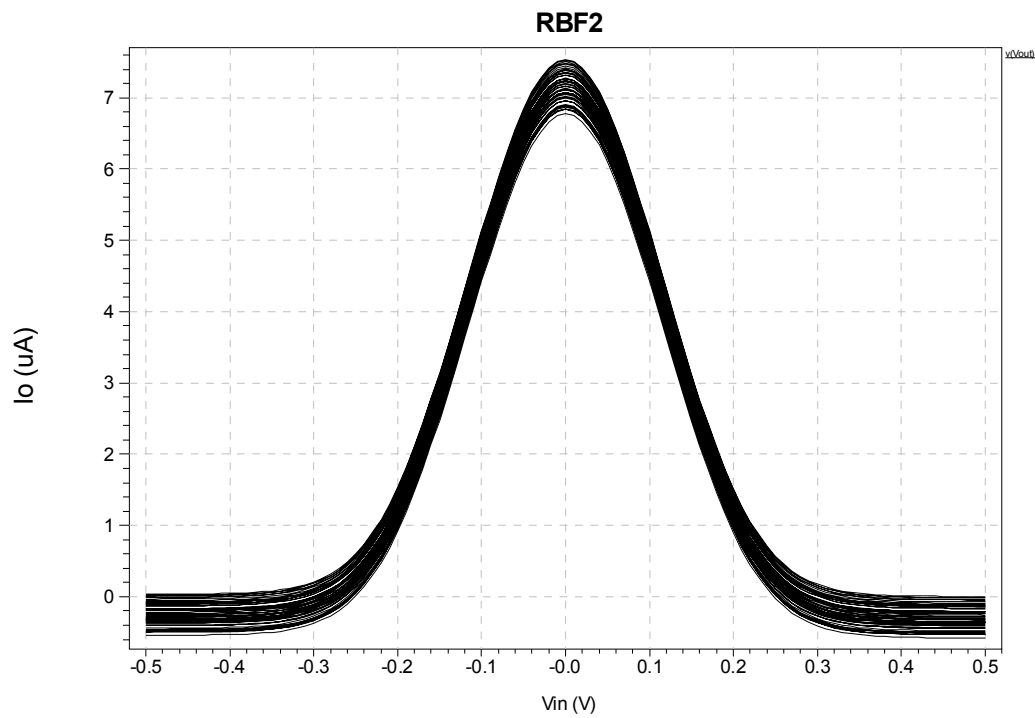


Figura 6.30: Análise Monte Carlo com variação de V_{TH0} em $\pm 1\%$.

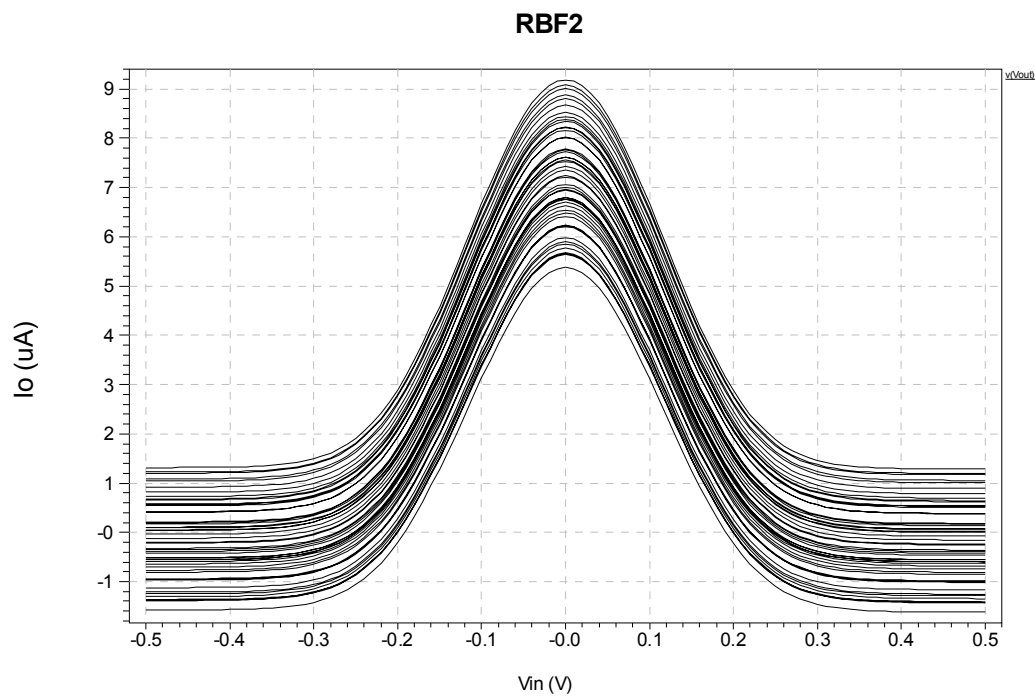


Figura 6.31: Análise Monte Carlo com variação de V_{TH0} em $\pm 5\%$.

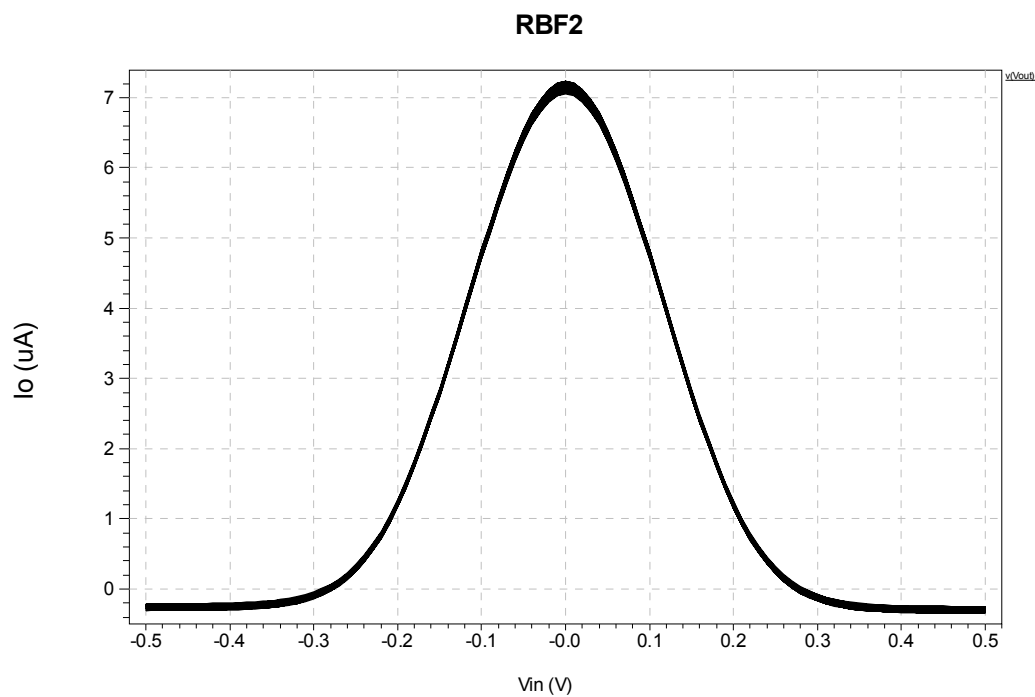


Figura 6.32: Análise Monte Carlo com variação de U_0 em $\pm 1\%$.

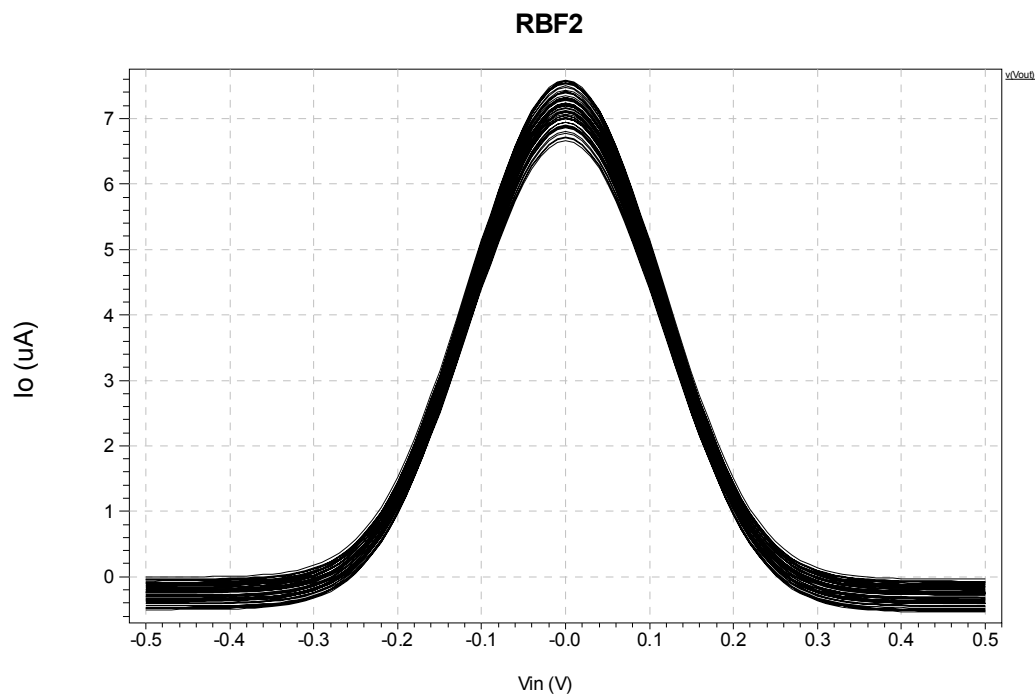


Figura 6.33: Análise Monte Carlo com variação de U_0 em $\pm 5\%$.

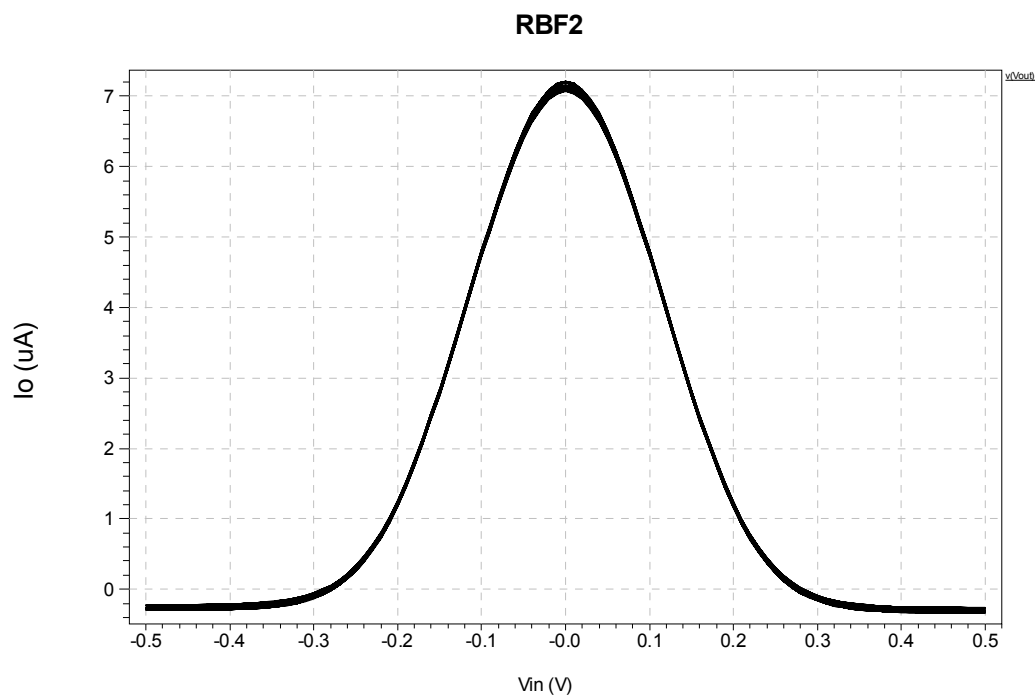


Figura 6.34: Análise Monte Carlo com variação de TOX em $\pm 1\%$.

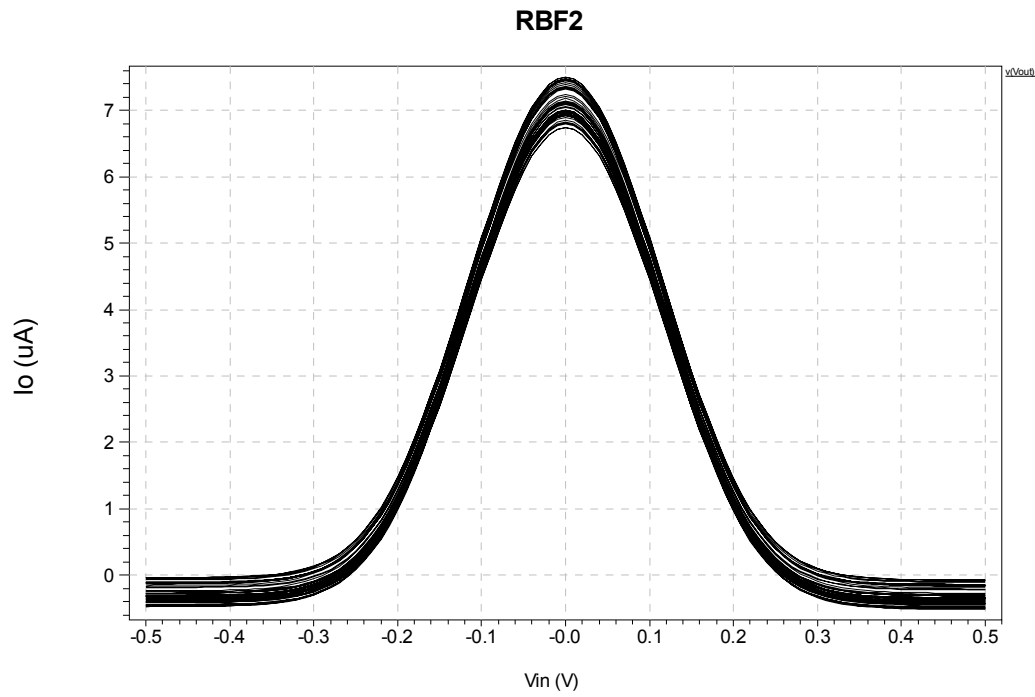


Figura 6.35: Análise Monte Carlo com variação de TOX em $\pm 5\%$.

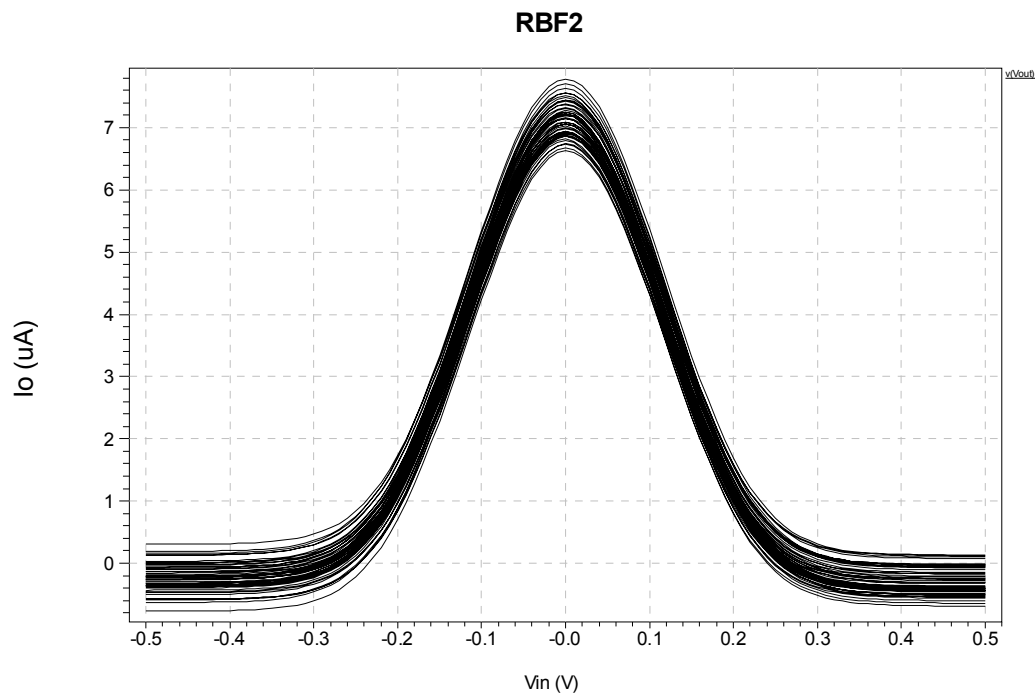


Figura 6.36: Análise Monte Carlo com variação combinada de VTH0, U0, TOX, W e L em $\pm 1\%$.

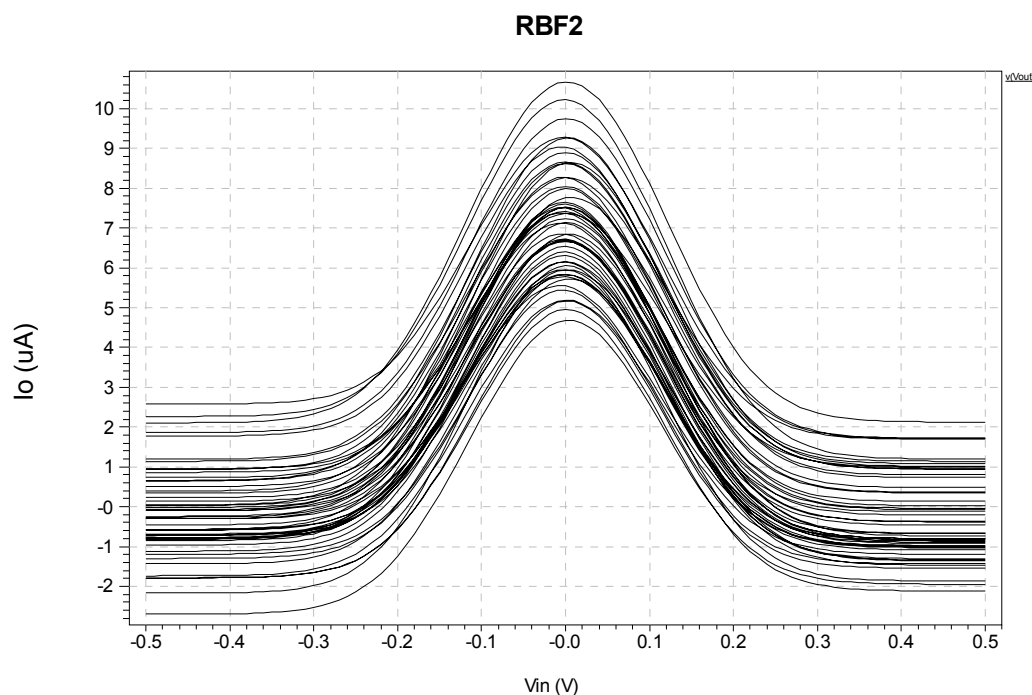


Figura 6.37: Análise Monte Carlo com variação combinada de V_{TH0} , U_0 , TOX , W e L em $\pm 5\%$.

Os resultados obtidos com a análise Monte Carlo foram compilados na Tabela 6.2, que apresenta a variação ocorrida na amplitude da RBF_2 em função da variação dos parâmetros de processo e de projeto.

Tabela 6.2: Variação no máximo da RBF_2 em função da variação de $\pm 1\%$ e $\pm 5\%$ dos parâmetros V_{TH0} , U_0 , TOX , W e L .

Parâmetro	Variação do parâmetro	
	$\pm 1\%$	$\pm 5\%$
V_{TH0}	787,97 nA	3,84 μA
U_0	212,26 nA	928,25 nA
TOX	162,75 nA	794,03 μA
W e L	486,01 nA	2,38 μA
Todos	1,16 μA	5,99 μA

6.6 Análise AC

Foram realizadas simulações para análise em frequência dos circuitos RBF₁ e RBF₂. Os resultados obtidos são apresentados nas Figuras 6.38 a 6.41. Foi utilizada uma corrente de polarização dos pares diferenciais de 5 μ A em ambos os circuitos.

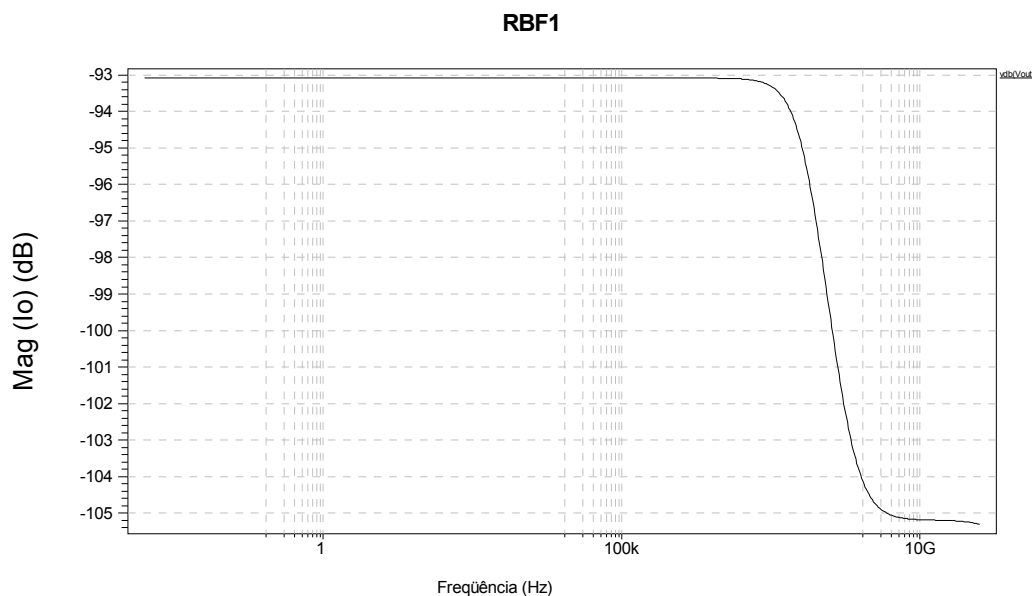


Figura 6.38: Magnitude da corrente de saída em função da frequência (RBF₁).

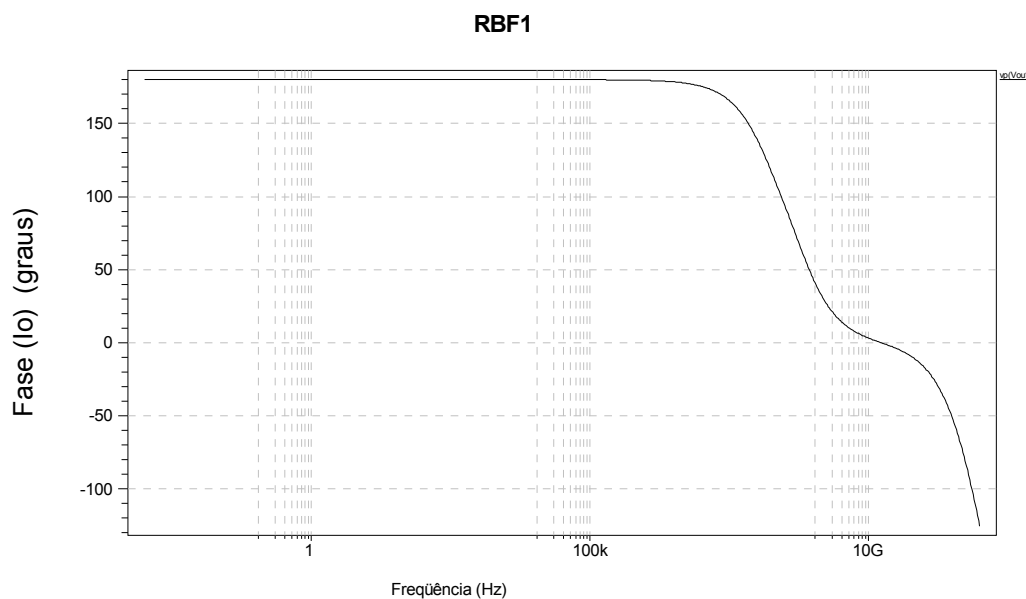


Figura 6.39: Fase da corrente de saída em função da frequência (RBF₁).

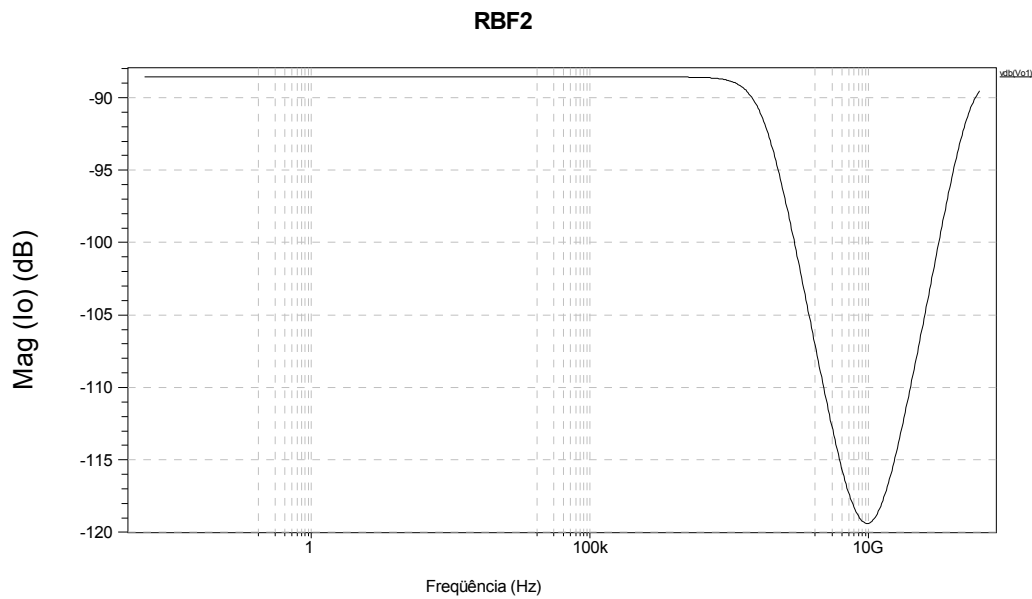


Figura 6.40: Magnitude da corrente de saída em função da frequência (RBF₂).

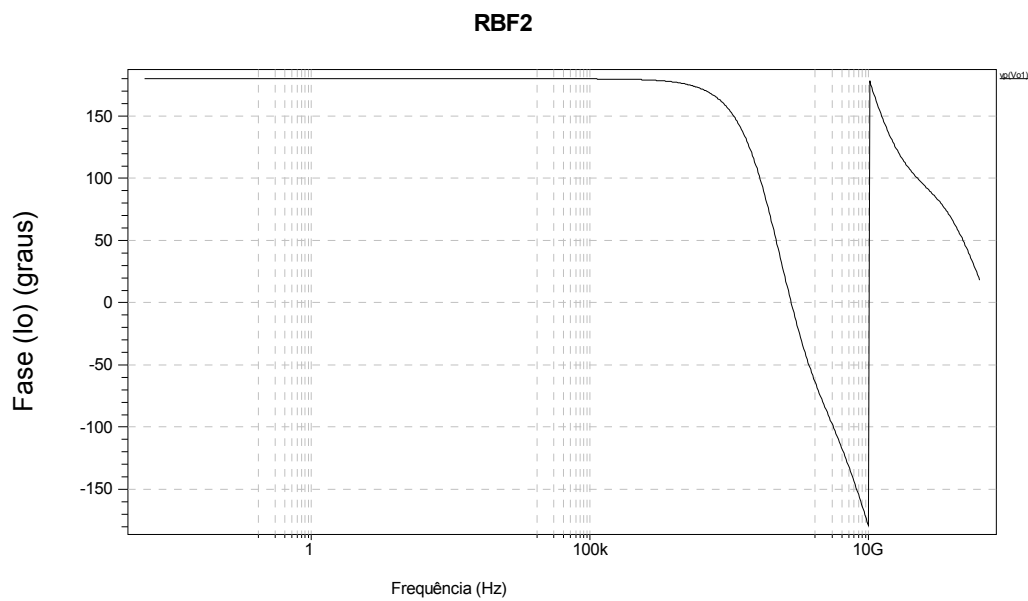


Figura 6.41: Fase da corrente de saída em função da frequência (RBF₂).

A análise em frequência mostrou que a frequência de corte do circuito (-3dB) está em aproximadamente 143 MHz para o circuito RBF₁ e em 130 MHz para o circuito RBF₂.

6.7 Circuito RBF₁ Bidimensional

O circuito RBF₁ bidimensional é constituído por dois circuitos RBF₁ unidimensionais, conforme mostrado na Figura 6.42. Os resultados de simulação são apresentados na Figura 6.44.

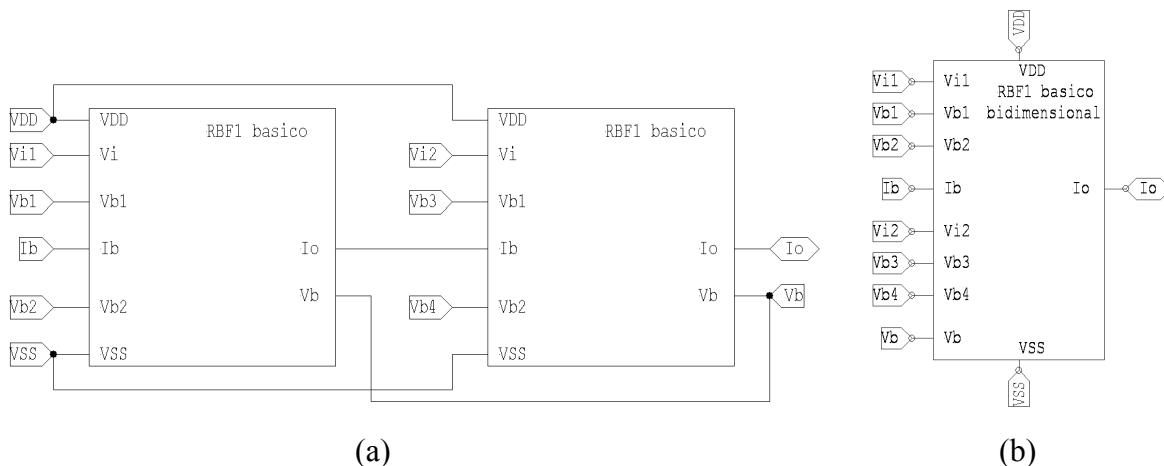


Figura 6.42: Circuito RBF₁ bidimensional. (a) Esquema interno.

(b) Símbolo do circuito RBF₁ bidimensional.

6.8 Circuito RBF₂ Bidimensional

O circuito RBF₂ bidimensional é constituído por dois circuitos RBF₂ unidimensionais, conforme mostrado na Figura 6.43. Os resultados de simulação são apresentados na Figura 6.45.

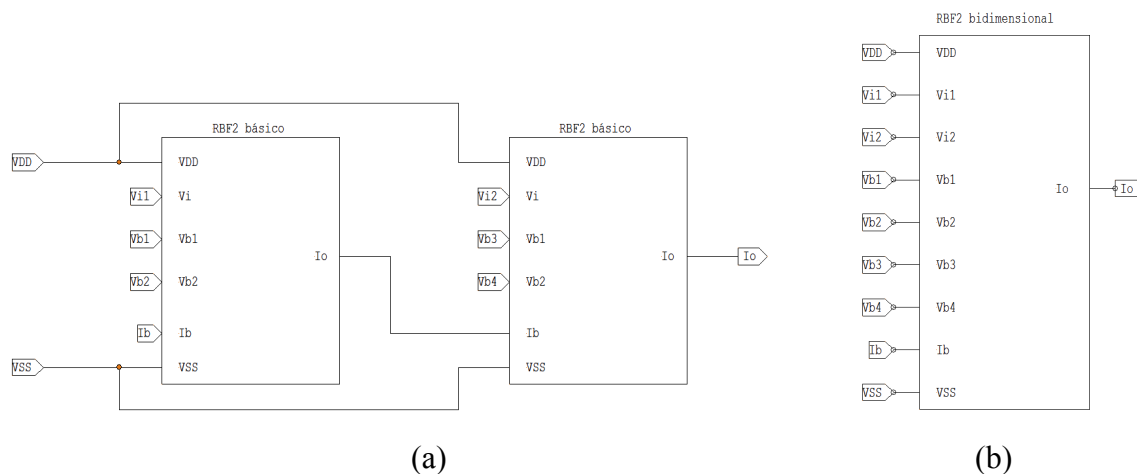


Figura 6.43: Circuito RBF₂ bidimensional. (a) Esquema interno.

(b) Símbolo do circuito RBF₂ bidimensional.

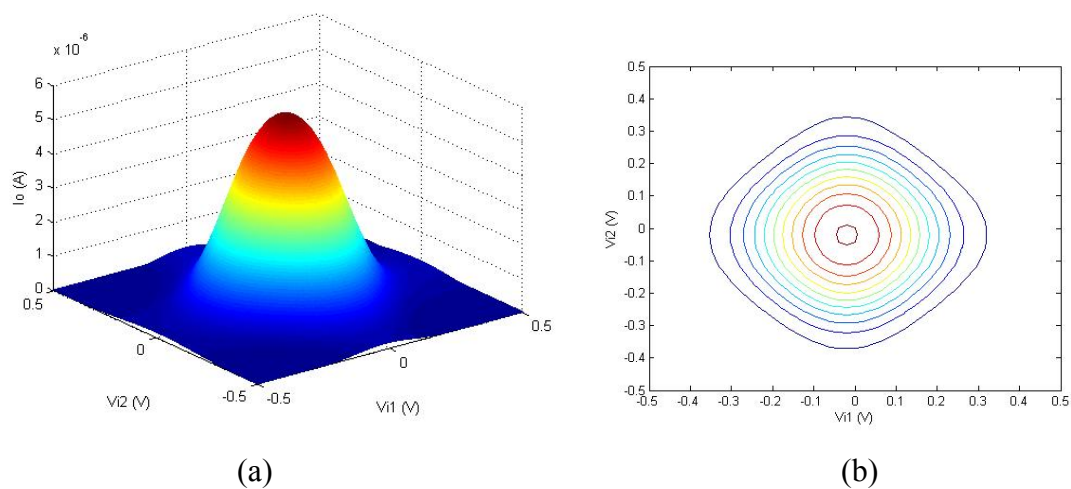


Figura 6.44: (a) Corrente de saída e (b) regiões de contorno da RBF₁ básica bidimensional.

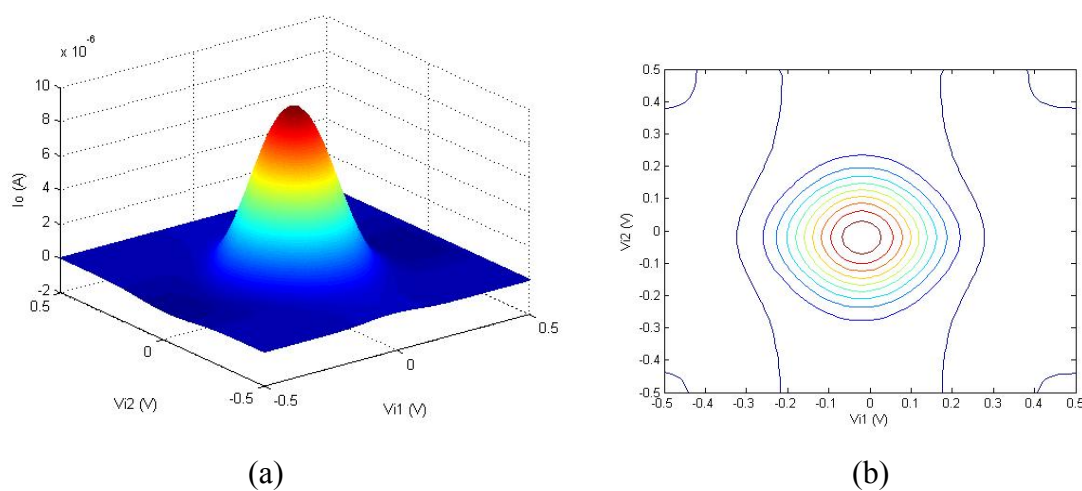


Figura 6.45: (a) Corrente de saída e (b) regiões de contorno da RBF₂ básica bidimensional.

Em ambos os casos, apresentados nas Figuras 6.44 e 6.45, os pares diferenciais que realizam as sigmóides estão polarizados com uma corrente de $5 \mu\text{A}$. As tensões de polarização são $V_{b1} = V_{b3} = -100\text{mV}$ e $V_{b2} = V_{b4} = +100\text{mV}$ para a RBF₁ e $V_{b1} = V_{b3} = +100\text{mV}$ e $V_{b2} = V_{b4} = -100\text{mV}$ para a RBF₂.

6.9 RBFN bidimensional

Uma RBFN RBF_1 bidimensional com quatro bases na camada intermediária simulada no *software* TANNER é apresentada na Figura 6.46. Na Figura 6.47, apresenta-se um exemplo de uma RBFN RBF_2 bidimensional.

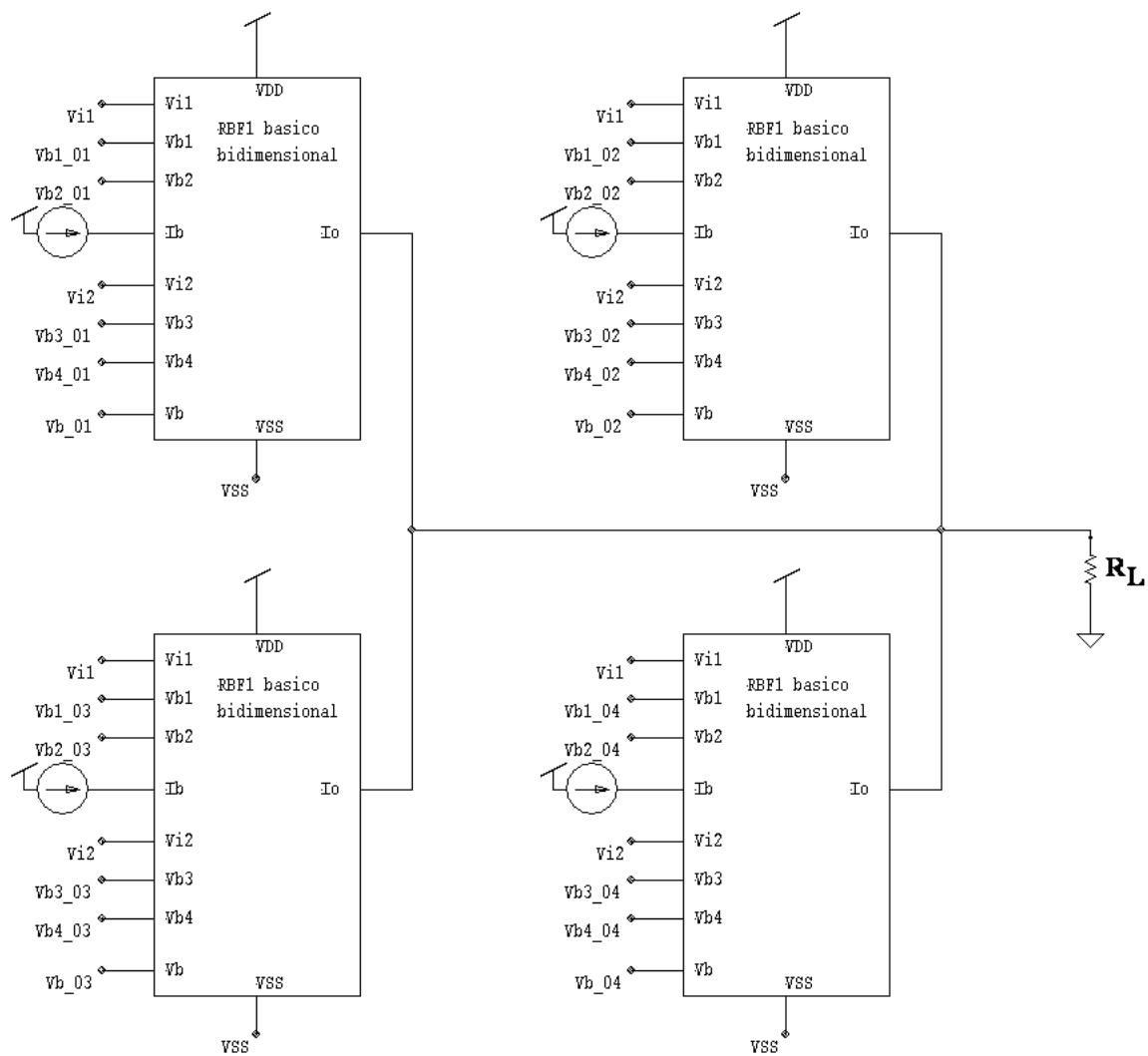


Figura 6.46: Diagrama esquemático de uma RBFN com duas entradas, uma saída e quatro bases RBF_1 bidimensionais na camada intermediária.

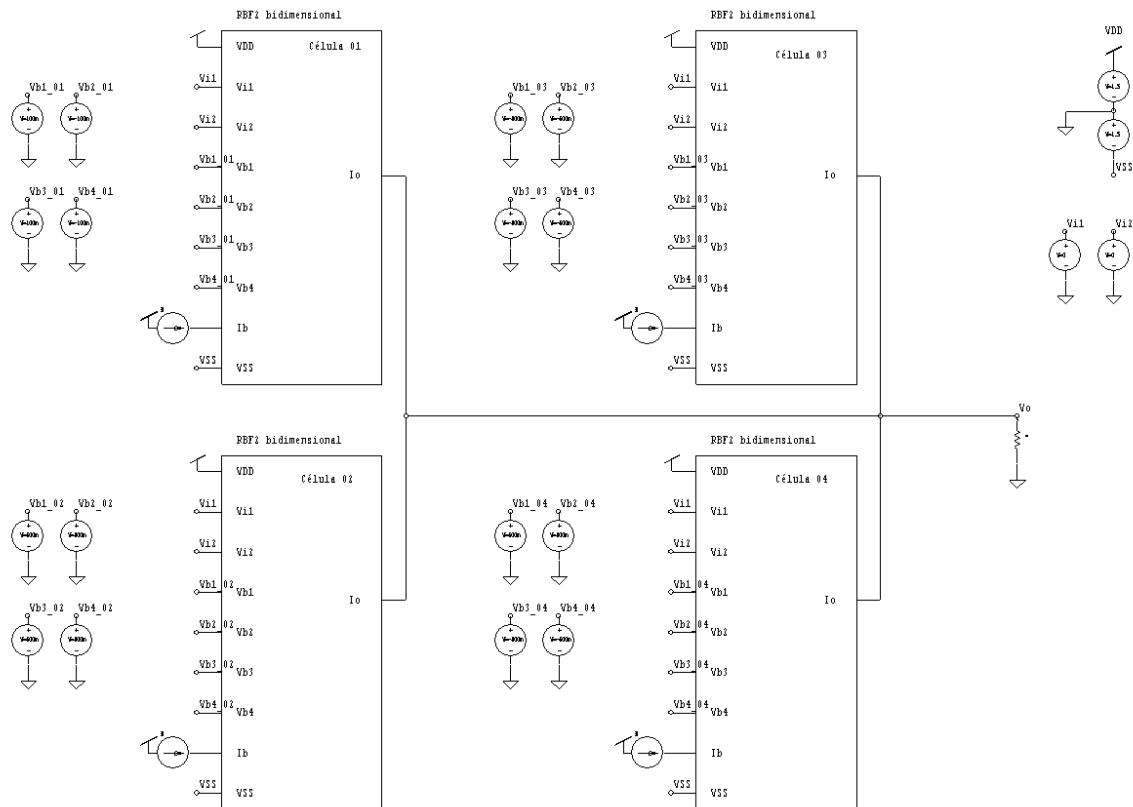


Figura 6.47: Diagrama esquemático de uma RBFN com duas entradas, uma saída e quatro bases RBF₂ bidimensionais na camada intermediária.

6.10 Discussão e Conclusões

Neste capítulo foram apresentados dois projetos de circuitos analógicos em tecnologia CMOS, os quais foram realizados por meio de simulações computacionais utilizando os modelos de transistores fornecidos no *design kit* AMS 0.35 e o pacote de simulação e projeto de circuitos integrados TANNER. Foram apresentados a arquitetura da rede, os blocos básicos e as simulações efetuadas. Pelos resultados obtidos da análise Monte Carlo, pode-se inferir que o principal efeito da variação de processo e do descasamento entre transistores é a introdução de um nível DC na saída do circuito. Com relação à frequência de operação, o circuito RBF₁ atingiu uma frequência de 143 MHz, enquanto o RBF₂ atingiu 130 MHz.

Capítulo 7

Resultados Obtidos para a RBFN CMOS

Neste capítulo serão mostradas algumas aplicações das redes desenvolvidas. A rede bidimensional será aplicada à resolução de algumas tarefas de aproximação de funções e classificação de padrões a fim de mostrar a sua funcionalidade. Essa verificação será realizada por meio de simulações computacionais.

7.1 Classificação de padrões

A função OU-Exclusivo é um exemplo clássico de uma função não linearmente separável. A RBFN proposta pode implementar essa função utilizando apenas duas bases, conforme mostrado nas Figuras 7.1 e 7.2. O nível lógico alto na entrada é dado por +0,5 V e o nível lógico baixo por -0,5 V.

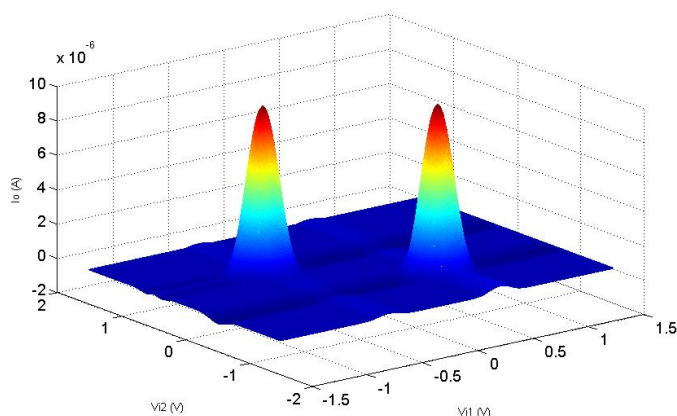


Figura 7.1: RBFN com duas bases que implementa uma função OU-Exclusivo.

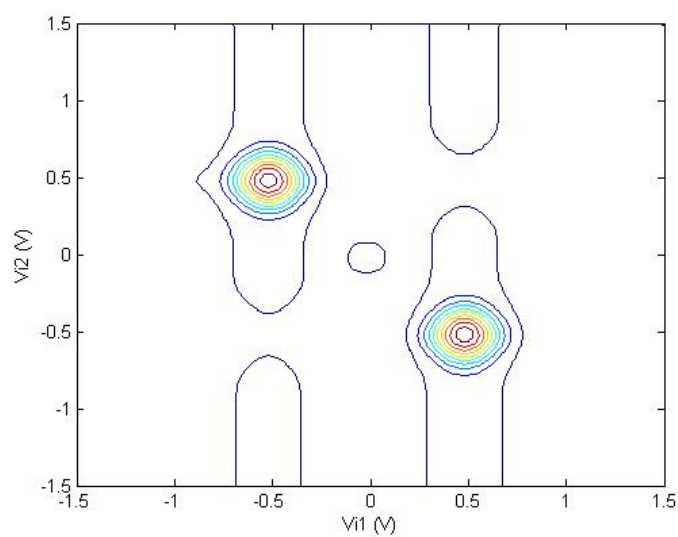


Figura 7.2: Regiões do espaço de entrada em que as RBFs estão localizadas para a realização da função OU-Exclusivo.

A Figura 7.3 apresenta os resultados obtidos para uma rede com cinco bases bidimensionais por meio de simulação no TANNER. A Figura 7.4 apresenta os mesmos resultados que a Figura 7.3, porém foi utilizado o *software* MATLAB, a fim de se obter uma visualização tridimensional dos resultados.

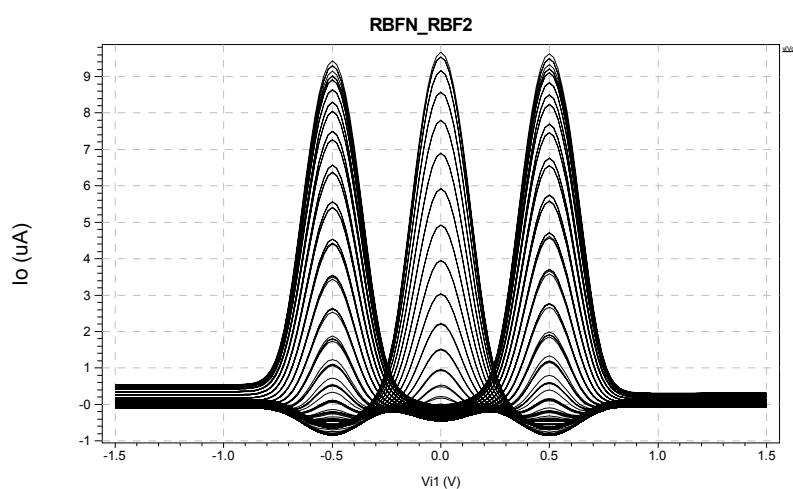


Figura 7.3: Resultados de simulação para uma rede bidimensional.

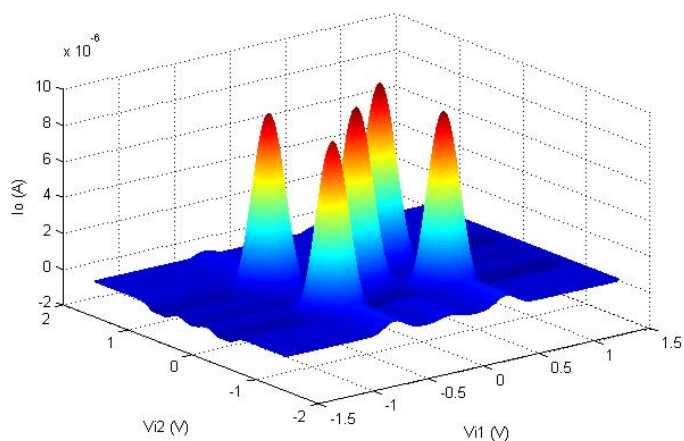


Figura 7.4: Corrente de saída de uma RBFN RBF₂ bidimensional com 05 unidades.

A Figura 7.5 mostra as regiões onde as RBFs estão localizadas no espaço definido pelas entradas V_{i1} e V_{i2} . A Figura 7.5 apresenta um exemplo de utilização da RBFN na separação de duas classes de padrões não linearmente separáveis.

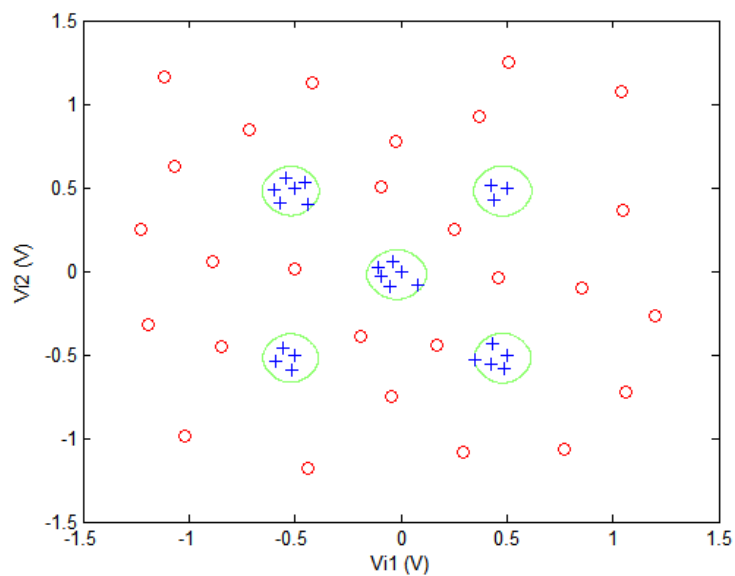


Figura 7.5: Exemplo de aplicação em classificação de padrões não linearmente separáveis.

7.2 Aproximação de Funções

Foram realizadas algumas simulações de aproximação de funções a fim de verificar o funcionamento da rede.

7.2.1 Função Senoidal

A rede proposta foi ajustada para aproximação de uma função senoidal. A entrada Vi2 da rede foi aterrada e variou-se a tensão na entrada Vi1. A resposta da rede em função da tensão de entrada Vi1 é mostrada na Figura 7.6. A resposta da rede a uma onda triangular na entrada é mostrada na Figura 7.7. A FFT do sinal de saída mostra uma componente DC e uma componente principal em uma frequência de 4 Hz.

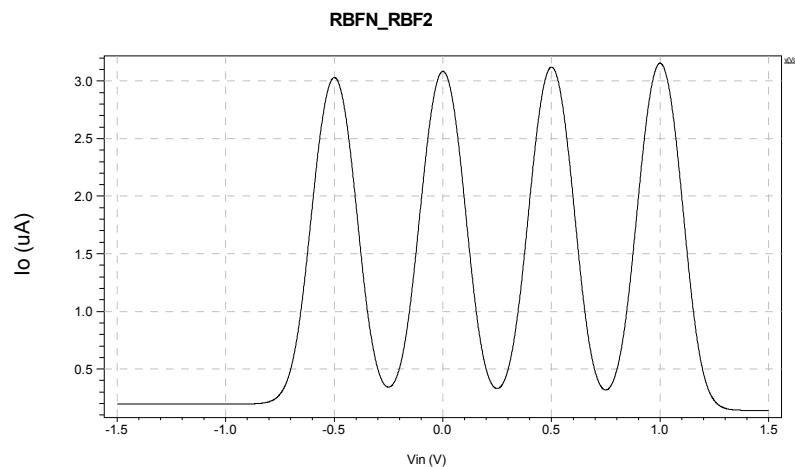


Figura 7.6: Análise DC.

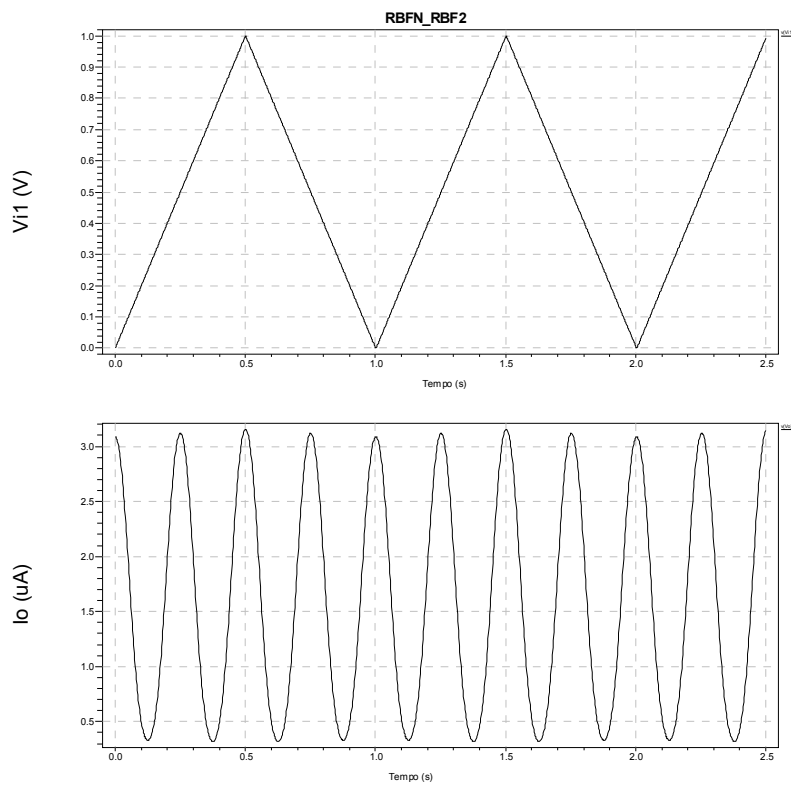


Figura 7.7: Análise transiente.

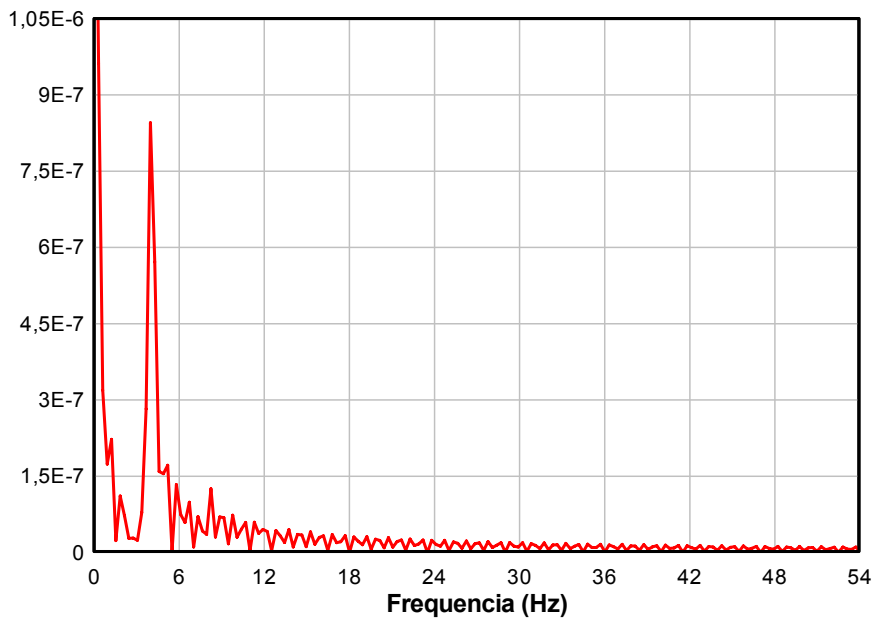


Figura 7.8: FFT do sinal de saída da rede.

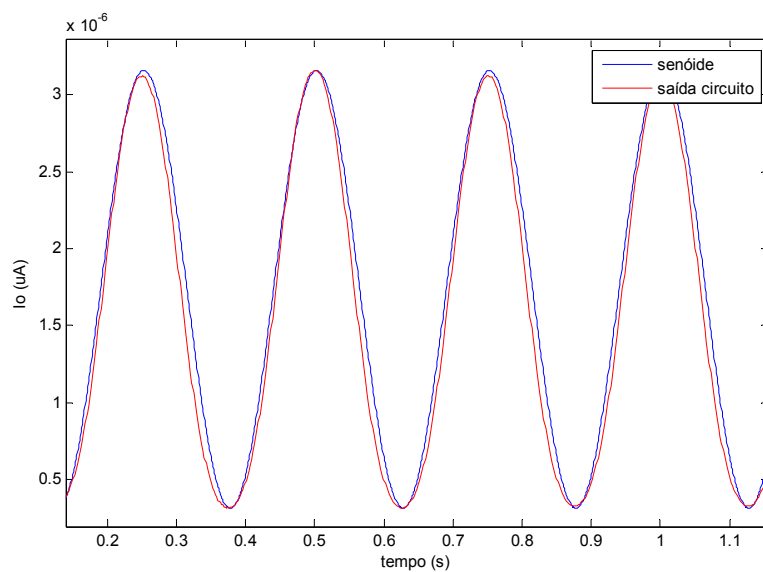


Figura 7.9: Sinal de saída da rede em relação à função senoidal

$$y = f(t) = 1,4195 \cdot 10^{-6} \cdot \text{sen}(8\pi t + 1,5) + 1,7364 \cdot 10^{-6}$$

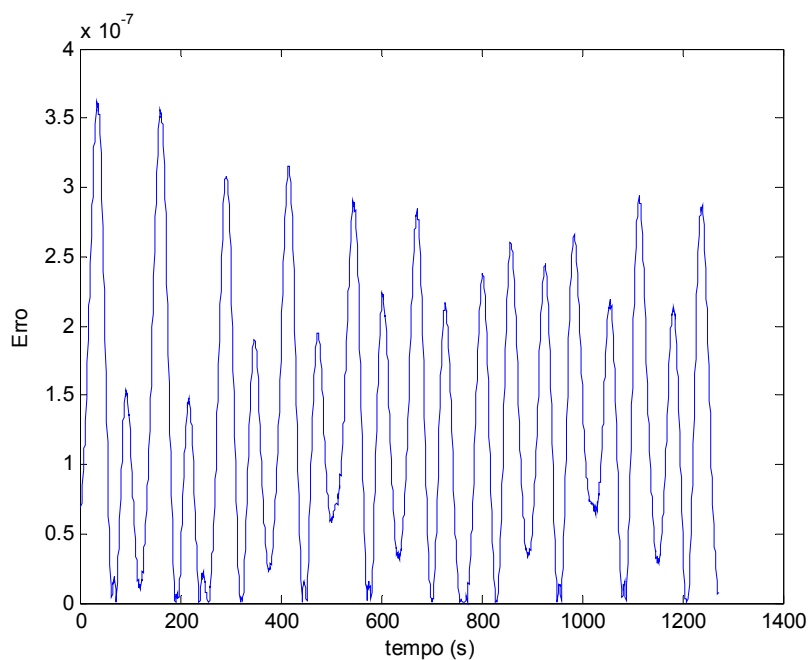


Figura 7.10: Erro absoluto entre o sinal de saída da rede e a função senoidal.

O erro absoluto máximo dessa aproximação em relação à função senoidal tomada como referência foi de $3,6142 \cdot 10^{-6}$, enquanto o erro absoluto médio foi de $1,2762 \cdot 10^{-6}$.

7.2.2 Funções de Pertinência para Lógica Nebulosa

Os circuitos RBF_1 e RBF_2 também podem ser ajustados para implementar outras funções além da função de base radial como, por exemplo, funções sigmóides. Isso é realizado pelo ajuste adequado de V_{b1} e V_{b2} , e também pela corrente de polarização I_b . Esse fato permite sua utilização para a implementação funções de pertinência em circuitos com lógica nebulosa.

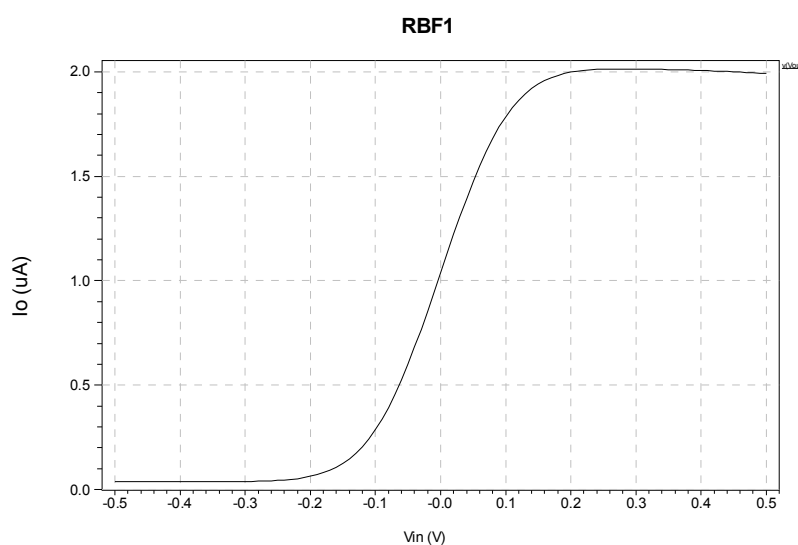


Figura: 7.11: Corrente de saída para $V_{b1} = 1.5$ V e $V_{b2} = 0$ V

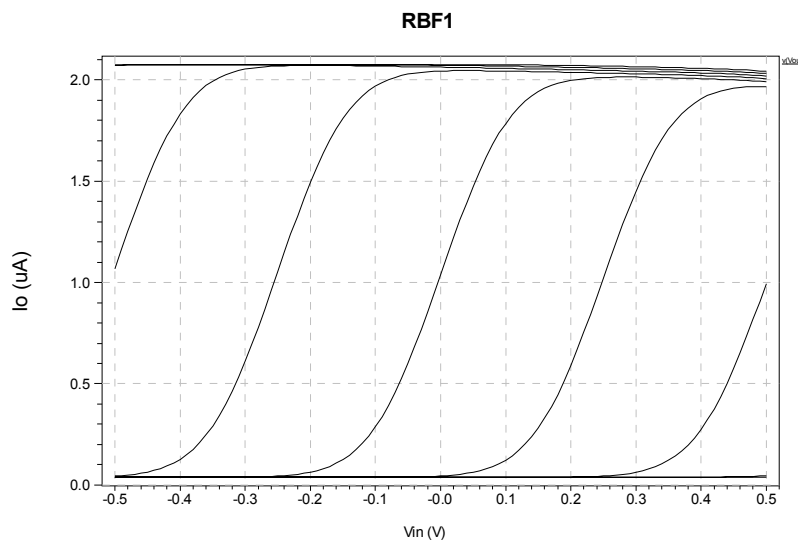


Figura 7.12: Corrente de saída para $V_{b1} = 1.5$ V e V_{b2} entre -500 mV e +500 mV.

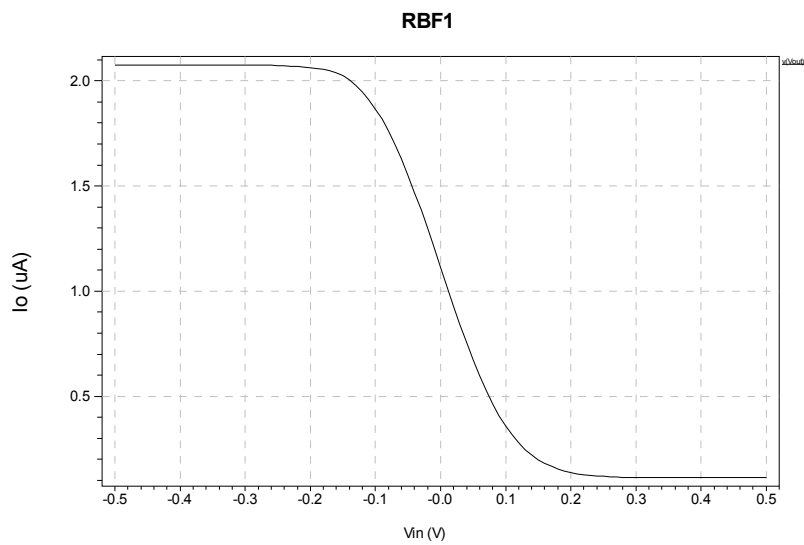


Figura 7.13: Corrente de saída para $V_{b1} = 0$ V e $V_{b2} = -1.5$ V.

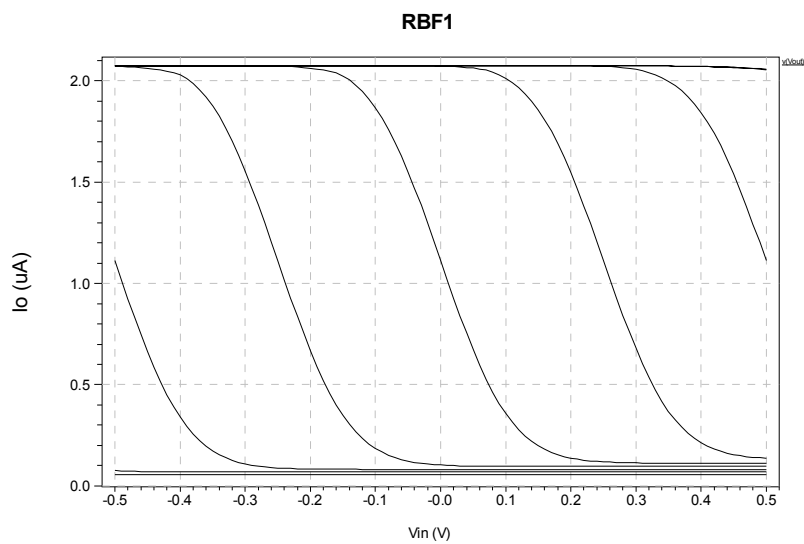


Figura 7.14: Corrente de saída para V_{b1} entre -500 mV e +500 mV e $V_{b2} = -1.5$ V.

Considerações Finais

Este trabalho foi dividido em três partes. A primeira parte apresentou uma introdução às redes neurais e às redes de base radial. Como contribuição, foi realizado um estudo sobre o efeito da quantização dos parâmetros na capacidade de aprendizado da RBFN. A capacidade da rede foi demonstrada com resultados de simulação para problemas de aproximação de funções. As simulações foram realizadas com vários níveis de quantização e os resultados mostraram que, mesmo com uma quantização severa de três bits, as redes apresentam a capacidade de aproximação de funções, porém de maneira limitada. Como era esperado, o erro aumenta para um número menor de níveis de quantização. Foi calculado um limite superior para o erro quadrático médio para uma rede nula, ou seja, com zero bit de quantização. Com relação à implementação mista (*mixed-signal*), a utilização de menos bits diminui o tamanho da memória e a complexidade necessária para armazenar os parâmetros das redes, o que pode levar a circuitos mais compactos e mais adequados para aplicações de baixo consumo de potência.

Na segunda parte da tese, foram propostos dois circuitos novos para a implementação de funções de base radial, os quais foram denominados RBF₁ e RBF₂, ambos implementados em tecnologia CMOS. Os circuitos RBF₁ e RBF₂ foram implementados utilizando circuitos integrados comerciais (MOSFET *arrays*) ALD1105, ALD1106 e ALD1107. Os resultados experimentais obtidos foram compatíveis com o que era esperado por meio de simulação computacional. Foi implementado um circuito RBF₁ bidimensional, consistindo de dois circuitos unidimensionais em série, sendo a corrente de polarização do segundo circuito controlada pela corrente de saída do primeiro. Foram realizados vários experimentos variando os parâmetros dos circuitos para o caso bidimensional, demonstrando assim a viabilidade de utilização dos circuitos para o caso multidimensional. Além disso, foram propostas versões em tecnologia BiCMOS e apresentados alguns resultados de simulação.

A terceira parte da tese apresenta a arquitetura da rede de base radial proposta, consistindo em um núcleo computacional analógico, utilizando os circuitos desenvolvidos na segunda parte da tese, o qual deverá ser futuramente utilizado em uma RBFN implementada na forma de circuito integrado. Os circuitos RBF1 e RBF2 permitem o ajuste de todos os parâmetros de forma da função de base radial, embora não de forma independente, ou seja, ao se ajustar um parâmetro os demais também são modificados. Os circuitos propostos foram aplicados na realização de algumas tarefas de aproximação de funções e classificação de padrões, tais como a função OU-Exclusivo, uma função senoidal e funções de pertinência utilizadas em lógica nebulosa, apresentando bons resultados.

Propostas para Trabalhos Futuros

Nesta tese foi apresentado o desenvolvimento de circuitos CMOS e BiCMOS para a implementação de RBFNs. Foram apresentados resultados de simulação e também de implementação prática utilizando componentes comerciais. Porém, a implementação das redes na forma de circuito integrado é importante para o desenvolvimento de redes práticas e para estender o leque de aplicações práticas das redes. Assim sendo, sugere-se como possíveis trabalhos futuros o seguinte:

- Implementação e teste dos circuitos propostos na forma integrada;
- Aplicação dos circuitos para mais de duas variáveis de entrada e mais de uma variável de saída;
- Utilização dos circuitos em lógica nebulosa;
- Utilização das redes propostas em controle automático;
- Utilização de metaheurísticas, como por exemplo, *particle swarm* e algoritmo genético, para a execução da fase de aprendizado;

Referências Bibliográficas

ABUELMA'ATI, M. T., SHWEHNEH, A., A Reconfigurable Gaussian/Triangular Basis Functions Computation Circuit. **Analog Integrated Circuits and Signal Processing**, 47, pp. 53–64, 2006.

AL-KAZEMI, B., MOHAN, C. K., Training Feedforward Neural Networks Using Multi-Phase Particle Swarm Optimization. **Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02)**, vol.5, pp. 2615-2619, Singapore, 2002.

BROOMHEAD, D. S., LOWE, D., Multivariate functional interpolation and adaptive network. **Complex Systems**, vol. 2, pp. 321-355, 1988.

ÇEVIKBAŞ, I. C., OGRENCI, A. S., DUNDAR, G., BALKUR, S., VLSI Implementation of GRBF (Gaussian Radial Basis Function) Networks. In: **ISCAS 2000 - IEEE International Symposium on Circuits and Systems**, vol.3, pp. 646-649, Geneva, Switzerland, May 28-31, 2000.

CHOI, J., SHEU, B. J., CHANG, C. F., A Gaussian Synapse Circuit for Analog VLSI Neural Networks. **IEEE Trans. on VLSI Systems**, vol. 2, no. 1, pp. 129-133, March, 1994.

CLERC, M., KENNEDY, J., The Particle Swarm – Explosion, Stability, and Convergence in a Multidimensional Complex Space. **IEEE Transactions on Evolutionary Computation**, vol.6, no.1, pp. 58-72, February, 2002.

CHURCHER S., MURRAY A. F., REEKIE H. M., Programmable Analogue VLSI for Radial Basis Function Networks. **Electronic Letters**, vol.29, no.18, pp. 1603-1605, September 1993.

DAYAN, P., ABBOTT, L.F., **Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems**. MIT Press, December, 2001.

DRAGHICI, S., MILLER, D.A., A VLSI neural network classifier based on integer-valued weights. **IJCNN99 - International Joint Conference on Neural Networks**, 1999. Volume 4, pp. 2419 – 2424, 1999.

DRAGHICI, S., The constraint based decomposition (CBD) training architecture. **Neural Networks**, 14, pp. 527-550, May, 2001.

DRAGHICI, S., On the capabilities of neural networks using limited precision weights. **Neural Networks** 15, pp. 395-414, 2002.

DONG, P., **Design, analysis and real-time realization of artificial neural network for control and classification.** PhD Dissertation, North Carolina State University, USA, 2006.

DORIGO, M., GAMBARDELLA, L. M., Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. **IEEE Transactions on Evolutionary Computation**, vol. I, no. 1, pp. 53-65, April, 1997.

DÜNDAR, G., ROSE, K., The Effects of Quantization on Multilayer Neural Networks. **IEEE Transactions on Neural Networks**, vol. 6, no.6, pp. 1446-1451, November, 1995.

EBERHART, R. C., SHI, Y., Tracking and Optimizing Dynamic Systems with Particle Swarms. **Proceedings of the 2001 Congress on Evolutionary Computation**, Volume 1, pp. 94 - 100, 2001.

GIROSI, F., POGGIO, T., **Networks and the best approximation property.** A.I. Memo N. 1164. MIT - Artificial Intelligence Laboratory, 1989.

GREENSTEIN, B., GREENSTEIN A., **Color Atlas of Neuroscience: Neuroanatomy and Neurophysiology.** Ed. Thieme, New York, NY, USA.

HAYKIN S., **Neural Networks - A Comprehensive Foundation.** Macmillan Publisher Company, 1994.

HO, S. L., YANG, S., NI, G., MACHADO, J. M., A Modified Ant Colony Optimization Algorithm Modeled on Tabu-Search Methods. **IEEE Transactions on Magnetics**, vol.42, no.4, pp 1195-1198, April, 2006.

JI, C., PSALTIS, D., Capacity of two-layer feedforward neural networks with binary weights. **IEEE Transactions on Information Theory**, vol. 44, no.1, pp. 256-268, January, 1998.

KAMIO T., TANAKA S., MORISUE M., Backpropagation Algorithm for Logic Oriented Neural Networks, **IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)** - Volume 2, pp. 123-128, Como, Italy, 2000.

KHAN, A. H., **Feedforward Neural Networks with Constrained Weights**, PhD dissertation, University of Warwick, UK, August, 1996.

KWON, O-J, BANG, S-Y., Comments on "The Effects of Quantization on Multilayer Neural Networks. **IEEE Transactions on Neural Networks**, vol. 9, no.4, pp. 718-719, July, 1998.

LACERDA, E. G. M., CARVALHO, A.C.P.L.F., LUDERMIR, T.B., Evolutionary Optimization of RBF Networks. **Proceedings of the VI Brazilian Symposium on Neural Networks (SBRN'00)**, Rio de Janeiro, Brasil, pp. 219-224, 22-25 de Novembro, 2000.

LACERDA, E. G. M., CARVALHO, A.C.P.L.F., LUDERMIR, T. B.: Evolutionary Optimization of RBF Networks. **International Journal of Neural Systems**, v.11, p. 1-12, 2001.

LACERDA, E. G. M., CARVALHO, A.C.P.L.F., LUDERMIR, T. B.: Model Selection via Genetic Algorithms for RBF Networks. **Journal of Intelligent & Fuzzy Systems**, v.13, pp. 111-122, 2003.

LEE S., KIL R. M., A Gaussian Potential Function Network With Hierarchically Self-Organizing Learning. **Neural Networks**, vol. 4, pp. 207-224, 1991.

LEHMANN, T., **Hardware Learning in Analogue VLSI Neural Networks**. PhD dissertation, Technical University of Denmark, 1994.

LENDASSE, A., LEE, J., de BODT, E., WERTZ, V., VERLEYSEN, M., **Approximation by Radial Basis Function Networks - Application to Option Pricing**. In: *Connectionist Approaches in Economics and Management Sciences*, Chapter 10. C. Lesage, M. Cottrell eds., Kluwer academic publishers, pp. 203-214, 2003.

LIN, B-S., LIN, B-S., CHONG, F-C., FEIPEI, L., Higher-Order-Statistics-Based Radial Basis Function Networks for Signal Enhancement. **IEEE Trans. on Neural Networks**, vol.18, no.3, pp. 823-832, May, 2007.

LIN, S-Y., HUANG, R-J., CHIUEH, T-D., A Tunable Gaussian/Square Function Computation Circuit for Analog Neural Networks. **IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing**, vol.45, no.3., pp. 441-446, March, 1988.

LUCKS, M. B., **Implementação Eletrônica de uma Rede Neural Artificial com Função de Base Radial e suas Aplicações**. Dissertação de mestrado. FEIS-UNESP, Ilha Solteira, 1998.

LUCKS, M. B., OKI, N., A Radial Basis Function Network (RBFN) Circuit for Function Approximation. In: **42nd IEEE Midwest Symposium on Circuits and Systems**, vol.2, pp. 1099-1101, Las Cruces, New Mexico, USA, 1999.

LUCKS, M. B., OKI, N., An Analog Implementation of Radial Basis Function Network Appropriate for Transducer Linearizer. In: **42nd IEEE Midwest Symposium on Circuits and Systems**, vol.2, pp. 1109-1112, Las Cruces, New Mexico, USA, 1999.

LUDERMIR, T. B., YAMAZAKI, A., ZANCHETTIN, C., An Optimization Methodology for Neural Network Weights and Architectures. **IEEE Transactions on Neural Networks**, v.17, pp. 1452 - 1459, 2006.

MCDONALD, D. B., GRANTHAM, W. J., TABOR, W. L. MURPHY, M. J. Global and local optimization using radial basis function response surface models. **Applied Mathematical Modelling**, 31, pp. 2095-2110, 2007.

MADRENAS, J., VERLEYSEN, M., THISSEN, P., VOZ, J. L., A CMOS Analog Circuit for Gaussian Functions **IEEE Trans. on Circuits and Systems-II : Analog and Digital Signal Processing**, vol. 43, no. 1, pp. 70-74, January, 1996.

MARSHALL, G.; COLLINS, S., A compact analogue radial basis function circuit. **Fourth International Conference on Artificial Neural Networks**, pp. 471 – 476, 26-28 Jun, 1995.

MOODY, J. E., DARKEN, C.J. Fast learning in networks of locally-tuned processing units. **Neural Computation**, vol. 1, pp. 281-294, 1989.

NAGASHIMA, R., OKI, N., A Radial Basis Neural Network Implementation in BiCMOS Process and its Learning through the Evolutionary Computation. In: **Proceedings of XI IBERCHIP Workshop, 2005. v. CD-ROM**, Salvador, 2005.

NAGASHIMA, R., **Treinamento de uma Rede Neural de Base Radial Usando Computação Evolutiva: Implementação e Aplicações**. Dissertação de mestrado, FEIS, UNESP, Ilha Solteira, 2006.

NERUDA, R., KUDOVÁ P., Learning Methods for radial basis function networks. **Future Generation Computer Systems**, 21, pp. 1131-1142, 2005.

NUNES DE CASTRO, L., Immune, Swarm, and Evolutionary algorithms Part I: Basic Models. **Proc. of the International Conference on Neural Information Processing (ICONIP)**, Workshop on Artificial Immune Systems, vol. 3, pp. 1464-1468, Singapore, 2002.

NUNES DE CASTRO, L., Immune, Swarm, and Evolutionary algorithms Part II: Philosophical Comparisons. **Proc. of the International Conference on Neural Information Processing (ICONIP)**, Workshop on Artificial Immune Systems, vol. 3, pp. 1469-1473, Singapore, 2002.

OGAWA, M., SHIBATA, T., NMOS-based Gaussian-element-matching analog associative memory. **Proceedings of the 27th European Solid-State Circuits Conference, ESSCIRC 2001**, pp. 257 – 260, 18-20 Sept., 2001.

OLIVEIRA, J. P.; OKI, N., An Analog Implementation of Radial Basis Neural Networks (RBNN) using BiCMOS Technology. In: **2001 Midwest Symposium on Circuits and Systems**, vol.2, pp. 705-708, Dayton, Ohio, August 14 - 17, 2001.

OLIVEIRA, J. P.; OKI, N., A Programmable Analog Gaussian Function Synthesizer. **Proceedings of SBMicro**, Porto Alegre, 2002.

PENG, S.-Y., MINCH, B. A., HASLER, P., A Programmable Floating-Gate Bump Circuit with Variable Width. In: **ISCAS 2005 - IEEE International Symposium on Circuits and Systems**, vol.5, pp. 4341-4344, Kobe, Japan, May 23-26, 2005.

PLAGIANAKOS, V. P., VRAHATIS, M. N., Training Neural Networks with Threshold Activation Functions and Constrained Integer Weights. In: **Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000**, pp.161-166, Italy, 2000.

ROSENBLUM, M., YACOOB, Y., DAVIS, L. S., Human Express Recognition from Motion Using Radial Basis Function Network for Visual Architecture. **IEEE Trans. on Neural Networks**, vol.7, no.5, pp. 1121-1137, September, 1996.

SEN, S., ROBERTSON, W., PHILLIPS, W., The Effects of Limited Precision Implementation on the Performance of Radial Basis Functions, **CCECE'97 Proceedings of the Canadian Conference on Electrical and Computer Engineering**, 1997.

SING, J. K., BASUA, D. K., NASIPURIA, M., KUNDUA M., Face recognition using point symmetry distance-based RBF network. **Applied Soft Computing**, v.7, pp. 58-70, 2007.

SHI, Y., EBERHART, R., A Modified Particle Swarm Optimizer. **IEEE International Conference on Evolutionary Computation Proceedings**. IEEE World Congress on Computational Intelligence, pp. 69 – 73, May, 1998.

THEOGARAJAN, L., KERS, L. A., A Scalable Low Voltage Analog Gaussian Radial Basis Circuit. **IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing**, vol.44, no.11, pp. 977-979, November, 1997.

TURCHETTI C., CONTI M., A General Approach to Nonlinear Synthesis with MOS Analog Circuits. **IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications**, vol.40, no.9, September, 1993.

VOITOVETSKY, I.; GUTERMAN, H., Encoding resolution in Hopfield associative memory networks. **Eighteenth Convention of Electrical and Electronics Engineers in Israel**, pp. 4.3.2/1 - 4.3.2/4, 07-08 Mar, 1995.

WANG, C., HILL, D. J., Deterministic Learning and Rapid Dynamical Pattern Recognition. **IEEE Transactions on Neural Networks**, vol. 18, no.3, pp.617-630, May, 2007.

WASSERMAN P. D., **Neural Computing: Theory and Practice**. Van Nostrand Reinhold, 1989.

WATKINS S. S., CHAU P. M., A Radial Basis Function Neurocomputer Implemented with Analog VLSI Circuits. **Int. Joint Conf. on Neural Networks**, vol.2, pp. 607-612, 1992.

WATKINS, S. S., CHAU, P. M., Reduced-complexity circuit for neural networks **Electronics Letters**, vol. 31, Issue 19, pp. 1644 -1646, Sep., 1995.

WRAY, J., GREEN, G. G. R., Neural networks, approximation theory, and finite precision computing. **Neural Networks**, 8(1):31-37, 1995.

XIE, Y. JABRI M. A., Analysis of the Effects of Quantization in Multilayer Neural Networks Using a Statistical Model. **IEEE Transactions on Neural Networks**, vol. 3, no.2, pp.334-338, March, 1992.

YAMAZAKI, A. Y., LUDERMIR, T. B., SOUTO, M. C. P., Global Optimization Methods for Designing and Training Neural Networks. In **Proceedings of the VII Brazilian Symposium on Neural Networks, SBRN'02**, pp. 136-141, Recife, Brazil, 2002.

ZANCHETTIN, C., LUDERMIR, T. B.: Global Optimization Methods for Designing and Training Feedforward Artificial Neural Networks. **Dynamics of Continuous, Discrete and Impulsive System**, v.14, p.328 - 337, 2007.

ZANCHETTIN, C., LUDERMIR, T. B.: Wavelet filter for noise reduction and signal compression in an artificial nose. **Applied Soft Computing**, v.7, p.246 - 256, 2007.

ZHAO, T., RBFN-based decentralized adaptive control of a class of large-scale non-affine nonlinear systems. **Neural Computing and Applications**, Special Issue: Neural networks for control, robotics and diagnostics, Volume 17, Issue 4, pp.357-364, June, 2008.

Apêndices

Apêndice A: Artigos publicados

Congressos Internacionais:

LUCKS, M. B. e OKI, N., “Radial Basis Function Network with Quantized Parameters”, **IEEE Computational Intelligence for Measurement Systems and Applications – CIMSA 2008**, Istambul, Turquia, Julho, 2008.

LUCKS, M. B. e OKI, N., “RBF Circuits Based on Folded Cascode Differential Pairs”, **SBCCI 2008**, Gramado, RS, Brasil, Setembro, 2008.

LUCKS, M. B. e OKI, N., “Radial Basis Function Network Applied to the Linearization of a Voltage Controlled Oscillator”, **SBCCI 2007**, Rio de Janeiro, RJ, Brasil, Setembro, 2007.

Congressos Nacionais:

LUCKS, M. B. e OKI, N., “A Radial Basis Function Network with Quantized Parameters Applied to Function Approximation”. In: **7º Congresso Temático de Dinâmica e Controle da SBMAC – Dincon 2008**, Presidente Prudente, SP, Maio, 2008.

LUCKS, M. B. e OKI, N., “Radial Basis Function Network with Quantized Parameters”. **XVII Congresso Brasileiro de Automática – CBA 2008**, Juiz de Fora, MG, Setembro, 2008.

LUCKS, M. B. e OKI, N., “Derivative Approximation for Analog Design”. **VII Microelectronics Students Forum – SFORUM 2007**, Rio de Janeiro, RJ, Brasil, Setembro, 2007.

Apêndice B: Funções desenvolvidas no MATLAB para as redes unidimensionais

defrbfn.m

```
% definição inicial da rede de base radial gaussiana unidimensional
% definição dos parametros do algoritmo de aprendizado
```

```
m=10; % m = número de bases
xd=0:0.1:4*pi; % padrão de entrada
yd=sin(xd); % saída desejada
eqsmaxd=0.01; % eqsmaxd = erro quadrático máximo desejado
```

```
maxiter=100; % maxiter = número máximo de iterações
n=length(xd); % n = comprimento do vetor de entrada
```

```
for i=1:m
    a(i)=1;
    b(i)=rand;
    c(i)=rand*(max(xd));
    w(i)=rand;
end;
```

```
miw=0.01; % taxa de aprendizado para o parametro w
mib=0.01; % taxa de aprendizado para o parametro b
mic=0.1; % taxa de aprendizado para o parametro c
```

% Função aprend.m

```
% utilizada no aprendizado com FBR's gaussianas
```

```
erro=[];eqs=0;
for j=1:maxiter
    saida;
    erro=[erro eqs];
    if eqs>eqsmaxd
        cdeltaw;
        cdeltab;
        cdeltac;
        w=w+deltaw;
        c=c+deltac;
        b=abs(b+deltab);
    else
        break;
    end
end
saida
figure(1);plot(xd,yd,xd,y);grid
figure(2);plot(erro);
figure(3);plot(xd,yd-y,'+');
```

% Função saida.m

```
% Cálculo da Saída da Rede
```

```
for i=1:m
    g(i,1:n)=vgauss(xd,a(i),b(i),c(i));
end;
```

```
y=w*g;  
eqs=sumsq(y-yd);
```

```
% Função saidaq.m  
% Cálculo da Saída da Rede após a  
% quantização dos parametros
```

```
quantizap;  
for i=1:m  
    gq(i,1:n)=vgauss(xd,aq(i),bq(i),cq(i));  
end;  
yq=wq*gq;  
eqsq=sumsq(yq-yd);
```

Apêndice C: Funções desenvolvidas no MATLAB para as redes bidimensionais

% Função aprendl2.m

% utilizada no aprendizado com FBR's gaussianas bidimensionais.

% Após o aprendizado os parametros são discretizados

% de acordo com o número de bits definido em defrbfn

```
tic
erroqs=[ ];eqsa=1e10;eqs=0;
for iter=1:maxiter
    saida2d;
    erroqs=[erroqs eqs];
    if eqs<eqsa
        cbest=c;
        wbest=w;
        bbest=b;
        end;

        if eqs>eqsmaxd
            cdelta2d;
            %cdeltab;
            %cdeltac;
            w=w+deltaw;
            c(:,1)=c(:,1)+deltacj1;c(:,2)=c(:,2)+deltacj2;
            b(:,1)=abs(b(:,1)+deltabj1);b(:,2)=abs(b(:,2)+deltabj2);
        else
            break;
        end
    end
end
eqsa=eqs;

c=cbest;
w=wbest;
b=bbest;

saida2d;
figure(4);mesh(x1dd,x2dd,yd);
figure(5);mesh(x1dd,x2dd,y);
figure(7);plot(erroqs);
figure(6);mesh(x1dd,x2dd,erro);

saidaq2d;

figure(1);mesh(x1dd,x2dd,yd-yq);
max(max(yd-yq))
figure(9);mesh(x1dd,x2dd,yq);
figure(10);mesh(x1dd,x2dd,yd-yq);
eqs
eqsq
toc
```

% Função defrbfn2d.m

```

% definição inicial da rede de base radial gaussiana bidimensional
% definição dos parametros do algoritmo de aprendizado

m=6; % m = número de bases
dim_rbf=2; % dim = dimensão das FBRs, por ex: dim_rbf=2 para caso bidimensional
maxiter=500; % maxiter = número máximo de iterações
eqsmaxd=0.001; % eqsmaxd = erro quadrático máximo desejado
miw=0.001; % taxa de aprendizado para o parametro w
mib=0.001; % taxa de aprendizado para o parametro b
mic=0.001; % taxa de aprendizado para o parametro c

% definição dos parametros de treinamento para a rede discretizada

alfax=0.001;
betax=16;
alfaro=0.1;
betaro=6;
nbitsw=3;
nbitsc=3;
nbitsb=3;
nbitsa=8;

% definição da função a ser aprendida;

% senoide
x1d=-pi:0.2:pi; % padrão de entrada
%x1d=-2:0.1:2;
x2d=x1d; % padrão de entrada
[x1dd,x2dd] = meshgrid(x1d);

% saída desejada
yd=sin(x1dd).*sin(x2dd);
%yd=cos(x1dd).*sin(x2dd);
%yd=exp(-10*(x1dd).^2-10*(x2dd).^2)+exp(-10*(x1dd-1).^2-10*(x2dd-1).^2);

%figure(1);mesh(x1dd,x2dd,yd); %plota a função desejada

% polinomio ortogonal de hermite 6 ordem
%x1d=-3:0.1:3; % padrão de entrada
%yd=(xd.^6)-15.*(xd.^4)+45.*(xd.^2)-15;
% saída desejada normalizada entre
%yd=yd./max([max(yd) abs(min(yd))]);

n=length(x1d); % n = comprimento do vetor de entrada
min_xd=min([x1d x2d]);
max_xd=max([x1d x2d]);
dif_max_min_xd=max_xd-min_xd;

for i=1:m % Inicialização dos parametros iniciais da rede
    for j=1:dim_rbf
        ai(i)=1;
        %b(i,j)=10*rand;
        bi(i,j)=rand;
        ci(i,j)=min_xd+rand*dif_max_min_xd;
        %w(i)=-1+2*rand;
        wi(i)=rand;
    end;
end;

```

```

end;
a=ai;b=bi;c=ci;w=wi;

saida2d;yi=y;figure(2);mesh(x1dd,x2dd,yi);
%t=1; % t= variação aleatória (temperatura virtual no simulated annealing)

```

% função quantizap2d.m

```

% discretiza os parametros das FBRs
% esta função chama a função qprat
% para efetuar a discretização.

```

```

aq=quantiza(a,nbitsa);
bqj1=quantiza(b(:,1),nbitsb);bqj2=quantiza(b(:,2),nbitsb);
cqj1=quantiza(c(:,1),nbitsc);cqj2=quantiza(c(:,2),nbitsc);
wq=quantiza(w,nbitsw);

```

```

bq=[bqj1 bqj2];cq=[cqj1 cqj2];

```

% Função saida2d.m

```

% Cálculo da Saída da Rede

```

```

quantizap2d;
gq=zeros(m,n);
yq=[];
for k=1:n;
    for i=1:m
        g1q=-bq(i,1).*(x1dd(k,:)-cq(i,1).*ones(size(x1dd(k,:))))).^2;
        g2q=-bq(i,2).*(x2dd(k,:)-cq(i,2).*ones(size(x2dd(k,:))))).^2;
        gq(i,:)=exp(g1q+g2q);
    end;
yq=[yq;wq*gq];
end;
erroq=(yq-yd);
eqsq=sumsq(erroq);
%figure(2);mesh(x1dd,x2dd,y);

```

Apêndice D: Funções desenvolvidas no MATLAB para as redes multidimensionais

Função aprend.m

% utilizada no aprendizado com FBR's gaussianas

```
erro=[];eqs=0;
for j=1:maxiter
    saida;
    erro=[erro eqs];
    if eqs>eqsmaxd
        cdeltaw;
        cdeltab;
        cdeltac;
        w=w+deltaw;
        c=c+deltac;
        b=abs(b+deltab);
    else
        break;
    end
end
saida
figure(1);plot(xd,yd,xd,y);grid
figure(2);plot(erro);
figure(3);plot(xd,yd-y,'+');
```

Função aprendl2.m

% utilizada no aprendizado com FBR's gaussianas bidimensionais.

% Após o aprendizado os parametros são discretizados

% de acordo com o número de bits definido em defrbfn

```
tic
maxtentativa=1;

%for tentativa=1:maxtentativa

erroqs=[];eqsa=1e10;eqs=0;
for iter=1:maxiter
    saida2d;
    erroqs=[erroqs eqs];
    if eqs<eqsa
        cbest=c;
        wbest=w;
        bbest=b;
        end;

        if eqs>eqsmaxd
            cdelta2d;
            %cdeltab;
            %cdeltac;
            w=w+deltaw;
            c(:,1)=c(:,1)+deltacj1;c(:,2)=c(:,2)+deltacj2;
```

```

                b(:,1)=abs(b(:,1)+deltabj1);b(:,2)=abs(b(:,2)+deltabj2);
            else
                break;
            end
        end
    end
    eqsa=eqs;
    %tentativa
    %end;

    c=cbest
    w=wbest
    b=bbest

    saida2d;
    figure(4);mesh(x1dd,x2dd,yd);
    figure(5);mesh(x1dd,x2dd,y);
    %xlabel('x');ylabel('y');
    figure(7);plot(erroqs);
    figure(6);mesh(x1dd,x2dd,erro);

    saidaq2d;
    %figure(7);plot(eqsq);
    %figure(8);plot(xd,yd,xd,yd,yq,'+');

    %erro entre a aprox quantizada e nao quantizada
    figure(1);mesh(x1dd,x2dd,yd-yq);
    max(max(yd-yq))

    figure(9);mesh(x1dd,x2dd,yq);
    figure(10);mesh(x1dd,x2dd,yd-yq);
    %pause(0.5);
    %end;
    eqs
    eqsq
    toc

```

Função aprendl2.m

```

% utilizada no aprendizado com FBR's gaussianas multidimensionais.
% Após o aprendizado os parametros são discretizados
% de acordo com o número de bits definido em defrbfmd

```

```

tic
maxtentativa=1;

%for tentativa=1:maxtentativa

erroqs=[ ];eqsa=1e10;eqs=0;
for iter=1:maxiter
    saidamd;figure(2);plot(xd,y,xd,yd,'o');%pause(0.2);
    erroqs=[erroqs eqs];
    if eqs<eqsa
        cbest=c;
        wbest=w;
        bbest=b;
    end;

    if eqs>eqsmaxd

```



```

        cdeltamd; % chama a subrotina cdeltamd

        w=w+deltaw;
        c=c+deltac;
        b=abs(b+deltab);

    else
        break;
    end
end
eqsa=eqs;
%tentativa
%end;

c=cbest
w=wbest
b=bbest

%=====
% necessario modificar a forma de apresentar as saidas
figure(1);plot(erros);

saidamd;
%saidamd;
eqs
%eqsq
toc

```

defrbfnmd.m

```

% definição inicial da rede de base radial gaussiana multidimensional
% definição dos parametros do algoritmo de aprendizado
% definição dos pares exemplo para o algoritmo de aprendizado

m=6; % m = número de bases

dim_rbfm=1; % dim_rbfm = dimensão das FBRs, por ex: dim_rbfm=2 para caso bidimensional
dim_saida=2; % dim_saida = dimensão da saída da rede

maxiter=10000; % maxiter = número máximo de iterações
eqsmaxd=0.001; % eqsmaxd = erro quadrático máximo desejado
miw=0.001; % taxa de aprendizado para o parametro w
mib=0.001; % taxa de aprendizado para o parametro b
mic=0.001; % taxa de aprendizado para o parametro c

% definição dos parametros de treinamento para a rede discretizada

alfax=0.001;
betax=16;
alfaro=0.1;
betaro=6;
nbitsw=8; % numero de bits de quantização
nbitsc=8;
nbitsb=8;
nbitsa=8;

% definição da função a ser aprendida;
% saída desejada : (senoide)

```

```

xd=-pi:0.5:pi;
n=length(xd); % n = número de padroes de entrada

for k=1:dim_saida

    for p=1:n
        yd(1,p)=sin(xd(p));
        yd(2,p)=cos(xd(p));
    end;

end;

min_xd=min(min(xd)); % usado para inicializar os centros
max_xd=max(max(xd)); % usado para inicializar os centros
dif_max_min_xd=max_xd-min_xd; % usado para inicializar os centros

for i=1:m % Inicialização dos parametros iniciais da rede
    for j=1:dim_rbfm
        ai(i)=1;
        bi(i,j)=rand;
        ci(i,j)=min_xd+rand*dif_max_min_xd;
    end;
    for k=1:dim_saida
        %w(i)=-1+2*rand;
        wi(i,k)=rand; %*max(max(yd));
    end;
end;

a=ai;b=bi;c=ci;w=wi;

```

função quantizapmd.m

```

% discretiza os parametros das FBRs multidimensionais
% esta função chama a função quantiza para efetuar a discretização.

```

```

aq=quantiza(a,nbitsa);
bq=quantiza(b,nbitsb);
cq=quantiza(c,nbitsc);
wq=quantiza(w,nbitsw);

bq=[bqj1 bqj2];cq=[cqj1 cqj2];

```

função quantiza.m

```

% discretiza os parametros das FBRs
% esta função chama a função qprat
% para efetuar a discretização.

```

```

function pq = quantiza(p,nbitsp)

% definição dos valores min e max de cada parametro
% definição do número de níveis de quantização
% de acordo com o número de bits de quantização

pmin=min(p);

```

```
pmax=max(p);  
np=(2^nbitsp)-1; % calcula o numero de niveis de quantizaçã  
pq=qprat(p,pmin,pmax,np); % calcula o valor quantizado do parametro p
```

Apêndice E: Netlists (T-Spice)

Circuito RBF2 célula básica

* SPICE netlist written by S-Edit Win32 10.10

* Written on Feb 24, 2009 at 17:22:47

* Waveform probing commands

* No Ports in cell: .instal_directory

.option search= C:/DK_C35_11x_v5

* End of module with no ports: .instal_directory

* No Ports in cell: .typical_mean

.include TECH_C35B4/tm_mod.lib

* End of module with no ports: .typical_mean

.SUBCKT Module3_RBF2_basico Ib2 Io Vb1 Vb2 VDD Vi VSS

M1 N3 Vi N2 VSS MODN W='4e-6' L='1e-6' M=1

M2 N7 Vb1 N2 VSS MODN W='4e-6' L='1e-6' M=1

M3 Ib2 Ib2 VSS VSS MODN W='4e-6' L='1e-6' M=1

M4 N2 Ib2 VSS VSS MODN W='4e-6' L='1e-6' M=1

M5 N3 Vb2 N5 VSS MODN W='4e-6' L='1e-6' M=1

M6 N7 Vi N5 VSS MODN W='4e-6' L='1e-6' M=1

M7 N8 N8 VSS VSS MODN W='4e-6' L='1e-6' M=1

M8 N5 Ib2 VSS VSS MODN W='4e-6' L='1e-6' M=1

M9 Io N8 VSS VSS MODN W='4e-6' L='1e-6' M=1

M10 N8 N3 VDD VDD MODP W='4e-6' L='1e-6' M=1

M11 N3 N3 VDD VDD MODP W='4e-6' L='1e-6' M=1

M12 Io N7 VDD VDD MODP W='4e-6' L='1e-6' M=1

M13 N7 N7 VDD VDD MODP W='4e-6' L='1e-6' M=1

.ENDS

* Main circuit: Module3 - Test bench RBF2 basico

XModule3_RBF2_basico_1 N7 Vo1 Vb1 Vb2 Vdd Vi VSS Module3_RBF2_basico

R1 Vo1 Gnd 1 TC=0.0, 0.0

i2 Vdd N7 5uA

v3 Vb1 Gnd 100m

v4 Vb2 Gnd -100m

v5 Vc Gnd 0

v6 Vdd Gnd 1.5

v7 Vi Gnd 0

v8 Gnd VSS 1.5

* End of main circuit: Module3 - Test bench RBF2 basico

.dc lin source V7 -1500m 1500m 10m sweep lin source i2 1u 10u 1u

*sweep lin source i3 10u 250u 10u

.probe

.options probefilename="RBF2.dat"

+ probesdbfile="C:\Arquivos de programas\Tanner EDA\trabalho\RBF2\RBF2.sdb"

+ probetopmodule="Module3 - Test bench RBF2 basico"

Análise Monte Carlo

```

* SPICE netlist written by S-Edit Win32 10.10
* Written on Mar 1, 2009 at 17:21:27

* Waveform probing commands
.probe
.options probefilename="C:\Arquivos de programas\Tanner EDA\trabalho\RBF2\RBF2.dat"
+ probesdbfile="C:\Arquivos de programas\Tanner EDA\trabalho\RBF2\RBF2.sdb"
+ probetopmodule="Monte Carlo RBF2 basico unidimensional"

* No Ports in cell: .instal_directory
.option search= C:/DK_C35_11x_v5
* End of module with no ports: .instal_directory

* No Ports in cell: .typical_mean
.include TECH_C35B4/tm_mod.lib
.param comp=unif('1u',0.05,1) larg=unif('4u',0.05,1)
.param vtn=unif('4.979e-01',0.05,1) vtp=unif('-6.915e-01',0.05,1)
.param U0n=unif('4.758e+02',0.05,1) U0p=unif('1.482e+02',0.05,1)
.model MODN2 nmos VTH0=vtn U0=U0n ako:MODN
.model MODP2 pmos VTH0=vtp U0=U0p ako:MODP
* End of module with no ports: .typical_mean

* Main circuit: Monte Carlo RBF2 basico unidimensional
M1 N6 N3 N2 N1 MODN2 W=larg L=comp M=1
M2 N7 Vb1 N2 N1 MODN2 W=larg L=comp M=1
M3 N5 N5 N1 N1 MODN2 W=larg L=comp M=1
M4 N2 N5 N1 N1 MODN2 W=larg L=comp M=1
M5 N6 Vb2 N4 N1 MODN2 W=larg L=comp M=1
M6 N7 N3 N4 N1 MODN2 W=larg L=comp M=1
M7 N8 N8 N1 N1 MODN2 W=larg L=comp M=1
M8 N4 N5 N1 N1 MODN2 W=larg L=comp M=1
M9 Vout N8 N1 N1 MODN2 W=larg L=comp M=1
M10 N8 N6 Vdd Vdd MODP2 W=larg L=comp M=1
M11 N6 N6 Vdd Vdd MODP2 W=larg L=comp M=1
M12 Vout N7 Vdd Vdd MODP2 W=larg L=comp M=1
M13 N7 N7 Vdd Vdd MODP W=larg L=comp M=1
R14 Vout Gnd 1 TC=0.0, 0.0
i15 Vdd N5 50uA
v16 Vdd Gnd 1.5
v17 Gnd N1 1.5
v18 Vb1 Gnd 100m
v19 Vb2 Gnd -100m
v20 N3 Gnd 0
* End of main circuit: Monte Carlo RBF2 basico unidimensional

* Analise Monte Carlo
.dc V20 -1500m 1500m 10m sweep monte=50

* Gera saída no formato txt

*.print dc "c:/simula3.txt" v(v5) v(v8) v(Vo)

.print dc "c:/simula_RBF2_uni.txt" v(Vout) i(Vout)

```

Circuito RBF2 Bidimensional

* SPICE netlist written by S-Edit Win32 10.10

* Written on Feb 25, 2009 at 08:32:31

* Waveform probing commands

.probe

.options probefilename="C:\Arquivos de programas\Tanner EDA\trabalho\RBF2\RBF2.dat"

+ probesdbfile="C:\Arquivos de programas\Tanner EDA\trabalho\RBF2\RBF2.sdb"

+ probetopmodule="Module3 - Test bench RBF2 basico"

.SUBCKT Module3_RBF2_basico Ib2 Io Vb1 Vb2 VDD Vi VSS

M1 N3 Vi N2 VSS MODN W='4e-6' L='1e-6' M=1

M2 N7 Vb1 N2 VSS MODN W='4e-6' L='1e-6' M=1

M3 Ib2 Ib2 VSS VSS MODN W='4e-6' L='1e-6' M=1

M4 N2 Ib2 VSS VSS MODN W='4e-6' L='1e-6' M=1

M5 N3 Vb2 N5 VSS MODN W='4e-6' L='1e-6' M=1

M6 N7 Vi N5 VSS MODN W='4e-6' L='1e-6' M=1

M7 N8 N8 VSS VSS MODN W='4e-6' L='1e-6' M=1

M8 N5 Ib2 VSS VSS MODN W='4e-6' L='1e-6' M=1

M9 Io N8 VSS VSS MODN W='4e-6' L='1e-6' M=1

M10 N8 N3 VDD VDD MODP W='4e-6' L='1e-6' M=1

M11 N3 N3 VDD VDD MODP W='4e-6' L='1e-6' M=1

M12 Io N7 VDD VDD MODP W='4e-6' L='1e-6' M=1

M13 N7 N7 VDD VDD MODP W='4e-6' L='1e-6' M=1

.ENDS

* No Ports in cell: .instal_directory

.option search= C:/DK_C35_11x_v5

* End of module with no ports: .instal_directory

* No Ports in cell: .typical_mean

.include TECH_C35B4/tm_mod.lib

* End of module with no ports: .typical_mean

* Main circuit: Module3 - Test bench RBF2 basico

XModule3_RBF2_basico_1 N7 Vo1 Vb1 Vb2 Vdd Vi VSS Module3_RBF2_basico

XModule3_RBF2_basico_2 N1 N10 Vb1_01 Vb2_01 Vdd Vi1 VSS Module3_RBF2_basico

XModule3_RBF2_basico_3 N10 Vo Vb1_02 Vb2_02 Vdd Vi2 VSS Module3_RBF2_basico

R1 Vo1 Gnd 100k TC=0.0, 0.0

R2 Vo Gnd 1 TC=0.0, 0.0

i3 Vdd N7 5uA

i4 Vdd N1 5uA

v5 Vi1 Gnd 0

v6 Vb2_01 Gnd -100m

v7 Vb1_01 Gnd 100m

v8 Vi2 Gnd 0

v9 Vb2_02 Gnd -100m

v10 Vb1_02 Gnd 100m

v11 Vb1 Gnd 100m

v12 Vb2 Gnd -100m

v13 Vc Gnd 0

v14 Vdd Gnd 1.5

v15 Vi Gnd 0

v16 Gnd VSS 1.5

* End of main circuit: Module3 - Test bench RBF2 basico

.dc lin source V5 -1500m 1500m 20m sweep lin source v8 -1500m 1500m 20m

.print dc "c:/simula.txt" v(v5) v(v8) v(Vo)

RBFN RBF2 Bidimensional

* SPICE netlist written by S-Edit Win32 10.10

* Written on Feb 27, 2009 at 23:07:16

* Waveform probing commands

.probe

.options probefilename="RBFN_RBF2.dat"

+ probesdbfile="C:\Arquivos de programas\Tanner EDA\trabalho\RBFN_RBF2\RBFN_RBF2.sdb"

+ probetopmodule="Module0"

* No Ports in cell: .instal_directory

.option search= C:/DK_C35_11x_v5

* End of module with no ports: .instal_directory

* No Ports in cell: .typical_mean

.include TECH_C35B4/tm_mod.lib

* End of module with no ports: .typical_mean

.SUBCKT Module3_RBF2_basico Ib Io Vb1 Vb2 VDD Vi VSS

M1 N38 Vi N8 VSS MODN W='4e-6' L='1e-6' M=1

M2 N22 Vb1 N8 VSS MODN W='4e-6' L='1e-6' M=1

M3 Ib Ib VSS VSS MODN W='4e-6' L='1e-6' M=1

M4 N8 Ib VSS VSS MODN W='4e-6' L='1e-6' M=1

M5 N38 Vb2 N30 VSS MODN W='4e-6' L='1e-6' M=1

M6 N22 Vi N30 VSS MODN W='4e-6' L='1e-6' M=1

M7 N35 N35 VSS VSS MODN W='4e-6' L='1e-6' M=1

M8 N30 Ib VSS VSS MODN W='4e-6' L='1e-6' M=1

M9 Io N35 VSS VSS MODN W='4e-6' L='1e-6' M=1

M10 N35 N38 VDD VDD MODP W='4e-6' L='1e-6' M=1

M11 N38 N38 VDD VDD MODP W='4e-6' L='1e-6' M=1

M12 Io N22 VDD VDD MODP W='4e-6' L='1e-6' M=1

M13 N22 N22 VDD VDD MODP W='4e-6' L='1e-6' M=1

.ENDS

.SUBCKT RBF2_bidimensional Ib Io Vb1 Vb2 Vb3 Vb4 VDD Vi1 Vi2 VSS

XModule3_RBF2_basico_1 Ib N14 Vb1 Vb2 VDD Vi1 VSS Module3_RBF2_basico

XModule3_RBF2_basico_2 N14 Io Vb3 Vb4 VDD Vi2 VSS Module3_RBF2_basico

.ENDS

* Main circuit: Module0

i1 Vdd N4 5uA

i2 Vdd N6 5uA

i3 Vdd N2 5uA

i4 Vdd N5 5uA

i5 Vdd N1 5uA

XRBF2_bidimensional_1 N4 Vo Vb1_01 Vb2_01 Vb3_01 Vb4_01 Vdd Vi1 Vi2 VSS

+ RBF2_bidimensional

XRBF2_bidimensional_2 N6 Vo Vb1_02 Vb2_02 Vb3_02 Vb4_02 Vdd Vi1 Vi2 VSS

+ RBF2_bidimensional

XRBF2_bidimensional_3 N5 Vo Vb1_04 Vb2_04 Vb3_04 Vb4_04 Vdd Vi1 Vi2 VSS

+ RBF2_bidimensional

XRBF2_bidimensional_4 N1 Vo Vb1_05 Vb2_05 Vb3_05 Vb4_05 Vdd Vi1 Vi2 VSS

+ RBF2_bidimensional

XRBF2_bidimensional_5 N2 Vo Vb1_03 Vb2_03 Vb3_03 Vb4_03 Vdd Vi1 Vi2 VSS

+ RBF2_bidimensional

R6 Vo Gnd 1 TC=0.0, 0.0

v7 Vdd Gnd 1.5

v8 Gnd VSS 1.5

v9 Vi1 Gnd 0

```
v10 Vi2 Gnd 0
v11 Vb1_01 Gnd 100m
v12 Vb2_01 Gnd -100m
v13 Vb3_01 Gnd 100m
v14 Vb4_01 Gnd -100m
v15 Vb1_02 Gnd 600m
v16 Vb2_02 Gnd 400m
v17 Vb3_02 Gnd 600m
v18 Vb4_02 Gnd 400m
v19 Vb1_03 Gnd -400m
v20 Vb2_03 Gnd -600m
v21 Vb3_03 Gnd -400m
v22 Vb4_03 Gnd -600m
v23 Vb1_04 Gnd 600m
v24 Vb2_04 Gnd 400m
v25 Vb3_04 Gnd -400m
v26 Vb4_04 Gnd -600m
v27 Vb4_05 Gnd 400m
v28 Vb3_05 Gnd 600m
v29 Vb2_05 Gnd -600m
v30 Vb1_05 Gnd -400m
* End of main circuit: Module0

.dc lin source V9 -1500m 1500m 20m sweep lin source v10 -1500m 1500m 20m

.print dc "c:/simula_RBFN_RBF2.txt" v(v9) v(v10) v(Vo)
```


Anexos

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)