

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

José de Ribamar Silva Fonseca

*Análise do Efeito da Precisão Finita no Algoritmo Adaptativo
Sigmoidal*

São Luís
2017

José de Ribamar Silva Fonseca

*Análise do Efeito da Precisão Finita no Algoritmo Adaptativo
Sigmoidal*

Dissertação apresentada ao Programa de Pós-Graduação
em Engenharia Elétrica da UFMA, como requisito
para a obtenção parcial do grau de MESTRE em
Engenharia Elétrica.

Orientador: Prof. Dr. Allan Kardec Duailibe Barros Filho

Departamento de Engenharia Elétrica - UFMA

Co-orientador: Prof. Dr. André Borges Cavalcante

Departamento de Engenharia Elétrica - UFMA

São Luís

2017

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Fonseca, José de Ribamar Silva.

Análise do Efeito da Precisão Finita no Algoritmo
Adaptativo Sigmoidal / José de Ribamar Silva Fonseca. -
2017.

64 p.

Coorientador(a): André Borges Cavalcante.

Orientador(a): Allan Kardec Duailibe Barros Filho.

Dissertação (Mestrado) - Programa de Pós-graduação em
Engenharia de Eletricidade/ccet, Universidade Federal do
Maranhão, PIB3, 2017.

1. Aritmética de Ponto Fixo. 2. Filtros Adaptativos.
3. Quantização. 4. Sigmoidal. I. Barros Filho, Allan
Kardec Duailibe. II. Cavalcante, André Borges. III.
Título.

José de Ribamar Silva Fonseca

*Análise do Efeito da Precisão Finita no Algoritmo
Adaptativo Sigmoidal*

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFMA, como requisito para a obtenção parcial do grau de MESTRE em Engenharia Elétrica.

Aprovado em 16 de Fevereiro de 2017.

BANCA EXAMINADORA

Prof. Dr. Allan Kardec Duailibe Barros Filho
Departamento de Engenharia Elétrica –UFMA

Prof. Dr. André Borges Cavalcante
Departamento de Engenharia Elétrica –UFMA

Prof. Dr. Francisco das Chagas de Souza
Departamento de Engenharia Elétrica –UFMA

Prof. Dr. Fausto Lucena de Oliveira

Prof. Dr. Marcos Antonio Ferreira de Araújo
Departamento de Matemática -UFMA

*À minha esposa Raquel, à minha filha Cecília
e ao meu filho Caio.*

Resumo

A filtragem adaptativa constitui atualmente uma ferramenta importante no processamento estatístico de sinais, especialmente quando é necessário processar sinais provenientes de ambientes com estatísticas desconhecidas que variam com o tempo. O estudo de filtragem adaptativa foi impulsionado com o desenvolvimento do algoritmo Least Mean Square (LMS) em 1960. Desde então outros algoritmos adaptativos têm surgido com um desempenho superior ao algoritmo LMS em relação ao desajuste e à taxa de convergência. Entre eles, o algoritmo Sigmoidal (SA) que se apresentou superior ao LMS, em relação a taxa de convergência e o desajuste em suas implementações na forma analógica. Nos dispositivos de hardware, tais como DSPs, Microcontroladores e FPGAs, os algoritmos adaptativos são implementados na forma digital, onde a precisão é finita, em geral, com aritmética de ponto fixo. Quando os filtros adaptativos são implementados em precisão finita alguns efeitos podem afetar o seu desempenho. Em última análise, levar à divergência devido aos erros de quantização especificados no processo de aproximação dos valores das variáveis envolvidas no processamento adaptativo de seus valores originais. Assim, este trabalho propõe analisar o desempenho do algoritmo adaptativo Sigmoidal (SA) em precisão finita, quando implementado utilizando aritmética de ponto fixo. Em particular, a análise de sua curva de desempenho e o desajuste, comparando-os em diferentes comprimentos de palavras (número de bits). Os resultados apresentados neste trabalho propõe uma aproximação em série de Taylor do gradiente da função de custo $\text{Ln}(\cosh \alpha e)$ do algoritmo SA para implementação em precisão finita. Analisamos a sua curva de desempenho para diferentes comprimentos de palavras. Mostra-se que o algoritmo apresenta estabilidade em seu desempenho em relação à convergência, para diferentes comprimentos de palavras, e que o aumento no nível do desajuste em estado estacionário é sensível ou influenciado pela quantização dos valores das variáveis envolvidas nos cálculos desse algoritmo.

Palavras-chaves: Filtros Adaptativos, Quantização, Aritmética de Ponto-Fixo, Sigmoidal.

Abstract

The adaptive filtering is currently an important tool in the statistical processing of signals, especially when it is necessary to process signals from environments with unknown statistics varying with time. The adaptive filtering study was driven by the development of the Least Mean Square algorithm (LMS) in 1960. Since then other adaptive algorithms have come up with a better performance than LMS algorithm with respect to misadjustment and convergence rate. Among them, the Sigmoidal algorithm (SA) which showed superior to the LMS, for the convergence rate and the mismatch in their implementations infinite precision. In hardware devices such as DSPs, microcontrollers and FPGAs, adaptive algorithms are implemented in finite precision, in general, fixed point arithmetic. When the adaptive filters are implemented in finite precision some effects can affect their performance. Ultimately lead to divergence due to quantization errors specified in the approximation process of the variables involved in the adaptive processing of their original values. Thus, this article aims to analyze the performance of the adaptive algorithm Sigmoidal (SA) in finite precision when implemented using fixed-point arithmetic. In particular, the analysis of its performance curve and mismatch, comparing them in different word lengths (number of bits). The results presented in this article proposes a series of Taylor Ln gradient of cost function ($\cosh \alpha e$) algorithm SA for implementation in finite precision. We analyze its performance curve for different lengths of words. It shows that the algorithm is stable in its performance compared to convergence to different lengths of words, and that the increase in mismatch level at steady state is sensitive or affected by the quantization of the variables involved in the calculations of this algorithm.

Keywords: Adaptive Filters, Quantization, Fixed-Point Arithmetic, Sigmoidal

Agradecimentos

Ao professor Allan Kardec Duailibe Barros Filho, pela confiança, orientação e dedicação com que encaminhou este trabalho.

Ao professor André Borges Cavalcante e ao professor Ewaldo Eder Carvalho Santana, pela empenho e disponibilidade na orientação deste trabalho.

Agradeço aos professores e amigos do Laboratório da Informação Biológica-PIB.

À toda minha família e amigos.

À CNPQ, pelo relevante apoio financeiro.

“Vencer a si próprio é a maior das vitórias”.

Platão

Sumário

Lista de Figuras	8
Lista de Tabelas	11
Lista de Abreviações	12
Artigos publicados, aceitos e apresentados	13
1 Introdução	14
1.1 Problema	15
1.2 Objetivos	15
1.3 Motivações	15
1.4 Estado da Arte	16
1.5 Organização do Texto	16
2 Filtragem Adaptativa	18
2.1 Características dos Filtros Adaptativos	19
2.2 Aplicações	22
3 Efeitos da Precisão Finita	24
3.1 Erros de Quantização	26
3.2 Filtragem Adaptativa em Precisão Finita	27
3.2.1 O LMS em Precisão Finita	29
3.3 Implementação em Dispositivo de Hardware Digital	31
3.4 Simulação de Monte Carlo	32

4	Análise em Precisão Finita do Algoritmo Adaptativo Baseado no Erro Não Linear	33
4.1	Análise de Adaptação	36
4.2	Comportamento de Convergência	40
5	O Algoritmo Sigmoidal em Precisão Finita	43
5.1	Método	44
5.2	Resultados	46
6	Discussão	53
7	Conclusões Finais e Perspectivas	57
	Referências	58
	Apêndices	61

Lista de Figuras

2.1	Configuração geral de um filtro adaptativo. Em que k representa o número de iteração, $\mathbf{x}_k$ designa o sinal de entrada, y_k é o sinal de saída do filtro adaptativo, e d_k define o sinal desejado. O sinal de erro e_k é calculada como $d_k - y_k$. O sinal de erro é então utilizado para formar uma função de desempenho (ou objetivo) que é exigida pelo algoritmo de adaptação para determinar a atualização apropriada dos coeficientes do filtro $\mathbf{w}_k$	19
2.2	Modelo de um filtro adaptativo discreto aplicado ao problema de identificação de sistema desconhecido, onde z^I representa as unidades do sinal de entrada $\mathbf{x}_k$ atrasadas.	20
2.3	Porção da superfície quadrática tridimensional juntamente com alguns contornos. O erro quadrático médio está plotado na vertical, w_0 e w_1 variam de -1 a 1.	21
2.4	Modelo de identificação de sistema	22
2.5	Modelo de Modelagem Inversa	23
2.6	Modelo de Predição Linear	23
2.7	Modelo de Cancelamento de Ruído	23
3.1	Representação em precisão infinita. O Sinal assume valores infinitos no intervalo considerado. Intervalo de amplitude $[0,L]$	24
3.2	Representação em precisão finita. O Sinal assume valores finitos no intervalo. O erro é diferente de zero. Intervalo de amplitude $[0,L]$	25
3.3	Esquema geral da aritmética de ponto fixo. Considerando comprimento de palavra $b + 1bit$ de sinal. Define número de b^n parte interia e b^{-n} parte fracionária.	26
3.4	Representação Níveis de Quantização. a) Quantidades de níveis $N=4$. b)Quantidades de níveis $N=8$	26

3.5	Representação esquemática de níveis de quantização para o caso de $b = 2$ bits e $\Delta = L/4$. Consideramos um valor L limiar. Usando b bits, o intervalo de $[0, L]$ pode ser dividido em 2^b níveis de largura $\Delta = L/2^b$ cada. Se, além disso, usamos 1 (um) bit de sinal, os resultantes $(b + 1)$ bits divide o intervalo $[-L, L]$ em 2^{2b} níveis. Aqui assumimos operação de arredondamento.	27
3.6	Variância devido à quantização com aritmética de ponto fixo. Para $C = 1$ número de <i>bits</i> variando entre <i>7bits</i> à <i>32bits</i>	28
4.1	Filtragem adaptativa em identificação de sistema.	34
4.2	Identificação de sistema implementado em precisão finita. Onde $\mathbf{x}_k$ designa o sinal de entrada. Os blocos Q_d e Q_c são quantizadores de b bits. Assim, o quantizador, Q_d , usa b_d bits para os dados de entrada e Q_c quantiza os coeficientes em b_c bits. O sinal de saída y_{Qk} indica a forma quantizada. O sinal desejado definido por d_k do sistema desconhecido e d_{Qk} sua forma quantizada. O ruído de medição n_k e o sinal de erro quantizado e_{Qk} . Os coeficientes $\mathbf{w}_k$ do sistema desconhecido e $\mathbf{w}_{Qk}$ do filtro adaptativo.	37
5.1	Porção da superfície tridimensional gerada pela função $\ln \cosh(\alpha e)$, juntamente com alguns contornos.	43
5.2	Diagrama do fluxo do projeto SA em precisão finita	45
5.3	Comparação da função $\tanh(e)$ e sua função em série de Taylor(5º grau) em ponto flutuante com 64 bits.	47
5.4	SA x LMS em ponto fixo 32 bits.	47
5.5	SA em diferente comprimento de palavra binária em aritmética de ponto fixo: 8 bits, 16 bits e 32 bits.	48
5.6	Curva de aprendizagem SA em precisão finita para $C = 0,0590$ e $n_k = 10^{-3}$ em diferentes comprimentos de palavras.	49
5.7	Curva de aprendizagem SA em precisão finita para $C = 0,0590$ em diferentes comprimentos de palavras	50

5.8	Curva de aprendizagem SA em precisão finita para $b_d = 15bits$, $n_k = 10^{-3}$ e $C = 0,0590$ em diferentes comprimentos de palavras	51
-----	--	----

Lista de Tabelas

5.1	Desajuste e MSE do SA em precisão finita	47
5.2	Desajuste e MSE do SA em precisão finita para $C = 0,0590$ e $n_k = 10^{-3}$.	49
5.3	Desajuste e MSE do SA em precisão finita para $b_d = 15bits$, $n_k = 10^{-2}$ e $C = 0,0590$	50
5.4	Desajuste e MSE do SA em precisão finita para $b_d = 15bits$, $n_k = 10^{-3}$ e $C = 0,0590$	52

Lista de Abreviações

[ADC]	Conversor Analógico Digital
[DSP]	Digital Signal Processing
[FIR]	Finite Impulse Response
[FPGA]	Field Programmable Gate Array
[IRR]	Infinite Impulse Response
[LMS]	Least Mean Square
[MSE]	Erro Quadrático Médio
[RLS]	Recursive Least Square
[SA]	Sigmoidal
[TLC]	Teorema do Limite Central

Artigos publicados, aceitos e apresentados

1. José de R.S. Fonseca, Ewaldo E. C. Santana, André C. B. Cavalcante, Raimundo C. S. Freire and Allan K. Barros. "FINITE PRECISION EFFECT OF ADAPTIVE ALGORITHM SIGMOIDAL"Your paper 1570325509. [AMC2016]-Kota Kinabalu, Sabah -Malaysia. <http://uksim.info/amc2016/amc2016>. Ano: 2016

2. José de R.S. Fonseca, Ewaldo E. C. Santana, André C. B. Cavalcante, Raimundo C. S. Freire and Allan K. Barros. "ANÁLISE DO EFEITO DA PRECISÃO FINITA NO ALGORITMO ADAPTATIVO SIGMOIDAL"X-EAMC- Petrópolis-RJ,BRASIL. <http://www.eamc.lncc.br/> Ano: 2017

1 Introdução

O estudo de sinais teve um grande avanço em suas aplicações com o surgimento e aperfeiçoamentos dos DSPs (do inglês *Digital Signal Processing*) que são microprocessadores especializados em processamento digital de sinal. O que antes era feito de forma analógica, com pouco controle sobre os fenômenos físicos envolvidos, pode ser feito de forma digital, controlada, muito mais precisa e rápida. O conhecimento das regras que governam os sinais discretos, bem como os dispositivos que os processam, alternativamente ao processamento de sinais analógicos, apresentando diversas vantagens. No processamento de sinal analógico as quantidades físicas variam ao longo de uma faixa de valores contínuos. Para que o sinal seja processado digitalmente, necessita ser, primeiramente, discretizado. Algumas vantagens se destacam no processamento digital de sinais, como a programabilidade, estabilidade, redução de custos e facilidade de implementação de algoritmos adaptativos. Muitos dos problemas relacionados ao processamento de sinais tem solução através da filtragem adaptativa [Oppenheim, (1996)]. O termo "filtro" é geralmente aplicado a qualquer dispositivo ou sistema que processa os sinais recebidos ou outros dados, de forma a eliminar o ruído, ou suavizar os sinais, ou identificar cada um dos sinais como pertencentes a uma determinada classe, ou mesmo prever o próximo sinal de momento a momento [Widrow, (1985)].

Nos últimos anos a demanda por equipamentos portáteis, de fácil manuseio e que tenham uma interface simples com o usuário tem crescido muito. Alguns dispositivos de hardware digitais procuram unir todos esses preceitos, diversos dispositivos de hardware (eletrônicos) que utilizam processamento de sinal digital, tais como DSPs, microcontroladores e FPGAs (do inglês *Field Programmable Gate Arrays*). Estes dispositivos de hardware digitais são muito versáteis, podendo se enquadrar em quaisquer tipos de projetos [Tocci, (2013)]. Uma das grandes desvantagens da técnica digital é que a maioria das grandezas físicas no mundo real consiste de uma infinidade de sinais analógicos [Haykin, (2014)].

Ao se trabalhar com sinais ou sistemas digitais podemos representá-los em precisão infinita (sinal analógico) e em precisão finita (sinal digital). No processamento de sinais digitais realiza-se uma aproximação de uma grandeza analógica por um valor digital

(discreto), tendo como consequência um erro de aproximação ou erro de quantização [Diniz, (2013)].

1.1 Problema

Erros de quantização podem degradar o desempenho de um filtro adaptativo de várias maneiras. Por exemplo, eles podem afetar a estabilidade do filtro e, em última análise conduzir à sua divergência. Também podem degradar o desempenho em estado estacionário do filtro fazendo-o atingir um erro quadrático médio maior do que o esperado a partir de uma análise de precisão infinita [Gupta et al., (2000)].

1.2 Objetivos

- Determinar se o algoritmo Sigmoidal (SA) é estável numericamente ou não quando se utiliza precisão finita.
- Modelar analiticamente o algoritmo Sigmoidal (SA) em precisão finita.

1.3 Motivações

Em termos práticos, os algoritmos adaptativos são implementados nos dispositivos de hardware em precisão finita, e muitas vezes com aritmética de ponto fixo [Gupta and Hero, (2000)]. A penalidade no desempenho de alguns algoritmos adaptativos incorridos com o resultado da implementação em precisão finita tem sido analisadas em estudos relacionados a filtragem adaptativa. O seu comportamento pode obter resultados ou efeitos indesejáveis devido aos erros de quantização introduzidos nos cálculos envolvidos no processamento adaptativo desses filtros [Haykin, (2014)].

1.4 Estado da Arte

A degradação do desempenho tende a ser mais grave para o algoritmo RLS (*Recursive Least Square*) convencional podendo ser mais sensível a erros de precisão finita e divergir como resultado [Sayed, (2013)] em oposição ao algoritmo LMS (*Least Mean Square*) [Gupta et al., (2000)]. Isto porque para filtros do tipo LMS, erros de precisão finita em algoritmos baseados no gradiente da função de custo, são mais significativos do que os erros de precisão finita durante a fase transitória em algoritmos não baseado nesta técnica. No entanto, em estado estacionário, um significativo excesso de erro quadrático pode ser formado [Verhaegen, (1989)].

Trabalhos recentes analisam filtros adaptativos em precisão finita utilizando aritmética de ponto fixo. Em Maluenda (2005), por exemplo, apresenta nova modelagem para o algoritmo LMS. Meher (2014), traz uma arquitetura eficiente para o algoritmo LMS e deriva expressões de erros de estados. Algoritmos de filtragem adaptativa com base em erro não quadrático médio tem sido sugeridos por vários autores, quer como uma forma de melhorar o desempenho de adaptação destes algoritmos para ambientes estatísticos particulares, ou como forma de simplificar o cálculo do algoritmo em hardware [Douglas and Meng, (1994)].

1.5 Organização do Texto

Este trabalho está dividido da seguinte forma: No capítulo 2 apresentamos inicialmente os conceitos relacionado aos filtros adaptativos bem como as suas características e aplicações.

No capítulo 3 apresentamos os efeitos da precisão finita, definindo o erro de quantização, implementação em dispositivos de hardware e por último mostramos o conhecido algoritmo adaptativo LMS em precisão finita.

No capítulo 4 realizamos uma análise de uma família de algoritmos baseados em não lineareidade do erro em precisão finita, onde mostramos uma análise de adaptação e o comportamento de convergência.

No capítulo 5 implementamos o algoritmo Sigmoidal (SA) em precisão finita,

onde mostramos o método utilizado e os resultados encontrados.

No capítulo 6 realizamos as discussões dos resultados encontrados e no capítulo 7 realizamos as conclusões finais juntamente com perspectivas futuras sobre o encaminhamento deste trabalho.

2 Filtragem Adaptativa

A filtragem adaptativa tem sido fortemente utilizada como técnica para o processamento de sinais digitais, e com os avanços desta teoria e da tecnologia dos circuitos integrados torna-se cada vez mais sofisticada, [Meher, (2014)]. Filtros adaptativos são dispositivos auto-ajustáveis, baseados em algoritmos recursivos, que modificam seus parâmetros de acordo com critérios pré-estabelecidos. Em ambiente estacionário, acompanham a solução ótima do problema. Em ambiente não estacionário, acompanham as modificações do sinal envolvido [Widrow, (1985)].

Os filtros adaptativos são classificados em dois grupos principais: lineares e não lineares. Filtros adaptativos lineares computam um valor estimado da resposta desejada utilizando uma combinação linear dos sinais de entrada, de outra forma, diz-se que os filtros adaptativos são não lineares [Boroujeny, (2013)].

Um trabalho pioneiro no campo de projetos de filtros foi desenvolvido por Norbert Wiener em 1949. Seus esforços tornaram possível o projeto de filtros lineares para eliminação de ruído e para predição e suavizamento de sinais estatisticamente estacionários. Outro trabalho mais recente de Kalman e Bucy (1961) conduziu à concepção de filtros lineares de tempo variável para sinais não estacionários. Para tais sinais, filtros de Kalman-Bucy podem entregar um desempenho consideravelmente melhor do que os filtros de Wiener [Douglas and Meng, (1992)].

As aplicações típicas dos filtros adaptativos são: predição, modelamento, cancelamento de ruído, identificação de sistema, dentre outras [Diniz, (2013)]. Duas medidas se destacam na análise do desempenho dos algoritmos adaptativos: o desajuste e a taxa de convergência [Caraiscos e Liu, (1984)]. Dentre os algoritmos adaptativos o LMS é referência e é de longe o mais conhecido [(Haykin, (2014))]. Novos algoritmos adaptativos têm surgido com um desempenho superior ao LMS em relação ao desajuste e à taxa de convergência, entre esses, tem-se o algoritmo RLS [Haykin, (2014)]. Outro algoritmo, que tem como base este trabalho é o Sigmoidal (SA), onde Santana (2006) demonstrou a sua superioridade em relação ao LMS comparando o desajuste e a taxa de convergência.

2.1 Características dos Filtros Adaptativos

A configuração geral de um ambiente de filtragem adaptativa é ilustrada na Fig. 2.1.

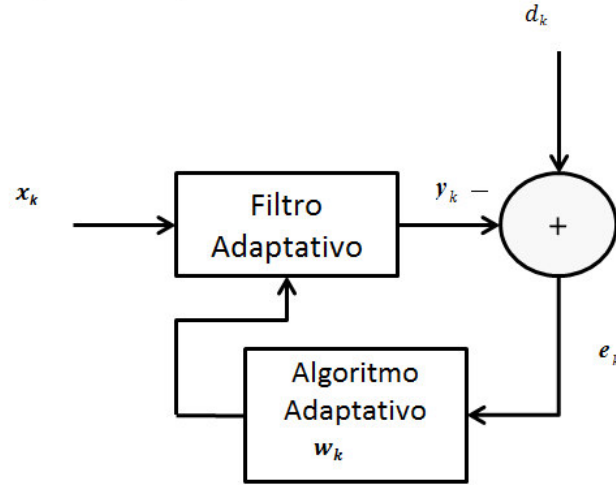


Figura 2.1: Configuração geral de um filtro adaptativo. Em que k representa o número de iteração, $\mathbf{x}_k$ designa o sinal de entrada, y_k é o sinal de saída do filtro adaptativo, e d_k define o sinal desejado. O sinal de erro e_k é calculada como $d_k - y_k$. O sinal de erro é então utilizado para formar uma função de desempenho (ou objetivo) que é exigida pelo algoritmo de adaptação para determinar a atualização apropriada dos coeficientes do filtro $\mathbf{w}_k$.

A especificação completa de um sistema adaptativo, mostrado na Fig. 2.1, consiste em três itens: Estrutura do filtro adaptativo, aplicação e algoritmo adaptativo.

A estrutura de um filtro adaptativo pode ser dada de duas formas: *Finite Impulse Response* (FIR) ou *Infinite Impulse Response* (IRR). Em nosso trabalho consideramos apenas a realização em estrutura FIR.

Filtros adaptativos podem ser contínuo ou discretos. Uma forma particular de filtro adaptativo a ser considerada é um tipo discreto (dados amostrados). O filtro discreto consiste de uma linha de atraso, pesos variáveis (ganhos variáveis) cuja entrada de sinais são os sinais na linha de entrada atrasados, uma soma para adicionar os sinais ponderados, e o mecanismo para ajustar automaticamente os pesos. A resposta de impulso de um sistema tal discreto é completamente controlada pelas configurações de peso. O processo de adaptação procura automaticamente uma resposta ao impulso do filtro ideal, ajustando os pesos. A Fig. 2.2, ilustra esquematicamente, a maioria dos componentes de

um filtro adaptativo, que é utilizado neste caso para a modelação de um sistema dinâmico desconhecido.

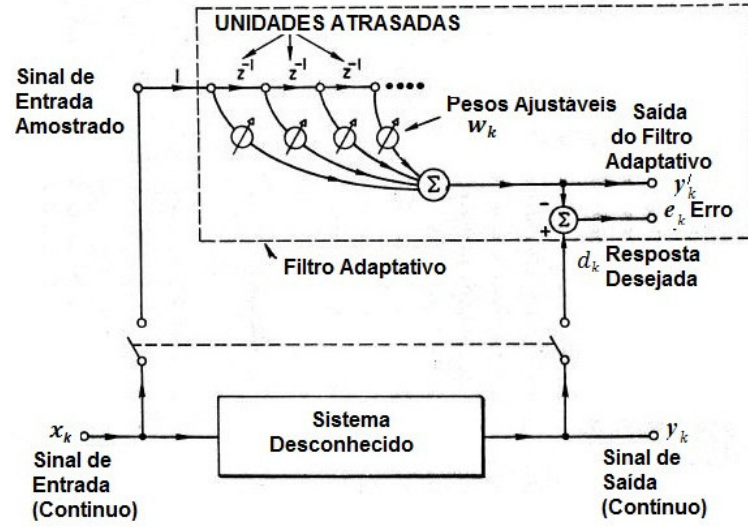


Figura 2.2: Modelo de um filtro adaptativo discreto aplicado ao problema de identificação de sistema desconhecido, onde z^I representa as unidades do sinal de entrada $\mathbf{x}_k$ atrasadas.

O sinal de saída y_k é uma combinação linear dos coeficientes do filtro $\mathbf{w}_k^t \mathbf{x}_k$, que produz um MSE (*Erro Quadrático Médio*) representado por $E[(e_k)^2]$ como função de uma solução ótima original [Diniz, (2013)].

Muitos algoritmos adaptativos utilizam-se do MSE como função de custo aplicada sobre o erro, no qual deseja-se minimizar. O MSE é uma função convexa dos componentes do vetor peso e gera uma superfície hiperparabolóide que garante a existência de um mínimo global, ver Fig. 2.3. O problema é como determinar procedimentos de forma tal a encontrar esse mínimo, o mais rápido possível e com o menor erro final [Widrow, (1985)].

Dois tipos de processos ocorrem na filtragem adaptativa: treinamento e operação. O processo de treinamento verifica a adaptação das ponderações na linha de atraso aproveitado. O processo de operação consiste em formar os sinais de saída ponderando os sinais da linha de atraso, utilizando os pesos resultantes do processo de adaptação [Sayed, (2013)].

Existe uma grande variedade de algoritmos adaptativos, a escolha de um ou outro algoritmo é determinada pelos seguintes fatores [Sayed, (2013)]:

- Taxa de convergência: definida como o número de iterações necessárias para que

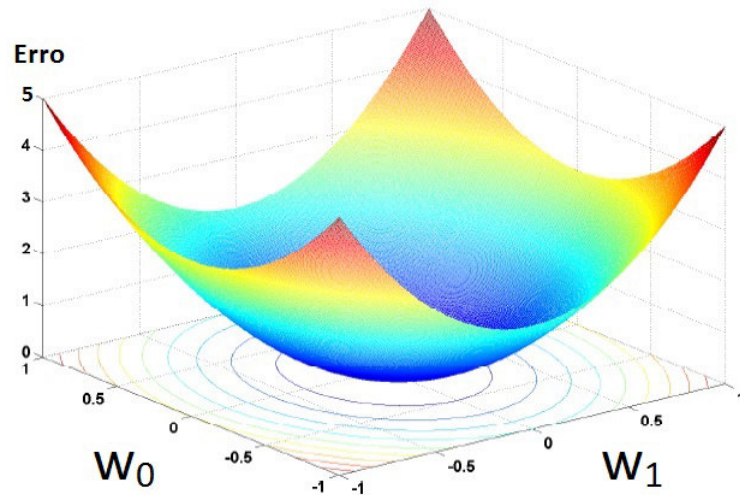


Figura 2.3: Porção da superfície quadrática tridimensional juntamente com alguns contornos. O erro quadrático médio está plotado na vertical, w_0 e w_1 variam de -1 a 1.

o algoritmo consiga ajustar os coeficientes do filtro a valores próximos da solução ótima.

- Desajuste: medida da diferença entre os parâmetros obtidos após um certo número de iterações e os parâmetros ótimos.
- Rastreamento: capacidade do algoritmo de acompanhar as variações das propriedades estatísticas de sinais não estacionários.
- Robustez: habilidade do algoritmo para operar satisfatoriamente com um sinal de entrada mal condicionado.
- Complexidade computacional: número de operações aritméticas requeridas para fazer uma iteração completa do algoritmo; tamanho da memória requerida e/ou flexibilidade da programação.
- Estrutura: refere-se à estrutura do fluxo da informação no algoritmo e a maneira de ser implementada em hardware.
- Propriedades numéricas: a implementação do algoritmo adaptativo em um processador digital pode fazer com que o algoritmo se desvie da operação ideal, determinada em precisão infinita, produzindo acumulação de erros de quantização. É possível que os erros se acumulem sem limite. Desta forma o algoritmo pode ser numericamente instável.

2.2 Aplicações

Os filtros têm sido aplicados com muito sucesso em diversas áreas. Todas estas aplicações têm em comum um vetor de entrada e um sinal desejado, os quais são utilizados juntamente com o sinal de erro para ajustar os coeficientes do filtro de forma a otimizar o seu desempenho. A diferença essencial entre as várias aplicações é a forma de extrair a resposta desejada, assim as aplicações dos filtros adaptativos podem ser divididas em [Diniz, (2013)]:

- **Identificação de sistemas:** o filtro adaptativo é usado para encontrar o modelo linear que melhor represente uma planta desconhecida. A planta e o filtro adaptativo são alimentados pelo mesmo sinal de entrada e a saída da planta é a resposta desejada para o filtro adaptativo. A identificação de sistemas adaptativa é frequentemente utilizada em sistemas de comunicações, sistemas de controle e em identificação de estruturas, (ver Fig. 2.4);

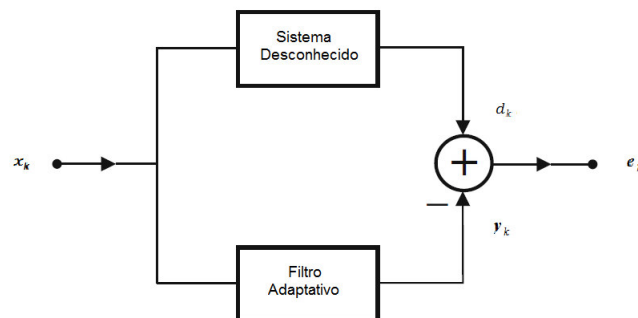


Figura 2.4: Modelo de identificação de sistema

- **Modelagem inversa:** o filtro adaptativo é usado para encontrar o modelo inverso de um sistema desconhecido. O sinal desejado para o filtro adaptativo é a entrada atrasada do sistema. Uma aplicação da modelagem inversa é a equalização de canal, em que o objetivo é reduzir a interferência intersimbólica através de um filtro adaptativo que inverta a resposta do canal, (ver Fig. 2.5);
- **Predição Linear:** este filtro adaptativo é utilizado para prover a predição do valor presente de um sinal aleatório. O valor presente serve como sinal desejado no filtro adaptativo e os valores passados alimentam o filtro. Dependendo da aplicação, pode-se ter duas saídas: y_k , denominada saída do filtro preditor ou e_k , denominada

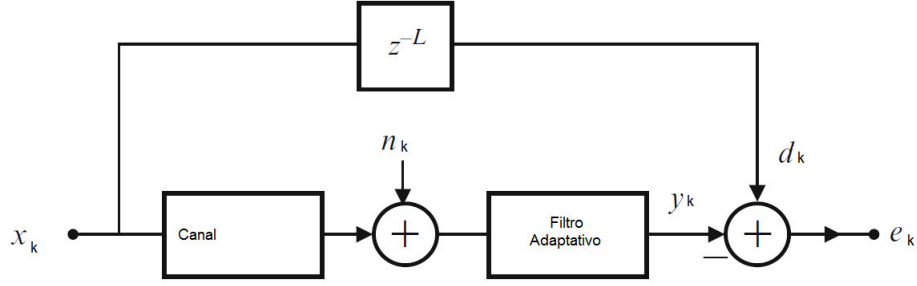


Figura 2.5: Modelo de Modelagem Inversa

erro de predição da saída do filtro. Uma de suas aplicações é na análise espectral, em que utiliza-se o modelamento preditivo para estimar a potência espectral de um sinal de interesse, (ver Fig. 2.6);

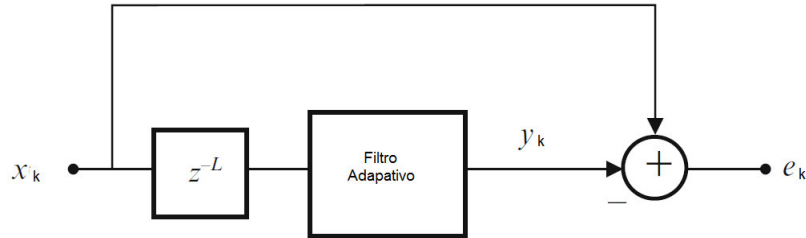
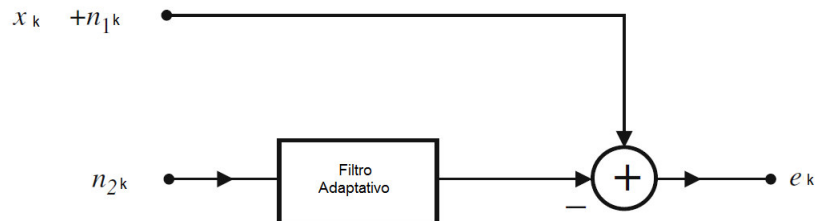


Figura 2.6: Modelo de Predição Linear

- **Cancelamento de interferência:** nesta aplicação, o filtro adaptativo é utilizado para cancelar uma interferência desconhecida contida em um sinal primário. Este sinal primário serve como sinal desejado para o filtro adaptativo. Um sinal correlacionado com a interferência alimenta a entrada do filtro. Utiliza-se em áreas tão diversar como o cancelamento de interferência de $50Hz$ ou $60Hz$ da rede elétrica em eletrocardiogramas, cancelamento de ruído em sinais de fala, dentre outros, (ver Fig. 2.7).

Figura 2.7: Modelo de Cancelamento de Ruído, onde n_{1k} e n_{2k} são ruídos conhecidos.

3 Efeitos da Precisão Finita

O mundo real consiste numa infinidade de sinais analógicos que são representados em precisão infinita, Fig. 3.1. Os sinais analógicos que não são ou não podem ser entendidos em linguagem de máquina, são geralmente observados na natureza e nos seres humanos na forma analógica. Para que nossos computadores, celulares e outros dispositivos de hardware possam trabalhar, é necessário que haja um processamento desses sinais analógicos na forma digital que são representados em precisão finita, ver Fig. 3.2. Alguns dispositivos de hardware podem interagir diretamente com programas, a fim de agilizar e dinamizar a execução. Dependendo do tipo e modelo do dispositivo de hardware pode-se encontrar placas de circuitos eletrônicos específicas para processamento de áudio, vídeo e imagem. Para tal são necessários dispositivos especiais de codificação e decodificação dos sinais, que realiza as conversões necessárias para que ocorra o processamento.

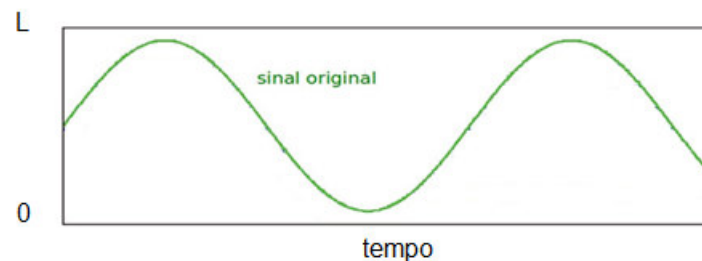


Figura 3.1: Representação em precisão infinita. O Sinal assume valores infinitos no intervalo considerado. Intervalo de amplitude $[0,L]$

O sinal digital é justamente uma representação numérica dos sinais analógicos. Para que haja essa representação, são utilizados os conversores A/D (*Conversores Analógicos Digitais*), que processam os sinais analógicos e os transformam em sinais digitais, dado numa sequência de 0s e 1s, podendo assim ser analisados por computadores e programas.

A representação dos números em dispositivos de hardware digital, números (de valor real ou valores complexos, inteiros ou frações) são representados usando dígitos binários (bits), que levam o valor de 0 ou 1. Para realizar as operações aritmética e lógicas

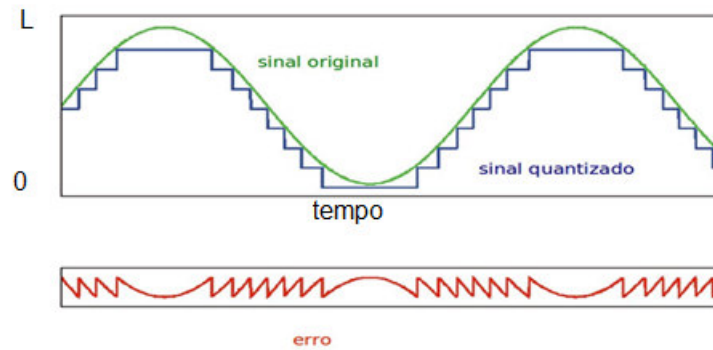


Figura 3.2: Representação em precisão finita. O Sinal assume valores finitos no intervalo. O erro é diferente de zero. Intervalo de amplitude $[0, L]$

necessárias para o processamento desses números, pode ser implementada duas abordagens diferentes, dependendo da facilidade de implementação da precisão, bem como da faixa dinâmica necessárias no processamento. A aritmética de ponto fixo que pode ser fácil de implementar, mas tem apenas uma faixa dinâmica e precisão fixa (isto é, quantidades de números muito grandes ou muito pequenas). A aritmética de ponto flutuante, por outro lado, tem uma vasta gama de faixa dinâmica e uma precisão variável (em relação à amplitude de um número), sendo portanto mais complicado de implementar e analisar [Tocci, (2013)].

Uma vez que um computador pode operar apenas em um binário variável (por exemplo, 0 ou 1), números positivos podem diretamente ser representados utilizando números binários. O problema é quanto ao modo de representar os números negativos. Os três formatos diferentes usados em cada uma dessas representações de sinal são: formato de 'sinal magnitude', formato 'complemento de um' e formato 'complemento de dois'. Considerando 1bits para a representação do sinal (positivo ou negativo) para os números binários. Com relação ao tipo de notação aritmética, tipicamente é utilizado pelos dispositivos de hardware a aritmética de ponto fixo [Tocci, (2013)]. Nela determina-se a quantidade de bits destinada para armazenamento da parte inteira e para parte fracionária, considerando $(b + 1)$ bits. [Padgett and Anderson, (2009)] ver Fig. 3.3.

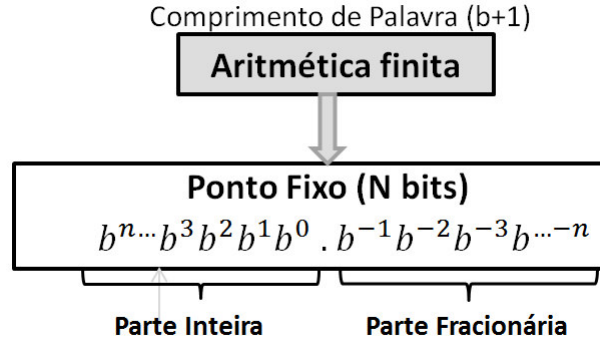


Figura 3.3: Esquema geral da aritmética de ponto fixo. Considerando comprimento de palavra $b + 1 \text{ bit}$ de sinal. Define número de b^n parte inteira e b^{-n} parte fracionária.

3.1 Erros de Quantização

Quantização é o processo de atribuição de valores discretos para um sinal cuja amplitude varia entre infinitos valores [Padgett and Anderson, (2009)].

Quando geramos um sinal digital, estamos realizando uma aproximação de uma grandeza por um valor quantizado tendo como consequência um erro de quantização [Padgett and Anderson, (2009)].

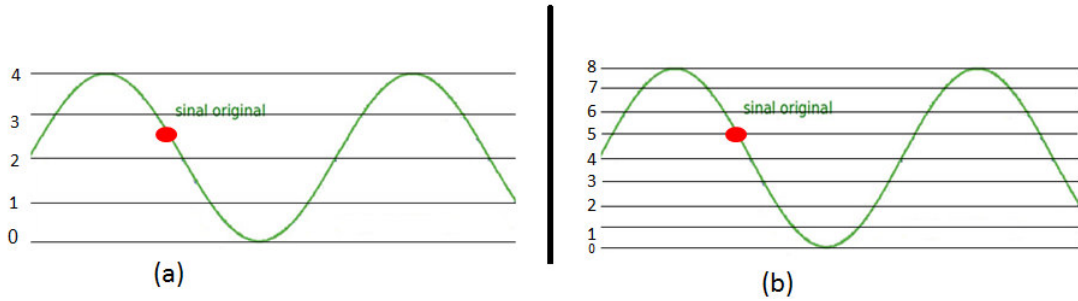


Figura 3.4: Representação Níveis de Quantização. a) Quantidades de níveis $N=4$. b) Quantidades de níveis $N=8$.

Considerando um ponto do sinal original (ponto vermelho) na representação em precisão finita cada valor do sinal é substituído pelo nível mais próximo. Quanto mais níveis menor será o erro de quantização, Fig. 3.4a e 3.4b.

Dessa forma podemos quantificar o erro de aproximação de seus valores originais, devido aos efeitos da quantização, dada por:

$$n_\theta = d_k - d_{Qk}, \quad (3.1)$$

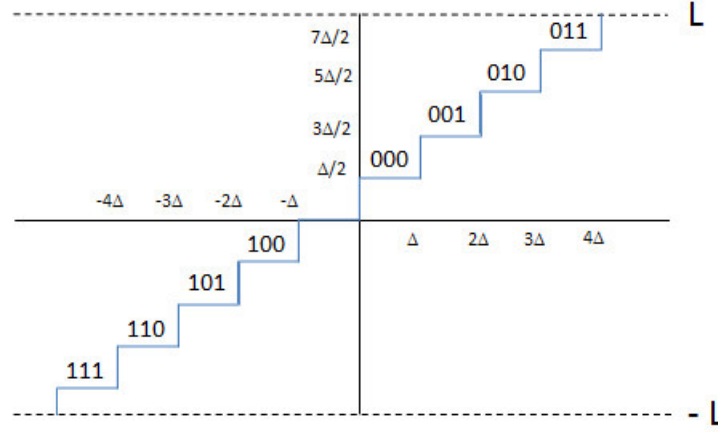


Figura 3.5: Representação esquemática de níveis de quantização para o caso de $b = 2$ bits e $\Delta = L/4$. Consideramos um valor L limiar. Usando b bits, o intervalo de $[0, L]$ pode ser dividido em 2^b níveis de largura $\Delta = L/2^b$ cada. Se, além disso, usamos 1 (um) bit de sinal, os resultantes $(b + 1)$ bits divide o intervalo $[-L, L]$ em 2^{2b} níveis. Aqui assumimos operação de arredondamento.

onde n_θ indica o erro ou ruído devido a quantização e θ representa a quantidade ou variável quantizada.

Utilizando aritmética de ponto fixo, com a operação de arredondamento assumido para a quantização, modelo mostrado na Fig. 3.5, o erro pode ser modelado em processo estocástico com média zero e variância dada por:

$$\sigma^2 = C \frac{2^{-2b}}{12}, \quad (3.2)$$

onde σ^2 é a variância, C é uma constante e b é o número de bits utilizado. Para filtragem adaptativa dada em precisão finita afim de facilitar os cálculos, supõe-se que os sinais internos são adequadamente dimensionados, de modo que nenhum excesso ocorre durante os cálculos, e que os valores dos sinais devam situar-se entre $[-1, 1]$.

Na Fig. 3.6, podemos observar o comportamento da variância dada por (3.2), para o número de bits variando entre $7bits$ e $32bits$.

3.2 Filtragem Adaptativa em Precisão Finita

Os efeitos numérico da precisão finita em algoritmos adaptativos, onde até

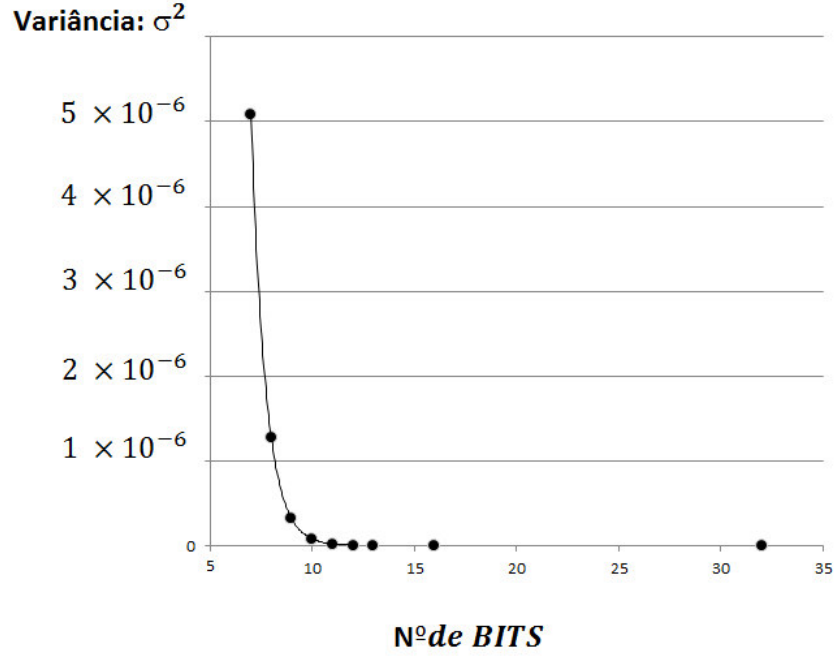


Figura 3.6: Variância devido à quantização com aritmética de ponto fixo. Para $C = 1$ número de *bits* variando entre *7bits* à *32bits*

agora nós consideramos modelos de filtros e implementações em que ambos os coeficientes do filtro e as operações do filtro, como adições e multiplicações foram expressos usando números em precisão infinita. Quando os sistemas de tempo discreto são implementados em hardware ou software, todos os parâmetros e operações aritméticas são implementados usando números de precisão finita e, portanto, seus efeitos são inevitáveis. Considere-se um filtro típico implementado como uma estrutura dada em precisão finita. Quando a representação em precisão finita é utilizado na sua aplicação, existem três possíveis considerações que afetam o desempenho global da sua implementação. Consideramos como as principais [Padgett and Anderson, (2009)]:

1. A quantização dos coeficientes filtro, w_k , para obter sua representação de comprimento finito, w_{Qk} ,
2. A quantização das variáveis e quantidades e da sequência de entrada, x_k para se obter x_{Qk} , e
3. Toda aritmética interna deve ser convertido em suas representações em precisão finita.

Assim, a saída, y_k , é também um valor quantizado y_{Qk} . Isso nos dá uma nova percepção do filtro. Espera-se dessa forma uma nova análise do filtro dado os novos sinais

e coeficientes encontrados.

Uma vez que a operação de quantização é uma operação não-linear, a análise global que leva em consideração todos os três efeitos descritos acima é de alta complexidade [Sayde, (2013)]. Portanto, faz-se necessário estudar cada um desses efeitos separadamente como se fosse o único agindo no momento. Isto torna mais fácil a análise e os resultados mais interpretáveis. Podemos definir os erros ou ruídos de quantização nas variáveis e quantidades relacionadas com a filtragem adaptativa de forma geral como segue:

$$n_d = d_k - d_{Qk}, \quad (3.3)$$

$$n_y = y_k - y_{Qk}, \quad (3.4)$$

$$n_e = e_k - e_{Qk}, \quad (3.5)$$

$$n_w = w_k - w_{Qk}, \quad (3.6)$$

onde n_d é o ruído de quantização do sinal desejado, n_y o ruído de quantização do sinal de saída, n_e o ruído de quantização do sinal de erro e n_w o ruído devido a quantização, dos coeficientes ou vetor peso. Supondo que o sinal de entrada $\mathbf{x}_k$ não sofre quantização. Os efeitos de quantização nos sinais de entrada e desejado pode ser facilmente tomado em consideração separadamente, a partir de outras fontes de erro de quantização. No caso do sinal desejado, o erro de quantização pode ser adicionado ao ruído de medição, enquanto para o sinal de entrada, o efeito de base na saída do filtro é um ruído adicional [Diniz, (2013)].

3.2.1 O LMS em Precisão Finita

O LMS é um algoritmo de busca baseado numa simplificação computacional do gradiente feita através de modificações pertinentes da função de custo. O algoritmo LMS, bem como outros com ele relacionados, é amplamente utilizado em várias aplicações de filtragem adaptativa devido à sua simplicidade computacional. O algoritmo LMS é, de longe, o algoritmo mais amplamente usado na filtragem adaptativa por várias razões. As principais características que atraíram a utilização do algoritmo LMS são a baixa complexidade computacional em ambiente estacionário, convergência imparcial na média a solução de Wiener e comportamento estável quando implementado com a aritmética de precisão finita [Diniz, (2013)] [Widrow, (1985)] [Haykin, (2014)].

O algoritmo LMS dado em precisão finita é descrito por.

$$\mathbf{w}_{k+1} = \mathbf{w}_k + 2\mu e_k \mathbf{x}_k, \quad (3.7)$$

onde $\mathbf{w}_k$ é o vetor peso, μ é um valor que controla a velocidade e estabilidade do algoritmo, e_k o erro e $\mathbf{x}_k$ o sinal de entrada.

As relações que descrevem os erros de cálculos no algoritmo LMS implementado em comprimento de palavra finita podem ser descritas como [Diniz, (2013)].

$$e_{Qk} = d_k - n_e - \mathbf{x}_k^t \mathbf{w}_k, \quad (3.8)$$

onde definimos n_e como ruído devido a quantização e $\mathbf{w}_k^t$ é o vetor dos pesos dado por

$$\mathbf{w}_{Q(k+1)} = \mathbf{w}_k + 2\mu e_{Qk} \mathbf{x}_k - n_w, \quad (3.9)$$

onde n_w é um vetor de ruído devido à quantização do vetor peso $\mathbf{w}_k$.

Os sinais de erro que consistem nos elementos de n_e e n_w são todos não correlacionados e independentes uns dos outros. A variância σ_e^2 de n_e e a covariância $\sigma_w^2 I$ de n_w são dadas por:

$$E[n_e^2] = \sigma_e^2, \quad (3.10)$$

$$E[n_w n_w^t] = \sigma_w^2 I. \quad (3.11)$$

Alguns autores como [Gupta e Hero, (2000)], [Caraiscos e Liu, (1984)] e [Meher e Park, (2014)], analisaram o algoritmo LMS em precisão finita, realizando análises sobre a matriz de covariâncias e vetor de coeficientes de erro, analisando o comportamento do algoritmo tanto na fase de adaptação como na fase de operação, observaram também o MSE e outras características importantes como o desajuste para análise desse algoritmo em precisão finita.

3.3 Implementação em Dispositivo de Hardware Digital

Muitos projetos em filtragem adaptativa são implementados em DSP ou FPGA, que apresentam-se muitas vezes em precisão finita. O uso desses dispositivos de hardware digital hoje em dia é muito grande. Vários itens como: celulares, equipamentos de laboratórios, automóveis e computadores fazem parte destes dispositivos de hardware. Estas implementações tem o potencial de gerar efeitos indesejados na resposta do filtro, além de instabilidade numérica. Como resultado, surgem dúvidas, na fase de implementação quanto à eficácia do filtro digital e a quantidades de bits necessária para a sua representação de modo que os parâmetros de projetos ainda sejam satisfeitos.

Estes por serem dispositivos programáveis necessitam de uma plataforma para desenvolvimento de projetos. Estes dispositivos trabalham com linguagem de programação, em sua grande maioria com as principais linguagens, como Assembly, C, VHDL e outras. Cada fabricante fornece junto ao produto uma plataforma, que foi feita para trabalhar com as funções pré-definidas de cada dispositivos de hardware. Estas plataformas fornecem um ambiente de trabalho necessário para a realização de um projeto. Estes software são na verdade compiladores que trabalham ligados aos dispositivos, gerenciando toda a execução, simulação e depuração do código [Tocci, (2013)].

Para a realização de um projeto utilizando-se um dispositivo de hardware digital podemos destacar três etapas gerais para execução do projeto: pesquisa, simulação e emulação [Padgett and Anderson, (2009)].

- 1º Etapa: A pesquisa envolve basicamente o entendimento do problema, onde serão buscadas informações se o dispositivo a ser utilizado envolve necessidade de cálculos em ponto fixo ou em ponto flutuante.
- 2º Etapa: A etapa de simulação ocorrerá na modelagem inicial do projeto, onde é realizada, através dos ambientes de desenvolvimento de cada dispositivo.
- 3º Etapa: Por fim, temos a etapa de emulação, que envolve a realização de testes do projeto já dentro de seu ambiente de uso. Nesta etapa o dispositivo de hardware é submetido as análises detalhadas, visando-se obter um equipamento o mais preciso possível, que não apresente falhas.

3.4 Simulação de Monte Carlo

A simulação de Monte Carlo ou Método de Monte Carlo (MMC) baseia-se em métodos estatísticos que reúnem uma grande quantidade de amostras aleatórias de uma simulação ou experimento com objetivo de chegar a um resultado aproximado de uma solução. Pode ser visto também, como método numérico universal para resolver problemas por meio de amostragem aleatória [Tao et al., (1995)].

O Método de Monte Carlo é aplicado em diversos campos de conhecimento. Alguns exemplos de aplicação deste método são [Tao et al., (1995)]:

- Finanças: séries macroeconômicas, opções futuras, etc.;
- Computação Gráfica: redução de artefatos, espalhamento, etc.;
- Geologia: caracterização de reservatórios;
- Análise de Projetos: opções reais;
- Jogos: geração de redes (grafos).

A principal característica desse Método de Monte Carlo é a aleatoriedade com que ocorre o fenômeno em estudo. Uma exigência para utilização desse método é que o fenômeno em estudo seja modelado em termos de funções estatísticas. A Simulação de Monte Carlo pode proceder fazendo amostragens aleatórias de funções estatísticas conhecidas. Este processo é repetido inúmeras vezes e o resultado desejado é obtido por meio de técnicas estatísticas (média, desvio padrão, e etc.) sobre um determinado número finito de realizações (amostras) que podem chegar a milhões.

4 Análise em Precisão Finita do Algoritmo Adaptativo Baseado no Erro Não Linear

A filtragem adaptativa tem conseguido uma grande quantidade de sucesso em numerosas aplicações. Muito deste sucesso é devido ao desenvolvimento de conhecidos e robustos algoritmos adaptativos, tais como o algoritmo LMS, que tenta encontrar os parâmetros do filtro adaptativo para minimizar o critério do erro quadrático médio. Característica de desempenho deste algoritmo pode ser aproximadamente prevista dadas informações de covariância da entrada estocástica. Além disso, dado que a entrada é estacionária estocástica, a média convergente de pesos do algoritmo é a solução de Wiener na teoria dos mínimos quadrados de estimativa lineares. Algoritmos adaptativos com base no critério de mínimo erro quadrático médio representam uma referência com características bem conhecidas que podem ser comparados a outros algoritmos [Widrow, (1985)].

A filtragem adaptativa com base em critérios de erro não quadrático médio também têm sido sugeridos por vários autores, quer como uma forma de melhorar o desempenho de adaptação destes algoritmos para ambientes estatísticos particulares [Walach e Widrow, (1984)], [Gibson e Gray, (1988)], ou como forma de simplificar o cálculo do algoritmo em hardware.

Douglas e Meng comentando a respeito de filtragem adaptativa baseadas em critérios de erro não quadrático médio, a fim de unificar a discussão desses algoritmos modificados realizaram um série de análise desse algoritmo em 1994 no artigo "Stochastic gradient adaptation under general error criteria. IEEE Transactions on Signal Proc., Vol. 42, n° 6."

Considere o problema de identificação de sistema, ver Fig. 4.1. Em que o objetivo é encontrar uma planta desconhecida com função de transferência polinomial dada por $\mathbf{w}_k^*(z)$ para ser modelado por um filtro adaptativo de tempo discreto $\mathbf{w}_k(z)$. A função de transferência da planta e do filtro adaptativo pode ser representada na forma

de resposta ao impulso finita (FIR) dado por:

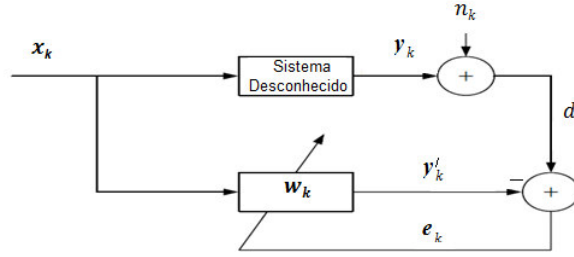


Figura 4.1: Filtragem adaptativa em identificação de sistema.

$$\mathbf{w}(z) = w_1 + w_2 z^{-1} + w_3 z^{-2} + \dots + w_N z^{-N+1}, \quad (4.1)$$

onde definimos o vetor $\mathbf{w} = [w_1 w_2 \dots w_N]^T$ tal que $\mathbf{w} = \mathbf{w}^*$ para o sistema desconhecido e $\mathbf{w} = \mathbf{w}_k$ para o filtro adaptável a cada iteração k . Desejamos obter estimativas precisas dos coeficientes do modelo da planta $\mathbf{w}^*$ dadas as observações ruidosas da saída da planta d_k , o que pode ser considerado como a resposta desejada do modelo adaptativo, quando acionado pela entrada estocástica $\mathbf{x}_k$ e perturbado com amostras de ruído n_k . Para derivar a família de algoritmos de adaptação de gradiente estocástico com base em critérios de erro não quadrático médio, que em primeiro lugar formar o erro de valor escalar e_k como a diferença entre a resposta desejada e a saída do modelo adaptativo y_k . Nós usamos esse erro para ajustar o vetor de coeficientes de filtro $\mathbf{w}_k$ em cada iteração no algoritmo dada da forma [Douglas and Meng, (1994)]:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \hat{\nabla}_k, \quad (4.2)$$

onde μ é um parâmetro de tamanho do passo e $\hat{\nabla}_k$ é o gradiente do vetor peso $\mathbf{w}_k$ de uma função de erro (e_k), o chamado critério de erro. Este critério de erro é ainda simétrico e geralmente satisfaz as condições seguintes [Douglas and Meng, (1994)]:

$$0 \leq \theta(e_k) = \theta(-e_k) \quad (4.3)$$

$$0 \leq e_1 \leq e_2 \rightarrow \theta(-e_k) \leq \theta(e_2). \quad (4.4)$$

Avaliando o gradiente $\hat{\nabla}_k$, descobrimos que [Douglas and Meng, (1994)]:

$$\hat{\nabla}_k = \frac{\partial \theta(e_k)}{\partial \mathbf{w}_k} \quad (4.5)$$

$$= \frac{\theta'(e_k) \partial e_k}{\partial \mathbf{w}_k} \quad (4.6)$$

$$= -\theta'(e_k) \mathbf{x}_k, \quad (4.7)$$

onde o primeiro funcional denota diferenciação. Definindo a função $f(e_k) = \theta(e_k)$, podemos expressar o algoritmo de gradiente estocástico pelo seguinte conjunto de Equações [Douglas and Meng, (1994)]:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu f(e_k) \mathbf{x}_k, \quad (4.8)$$

$$e_k = d_k - y_k, \quad (4.9)$$

$$y_k = \mathbf{w}_k^t \mathbf{x}_k. \quad (4.10)$$

Comparando (4.7) e (4.9) com equações semelhantes definidas para a adaptação do algoritmo LMS [Widrow, (1985)], percebemos que um critério de erro geral leva a uma modificação do erro no algoritmo LMS pela função ímpar-simétrica não-linear $f(\cdot)$ em estimativa instantâneas do gradiente $\hat{\nabla}_k$.

Algoritmos adaptativos formulados por (4.7) e (4.9) foram abordados dentro da comunidade de processamento de sinal com menos frequência, principalmente por causa de várias dificuldades matemáticas associadas à sua análise. Estas dificuldades não descartam a sua utilidade potencial na filtragem adaptativa. No entanto, onde foi indicado que o uso de um critério de erro não quadrático médio pode melhorar o desempenho de um algoritmo adaptativo quando a distribuição do ruído de interferência não é gaussiana [Gibson and Gray, (1988)]. Esta negligência de critérios de erro não quadrático médio ilumina um viés predominante na comunidade de processamento de sinal em direção estatísticas gaussianas, um viés que é motivada mais pela tratabilidade matemática ao invés de experiências práticas. Estatísticas gaussianas são muitas vezes justificados através do Teorema do Limite Central (TLC); No entanto, essas referências à TLC são questionáveis, como o número de sinais de interferência independentes, deve ser muito grande para alcançar gaussianidade aproximada. Além disso, há uma série de comunicações práticas e ambiente

de processamento de sinal, tais como canais eletromagnéticos de transmissão, ambientes acústicos submarinos, e canais telefônicos, em que a interferência não-Gaussiana existe, é conhecida e tem sido com precisão modelada. Algoritmos usando critérios de erro não médios quadráticos são úteis para potenciais melhorias de desempenho nessas situações. Para interferências gaussianas, esses algoritmos também levam a uma estimativa imparcial dos coeficientes do filtro sob as restrições dadas em (4.3) [Johnson and Rao, (1991)].

A escolha de um critério de erro dada uma informação a priori sobre a distribuição de ruído foi abordada na análise de filtragem adaptativa como uma procura de parâmetro em que uma estrutura adequada do critério é assumida, e o algoritmo é otimizado para encontrar os parâmetros que permitam atingir o máximo desempenho melhorado em relação ao algoritmo LMS. Tal método de concepção geralmente leva a um algoritmo sub-ótimo como o critério é limitado a uma classe restrita de funções. No campo de identificação de sistema, os critérios de erro não quadráticos médios foram usados para resolver o problema de outliers de dados, que tendem a causar grandes erros nas estimativas dos parâmetros [Walach and Widrow, (1984)].

4.1 Análise de Adaptação

Considere o diagrama em blocos mostrado na Fig. 4.2, referente ao problema de identificação de sistemas com filtragem adaptativa em um hardware.

Nesta seção, invocamos uma série de pressupostos sobre a natureza estatística dos dados de entrada e ruído característico do modelo, comumente utilizados na literatura [Walach and Widrow, (1984)]. Nesta análise, nós linearizamos uma função não linear arbitrária $f(e_k)$ sobre um ponto de funcionamento adequado (amostra de ruído n_k) e, em seguida, prosseguimos com a análise de covariâncias em estado estacionário. Aqui assumimos os seguintes pressupostos [Fonseca et al., (2017)]:

- assume-se que $\mathbf{x}_k$ é um sinal aleatório discreto com $-1 < \mathbf{x}_k \leq 1$, não sofre quantização e cada $\mathbf{x}_k$ vetor de dados de entrada é estatisticamente independente de todos os vetores de dados anteriores $\mathbf{x}_k$ para $j < k$.
- a sequência de ruído amostrado (por medição ou erros de quantização gerado em

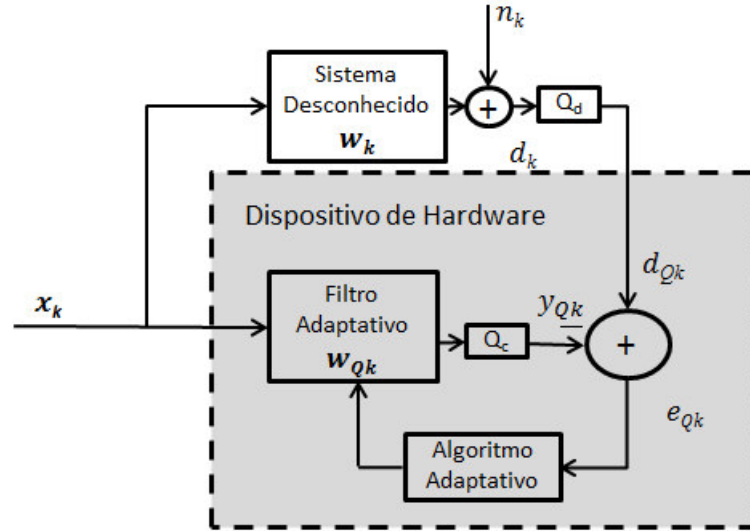


Figura 4.2: Identificação de sistema implementado em precisão finita. Onde $\mathbf{x}_k$ designa o sinal de entrada. Os blocos Q_d e Q_c são quantizadores de b bits. Assim, o quantizador, Q_d , usa b_d bits para os dados de entrada e Q_c quantiza os coeficientes em b_c bits. O sinal de saída y_{Qk} indica a forma quantizada. O sinal desejado definido por d_k do sistema desconhecido e d_{Qk} sua forma quantizada. O ruído de medição n_k e o sinal de erro quantizado e_{Qk} . Os coeficientes $\mathbf{w}_k$ do sistema desconhecido e $\mathbf{w}_{Qk}$ do filtro adaptativo.

qualquer etapa no algoritmo) é considerado uma variável aleatória com média zero, que é estatisticamente independente de quaisquer outros erros e/ou quantidades relacionadas.

- o parâmetro de tamanho do passo μ deve ser escolhido suficientemente pequeno de tal modo que o excesso de erro médio quadrático na iteração k seja muito menor do que o mínimo erro médio quadrático perto de convergência.
- as variações dos erros dependem do tipo da quantização e aritmética que vai ser utilizada na implementação do algoritmo. Em nosso caso utilizamos aritmética de ponto fixo. Utilizando aritmética de ponto fixo o erro pode ser modelado em processo estocástico com média zero, com variância dada por (3.2). [Diniz, (2013)].

Expressando a saída dos sistema desconhecido como [Fonseca et al., (2016)]:

$$d_k = \mathbf{w}^{*t} \mathbf{x} + n_k, \quad (4.11)$$

onde $\mathbf{w}^*$ é o coeficiente ótimo. Supõe-se que o sinal de entrada e o sinal desejado não

sofre quantização, de modo que apenas a quantização interna de computação são tidas em conta. Assim, podemos expressar o erro como [Fonseca et tal., (2016)]:

$$e_k = (\mathbf{w}^{*t} \mathbf{x}_k - n_k) - \mathbf{w}_k^t \mathbf{x}_k. \quad (4.12)$$

Em sua forma quantizada como [Fonseca et tal., (2016)]:

$$\begin{aligned} e_{Qk} &= ((\mathbf{w}^{*t} \mathbf{x}_k - n_k) - \mathbf{w}_k^t \mathbf{x}_k)_Q \\ &= n_k - n_e - \mathbf{v}_k^t \mathbf{x}_k, \end{aligned} \quad (4.13)$$

onde definimos n_e como o ruído devido a quantização e $\mathbf{v}_k^t$ é o vetor desvio do peso dado por [Fonseca et tal., (2016)]:

$$\mathbf{v}_k = \mathbf{w}_k - \mathbf{w}^*. \quad (4.14)$$

Subtraindo $\mathbf{w}^*$ de ambos os lados de (4.13), o algoritmo de adaptação de erro torna-se [Fonseca et tal., (2016)]:

$$(\mathbf{v}_{k+1})_Q = (\mathbf{v}_k + \mu f(n_k - \mathbf{v}_k^t \mathbf{x}_k) \mathbf{x}_k)_Q. \quad (4.15)$$

Que agora é uma equação não linear iterativo no vetor de desvio $\mathbf{v}_k$.

Expressando a não linearidade $f(e_k)$ em uma expansão em série de Taylor sobre o valor do ruído, obtemos [Fonseca et tal., (2016)]:

$$\begin{aligned} f(e_k) &= \sum_{i=0}^{\infty} \frac{f^i}{i!} (-n_{ek} - \mathbf{v}_k^t \mathbf{x}_k)^i \\ &= f(n_k) - f'(n_k)(n_e + \mathbf{v}_k^t \mathbf{x}_k) + \frac{f''(n_k)}{2}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^2 \\ &\quad - \frac{f'''(\delta)}{6}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^3, \end{aligned} \quad (4.16)$$

onde $f^i(\cdot)$ denota a derivada (i) da função $f(\cdot)$. Temos utilizado a forma derivada para representar o termo restante de Taylor tal que δ encontra-se no intervalo $[0, n_{ek} + \mathbf{v}_k^t \mathbf{x}_k]$. Tecnicamente esta representação de série de Taylor limita a não linearidade $f(\cdot)$ para uma classe restrita de funções que são diferenciáveis até a terceira ordem ao longo de toda extensão do erro e_k .

Agora vamos examinar o comportamento de (4.16) no caso em que o vetor de desvio do peso é pequeno, de tal modo que o termo $\frac{f'''(\delta)}{6}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^3$ pode ser desprezado em relação à menor ordem do termo. Sendo representado por [Fonseca et tal., (2016)]:

$$\mathbf{v}_{Q(k+1)} = \mathbf{v}_k + \mu[f(n_k)\mathbf{x}_k - f'(n_k)(n_e + \mathbf{v}_k^t \mathbf{x}_k)\mathbf{x}_k + \frac{f''(n_k)}{2}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^2 \mathbf{x}_k] - n_v, \quad (4.17)$$

onde n_v é um vetor de ruído devido à quantização do vetor peso $\mathbf{v}_k$.

Tomando a expectância de ambos os lados, obtemos [Fonseca et tal., (2016)]:

$$E[\mathbf{v}_{Q(k+1)}] = E[\mathbf{v}_k] + \mu E[f(n_k)\mathbf{x}_k - f'(n_k)(n_e + \mathbf{v}_k^t \mathbf{x}_k)\mathbf{x}_k + \frac{f''(n_k)}{2}(n_e + \mathbf{v}_k^t \mathbf{x}_k)^2 \mathbf{x}_k] - E[n_v]. \quad (4.18)$$

E utilizando os pressupostos apresentados ficamos [Fonseca et tal., (2016)],

$$E[\mathbf{v}_{Q(k+1)}] = E[\mathbf{v}_k] + \mu E[-f'(n_k)]\{E[n_e] + E[\mathbf{v}_k^t]E[\mathbf{x}_k \mathbf{x}_k]\}. \quad (4.19)$$

Aqui consideramos $\mathbf{R} = E[\mathbf{x}_k \mathbf{x}_k^t]$, como matriz de autocorrelação do sinal de entrada.

$$E[\mathbf{v}_{Q(k+1)}] = E[\mathbf{v}_k] - \mu E[f'(n_k)]\{E[n_e] + E[\mathbf{v}_k^t]\mathbf{R}\}, \quad (4.20)$$

$$= (I - \mu E[f'(n_k)]\mathbf{R})E[\mathbf{v}_k]. \quad (4.21)$$

Em que utilizamos o fato de que os momentos ímpares do ruído de medição n_k são iguais a zero. Definindo um tamanho efetivo do passo de adaptação μ_{eff} como [Fonseca et tal., (2016)]:

$$\mu_{eff} = \mu E[f'(n_k)]. \quad (4.22)$$

A (4.21), se torna:

$$E[\mathbf{v}_{Q(k+1)}] = (I - \mu_{eff} \mathbf{R})E[\mathbf{v}_k]. \quad (4.23)$$

Em que $\mathbf{R}$ é a matriz de autocorrelação dos sinais de entrada. Supondo que o sinal de entrada $(\mathbf{x}_k)_Q$ sofre quantização devido a limitação do comprimento de palavra em sua representação em precisão finita, essa quantização no sinal entrada influencia diretamente a matriz de autocorrelação do sinal de entrada $\mathbf{R} = E[\mathbf{x}_k \mathbf{x}_k^t]$ variando o espalhamento de autovalores podendo impactar no comportamento transitório do algoritmo [Fonseca et al., (2016)].

4.2 Comportamento de Convergência

A (4.24) corresponde exatamente a forma derivada para o algoritmo de função de custo de erro geral em precisão infinita, podendo ser usada para determinar um limite aproximado para o tamanho de μ que garanta a convergência em média, respeitando os pressupostos apresentados. Pode-se determinar a constante de tempo associado com o autovalor, em geral, o critério de erro de adaptação será dado na forma de [Fonseca et al., (2016)]:

$$\mu_{eff} = \frac{1}{\mu E[f'(n_k)]}. \quad (4.24)$$

A fim de determinar equações para o desajustes, utilizamos a (4.17) e multiplicando ambos os lados por sua transposta, em seguida tomando a esperança, invocando os pressupostos e assumindo que o vetor de erro de peso é pequeno, podemos ignorar os termos de ordem O^3 [Fonseca et al., (2017)].

$$\begin{aligned} E[\mathbf{v}_{Q(k+1)} \mathbf{v}_{Q(k+1)}^t] &= E[\mathbf{v}_{Qk} \mathbf{v}_{Qk}^t] - 2\mu[f'(n_k)]\mathbf{R}E[\mathbf{v}_{Qk} \mathbf{v}_{Qk}^t] + \mu^2 E[f^2(n_k)]\mathbf{R} \\ &+ \mu^2 E[f^2(n_k)]\mathbf{R}\sigma_e^2 + \mu^2 E[f'^2(n_k)]E[\mathbf{x}_k \mathbf{x}_k^t \mathbf{v}_k \mathbf{v}_k^t \mathbf{x}_k \mathbf{x}_k^t] \\ &+ \mu^2 E[f(n_k) f''(n_k)]E[\mathbf{x}_k \mathbf{x}_k^t \mathbf{v}_k \mathbf{v}_k^t \mathbf{x}_k \mathbf{x}_k^t] + \sigma_v^2 I. \end{aligned} \quad (4.25)$$

Para $\mathbf{R} = E[\mathbf{x}_k \mathbf{x}_k^t]$, $E[\mathbf{n}_v \mathbf{n}_v^t] = \sigma_v^2 I$ e $E[n_e^2] = \sigma_e^2$.

Podemos desprezar o quinto e sexto termo em (4.26), usando os pressupostos apresentados, porque se μ e $\mathbf{v}_{Qk}$ são pequenos, esses termos são muito menores do que os outros termos. Essa aproximação é justificada para continuar com a derivação sem fazer

suposições mais restritivas sobre a distribuição de dados de entrada. No caso de dados de entrada Gaussianos podemos assumir que [Fonseca et al., (2017)]:

$$\Lambda E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] = \frac{E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] + \mu^2 E[f^2(n_k)] \Lambda + \mu^2 E[f^2(n_k)] \Lambda \sigma_e^2 + \sigma_v^2 I \Lambda}{2\mu E[f'(n_k)]}, \quad (4.26)$$

onde, Λ é a matriz diagonal de autovalores e $\tilde{\mathbf{v}}_{Qk}$ uma transformação linear do vetor peso.

Podemos utilizar a transformação de similaridade para expressar (4.26) como um conjunto de equações iterativas dissociadas. Definindo a matriz unitária $\mathbf{L}$ de autovetores e a matriz diagonal Λ de autovalores podemos expressar a matriz de valor real e covariância simétrica como $\mathbf{R} = \mathbf{L} \Lambda \mathbf{L}^t$. Assim, definindo o vetor de erro de peso no domínio de coordenadas transformadas como $\tilde{\mathbf{v}}_{Qk} = \mathbf{L}^t \mathbf{v}_{Qk}$, e para $f(n_k) \gg f'(n_k)$, temos [Fonseca et al., (2017)]:

$$\begin{aligned} E[\tilde{\mathbf{v}}_{Q(k+1)} \tilde{\mathbf{v}}_{Q(k+1)}^t] &= E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] - 2\mu E[f'(n_k)] \Lambda E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] + \mu^2 E[f^2(n_k)] \Lambda \\ &+ \mu^2 E[f^2(n_k)] \Lambda \sigma_e^2 + \sigma_v^2 I \Lambda. \end{aligned} \quad (4.27)$$

A solução resultante de estado estacionário pode ser encontrado através da substituição dada por:

$$E[\tilde{\mathbf{v}}_{Q(k+1)} \tilde{\mathbf{v}}_{Q(k+1)}^t] = E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t], \quad (4.28)$$

resultando em:

$$\Lambda E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] = \frac{E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] + \mu^2 E[f^2(n_k)] \Lambda + \mu^2 E[f^2(n_k)] \Lambda \sigma_e^2 + \sigma_v^2 I \Lambda}{2\mu E[f'(n_k)]}. \quad (4.29)$$

Multiplicando ambos os membros por Λ^{-1}

$$\begin{aligned} \Lambda^{-1} \Lambda E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] &= \Lambda^{-1} \left[\frac{E[\tilde{\mathbf{v}}_{Qk} \tilde{\mathbf{v}}_{Qk}^t] + \mu^2 E[f^2(n_k)] \Lambda + \mu^2 E[f^2(n_k)] \Lambda \sigma_e^2 + \sigma_v^2 I \Lambda}{2\mu E[f'(n_k)]} \right] \\ &= \frac{1}{2E[f'(n_k)]} (\mu E[f^2(n_k)] + \mu E[f^2(n_k)] \sigma_e^2 + \frac{\sigma_v^2}{\mu}) I. \end{aligned} \quad (4.30)$$

Considerando que $f(n_k) \gg f'(n_k)$, então:

$$E[\tilde{\mathbf{v}}_{Q_k} \tilde{\mathbf{v}}_{Q_k}^t] = \frac{1}{2E[f'(n_k)]} (\mu E[f^2(n_k)] + \frac{\sigma_v^2}{\mu}) I. \quad (4.31)$$

Que possui a solução dada por:

$$E[\tilde{\mathbf{v}}_{Q_k} \tilde{\mathbf{v}}_{Q_k}^t] = E[\mathbf{v}_{Q_k} \mathbf{v}_{Q_k}^t]. \quad (4.32)$$

Sabe-se que o excesso de erro médio quadrático na saída do filtro é causado por flutuações aleatórias (i) nos coeficientes de peso durante a adaptação, ver (4.33). No qual assumimos a independência de $\mathbf{x}_k$ e $\mathbf{v}_k$, bem como a natureza não correlacionada do vetor de erro de peso:

$$\begin{aligned} E[(\mathbf{v}_{Q_k}^t \mathbf{x}_{Q_k})^2] &= E[(\sum_{i=1}^N v_{i,k} x_{i,k})^2] \\ &= \sum_{i=1}^N E[v_{i,k}^2] E[x_{i,k}^2], \end{aligned} \quad (4.33)$$

$$E[(\mathbf{v}_{Q_k}^t \mathbf{x}_{Q_k})^2] = \frac{1}{2E[f'(n_k)]} (\mu E[f^2(n_k)] + \frac{\sigma_v^2}{\mu}) E[x_{i,k}^2]. \quad (4.34)$$

O desajuste é definido como excesso de erro quadrático médio sobre o erro mínimo quadrático médio e dado por [Fonseca et al., (2017)]:

$$M = \frac{\xi_{excess}}{\xi_{min}}, \quad (4.35)$$

$$M = \frac{E[(\mathbf{v}_{Q_k}^t \mathbf{x}_{Q_k})^2]}{E[n_k^2]}, \quad (4.36)$$

onde $tr[\cdot]$ denota a operação de traço $E[x_{i,k}^2] = tr[\mathbf{R}]$ e $E[n_k^2] = \sigma_k^2$ variância do ruído de medição. O desajuste pode ser dado por:

$$M = \frac{tr[\mathbf{R}]}{2E[f'(n_k)]\sigma_k^2} (\mu E[f^2(n_k)] + \frac{\sigma_v^2}{\mu}). \quad (4.37)$$

Para $\frac{\sigma_v^2}{\mu} = 0$, não tem o ruído devido a quantização.

5 O Algoritmo Sigmoidal em Precisão Finita

O filtro Adaptativo Sigmoidal (SA) tem como função de custo o logaritmo natural do cosseno hiperbólico aplicado sobre o erro, definida por: $F_k(e) = Ln \cosh(\alpha e)$, essa função é uma não linearidade par, contínua, simétrica e não tem mínimo local, apenas o mínimo global [Santana, (2006)].

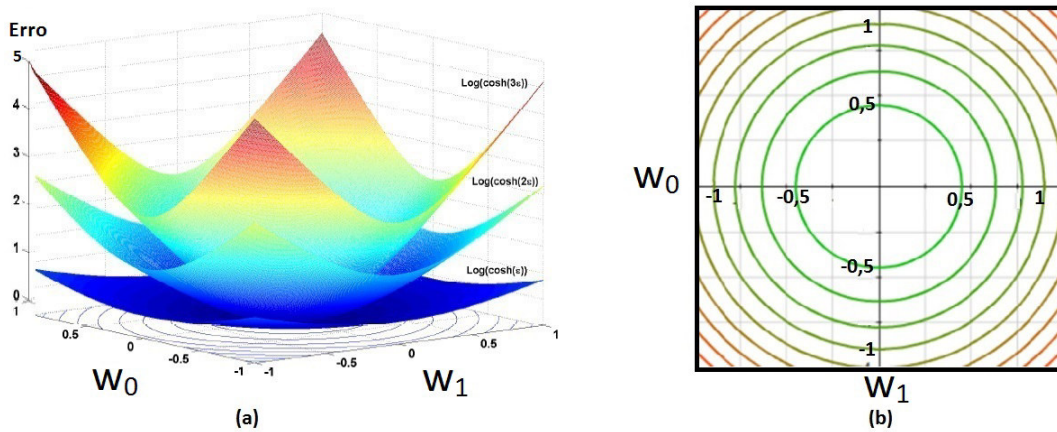


Figura 5.1: a) Porção da superfície tridimensional gerada pela função $Ln \cosh(\alpha e)$. b) Curva de Nível da função $Ln \cosh(\alpha e)$.

A partir da função $F_k(e) = Ln \cosh(\alpha e)$, pode-se gerar uma família de funções, multiplicando o argumento e por um inteiro positivo α . Ou seja, quando aumenta α , aumenta-se a inclinação da superfície de desempenho, onde as curvas destas funções tem inclinações maiores do que a curva da função quadrática do algoritmo LMS [Santana, (2006)]. Para desenvolver o algoritmo SA, estimamos o gradiente de $F_k(e)$. Isto é,

$$\hat{\nabla} F_k(e) = -\alpha \tanh(\alpha e) \mathbf{x}_k, \quad (5.1)$$

onde $\tanh$, representa a tangente hiperbólica aplicada sobre o erro. Logo o algoritmo adaptativo será dado por:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \alpha \tanh(\alpha e) \mathbf{x}_k, \quad (5.2)$$

onde o μ é uma constante que regula a velocidade e a estabilidade da adaptação [Santana, (2006)].

Considerando que a Fig. 5.1, e a estrutura do algoritmo SA, as equações que definem a atualização dos coeficientes do filtro adaptativo em precisão finita podem se descrever por [Fonseca et al., (2016)]:

$$d_{Qk} = (d_k + n_k)_Q, \quad (5.3)$$

$$y_{Qk} = (\mathbf{w}_{Qk} \mathbf{x}_k)_Q, \quad (5.4)$$

$$e_{Qk} = (d_{Qk} - y_{Qk})_Q. \quad (5.5)$$

Tal que (5.2) pode ser representado por:

$$\mathbf{w}_{Q(k+1)} = (\mathbf{w}_{Qk} + \mu \alpha \tanh(\alpha e_{Qk}) \mathbf{x}(k))_Q, \quad (5.6)$$

onde Q é a forma quantizada. Todos os escalares e elementos do vetor do algoritmo SA irá desviar os seus valores originais, devido aos efeitos da quantização. Podemos definir os erros ou ruídos de quantização nas variáveis e quantidades relacionadas com algoritmo SA como [Fonseca et al., (2016)]:

$$n_\theta = \theta_k - \theta_{Qk}, \quad (5.7)$$

onde θ representa a variável ou quantidade e n_θ é o ruído de quantização.

5.1 Método

O processo de implementação do filtro adaptativo SA em aritmética de ponto fixo obedeceu ao seguinte fluxograma de etapas, ver Fig. 5.2:

Etapa 1: antes de implementar um filtro adaptativo em precisão finita de ponto fixo, é necessário criar um filtro adaptativo de ponto flutuante com resultados semelhantes ao filtro adaptativo na forma analógica (precisão infinita) como filtro de referência. Em seguida procedem-se algumas etapas:

Etapa 2: simular em ponto flutuante o filtro de referência. Nesta etapa determina-se o comprimento apropriado do filtro, tamanho do passo, a variação dos coeficientes e outras características do filtro de referência.

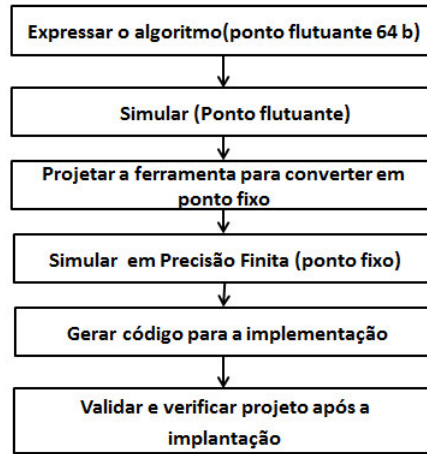


Figura 5.2: Diagrama do fluxo do projeto SA em precisão finita

Etapa 3: projeta-se a ferramenta para converter o filtro adaptativo em ponto flutuante para ponto fixo.

Etapa 4: simula-se o filtro adaptativo de ponto fixo. Nesta etapa determina-se o comprimento apropriado de palavra (número de bits) para parte inteira e parte fracionária para o conjunto de variáveis ou parâmetros do filtro adaptativo de ponto fixo.

Etapa 5: gerar-se o código para o filtro adaptativo de ponto fixo.

Etapa 6: valida-se o projeto analisando as suas curvas de aprendizagem.

Para utilizar a função $\tanh(e)$ em precisão finita em aritmética de ponto fixo, foi necessário realizarmos uma aproximação desta função em série de Taylor. Dada uma função indefinidamente diferenciável num certo ponto a interior ao seu domínio (isto é, existe e é finita a derivada de qualquer ordem de f em a , $f^n(a)$) podemos sempre escrever a sua série de Taylor relativa em torno de um ponto e [Douglas e Menq, (1994)].

$$\tanh(a) = \sum_{n=0}^n \frac{f^n(a)(a - e)^n}{n!}. \quad (5.8)$$

Aplicamos o algoritmo adaptativo em um problema largamente utilizado na literatura sobre filtros adaptativos que consistem na identificação de um sistema desconhecido, ver Fig. 4.2, onde temos uma planta representada pela função de transferência polinomial $P(z) = 0.2037z^{-1} + 0.5926z^{-2} + 0.2037z^{-3}$, cuja saída é corrompida por um ruído, n_k . Nosso objetivo é encontrar, de modo adaptativo, um modelo de planta, $\hat{P}(z)$ em precisão finita utilizando aritmética de ponto fixo. Com este objetivo, utilizamos o

algoritmo SA em diferentes comprimentos de palavras. O sinal de entrada foi simulado como um sinal aleatório uniformemente distribuído, limitado no intervalo $[-1, 1]$. Como ruído utilizou um sinal de ordem 10^{-3} com distribuição uniforme de probabilidades. O sinal desejado foi posto como a soma do sinal de entrada mais ruído. O parâmetro tamanho do passo para o algoritmo SA foi $\mu_{SA} = 0,01$ e utilizamos para o parâmetro α com o valor 3. Além do bit das variáveis utilizamos aqui um bit de sinal, resultante $(b + 1)$. Foram realizadas 100 simulações de Monte Carlo, onde calculamos a média aritmética dessas simulações [Santana et al., (2006)].

5.2 Resultados

Para a aproximação da função $\tanh(e)$ em série de Taylor foi suficiente uma aproximação até o quinto grau com $\alpha = 3$, como apresentado na Fig. 5.3, dada por [Fonseca et al., (2016)]:

$$\tanh(\alpha e) = \alpha \left(\frac{2e^5}{15} - \frac{2e^3}{3} + e \right). \quad (5.9)$$

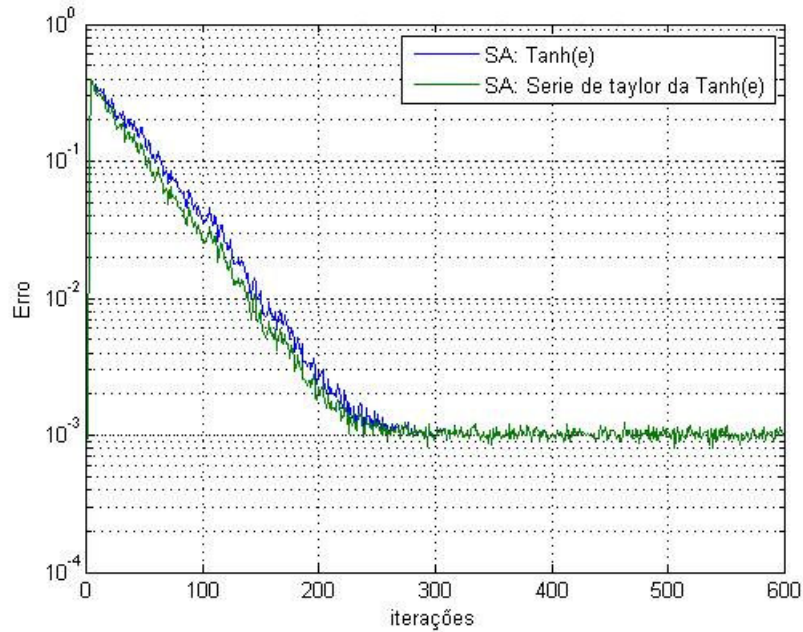


Figura 5.3: Comparação da função $\tanh(e)$ e sua função em série de Taylor(5º grau) em ponto flutuante com 64 bits.

O erro na atualização dos coeficientes internos do algoritmo adaptativo SA

quantizado pode ser dado por:

$$\mathbf{w}_{Q(k+1)} = \mathbf{w}_{Qk} + \mu\alpha \tanh(\alpha e_{Qk}) \mathbf{x}(k) - n_w, \quad (5.10)$$

onde n_w é o ruído de quantização do vetor de coeficiente $\mathbf{w}_{Qk}$ no algoritmo SA. Aqui consideramos que o ruído de quantização gerado em qualquer etapa no algoritmo, como uma variável aleatória com média zero, que é estatisticamente independente de quaisquer outros erros e/ou quantidades relacionadas no algoritmo SA.

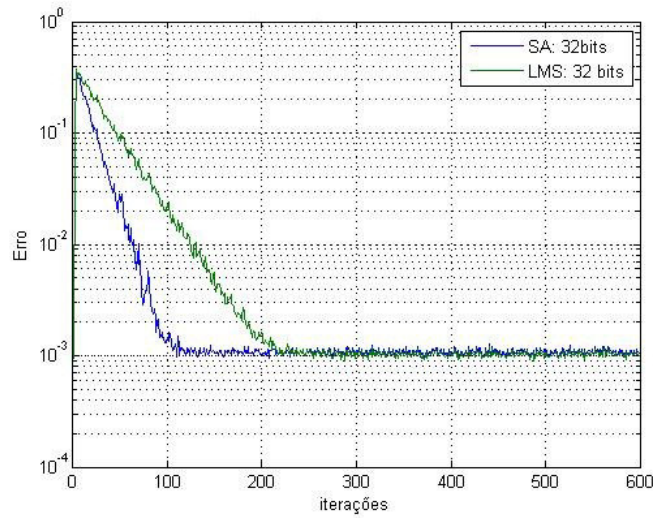


Figura 5.4: SA x LMS em ponto fixo 32 bits.

Na Fig. 5.5, comparamos o algoritmo SA e o algoritmo LMS com 32 bits de comprimento de palavra. Utilizamos o passo para o algoritmo SA $\mu = 0,03$ e passo para o LMS de modo que tivessem o mesmo desajuste, relação satisfeita por:

$$\mu_{SA} = \frac{2E[\text{sech}^2(\alpha n_k)]}{E[\tanh^2(\alpha n_k)]} \mu_{LMS}. \quad (5.11)$$

Tabela 5.1: Desajuste e MSE do SA em precisão finita

Nº de Bits b_c e b_d	Experimental	Excesso MSE
8	2,6978	$2,6978 \times 10^{-6}$
16	0,0070	$7,0851 \times 10^{-9}$
32	0,0063	$6,3681 \times 10^{-9}$

Na Fig. 5.6, comparamos o SA em três comprimentos de palavras padrões (8, 16 e 32 bits), onde observamos alguns fenômenos devido a quantização, como o aumento da velocidade de convergência nas iterações iniciais em relação a trajetória ideal,

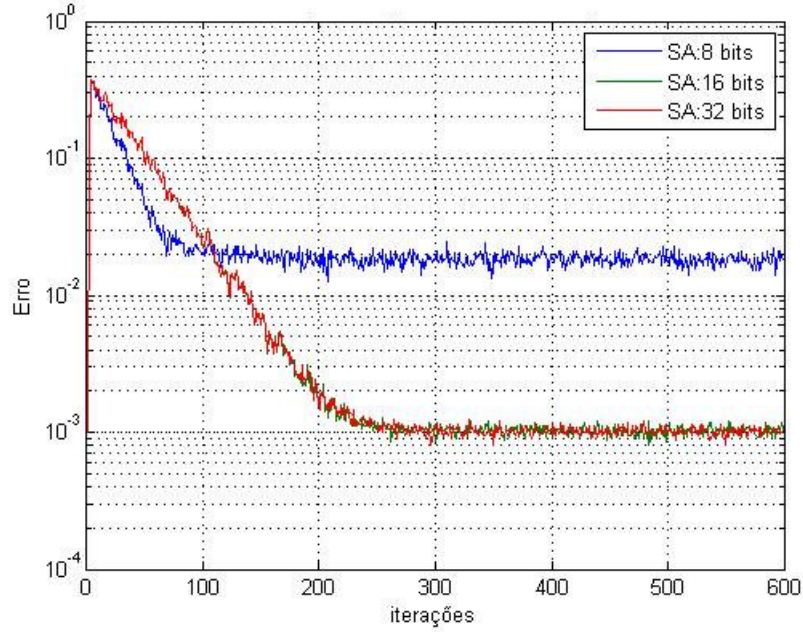


Figura 5.5: SA em diferente comprimento de palavra binária em aritmética de ponto fixo: 8 bits, 16 bits e 32 bits.

conjuntamente com um ruído devido a quantização incrementado na simulação com 8 bits, causando um certo incremento no desajuste em estado estacionário desse algoritmo [Fonseca et al., (2016)].

Na Tabela (5.1), mostramos os valores numéricos para o desajuste e excesso de erro médio quadrático experimental encontrado para os comprimentos de palavras utilizados no exemplo da Fig. 5.6, mostrando assim um valor incremental considerável para o desajuste no comprimento de palavra de 8 bits. Estes resultados mostram que para os comprimentos de palavras de 16 e 32 bits apresenta um comportamento semelhantes e que não houve um incremento considerável de ruído de quantização, podendo ser desprezível, diferentemente para o caso de comprimento de palavra de 8 bits [Fonseca et al., (2016)].

Os resultados encontrados na Fig. 5.7, onde consideramos tanto os comprimentos de palavras para os dados, quanto para os coeficientes iguais $b_d = b_c$, sugerem um aumento do nível de erro ou ruído adicional devido à quantização para alguns comprimentos de palavras (7, 8, 9, 10 e 11 bits), além de uma alteração na velocidade de convergência de sua trajetória ideal para os comprimentos de palavras (7, 8 e 9 bits).

Os resultados mostrados na Tabela (5.2), indicam os valores numéricos encontrados para o desajuste e excesso de erro médio quadrático experimental, assim como

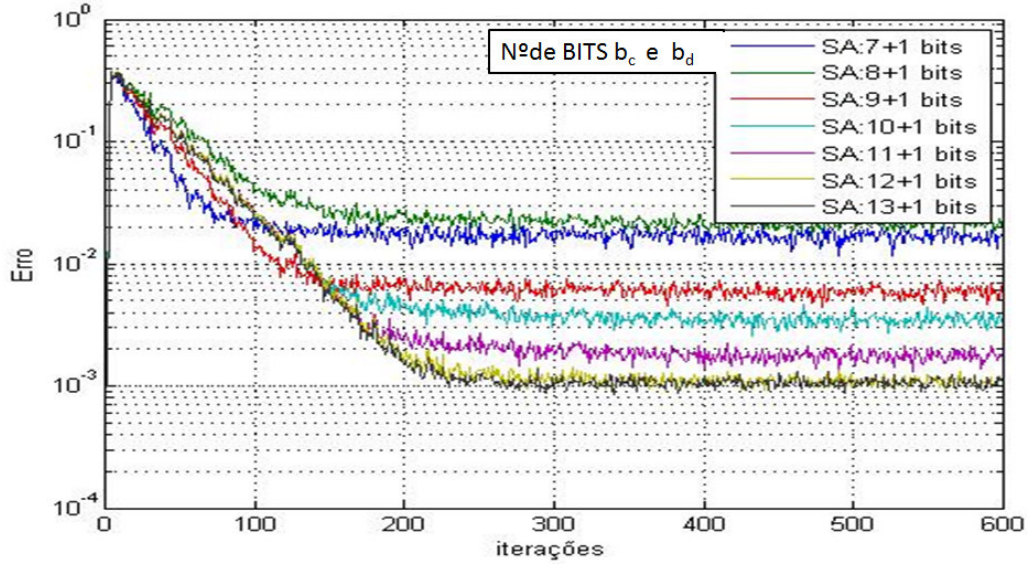


Figura 5.6: Curva de aprendizagem SA em precisão finita para $C = 0,0590$ e $n_k = 10^{-3}$ em diferentes comprimentos de palavras.

para o desajuste teórico proposto pelo nosso trabalho na subseção (4.2), onde encontramos (4.37) para o desajuste, onde podemos analisar o comportamento em estado estacionário em sua forma quantizada, modelando matematicamente o incremento do ruído devido a quantização em relação ao comprimento de palavra utilizado.

Tabela 5.2: Desajuste e MSE do SA em precisão finita para $C = 0,0590$ e $n_k = 10^{-3}$

Nº de Bits b_c e b_d	Experimental	Teorico	Excesso MSE
7	2,8978	1,9999	$2,8978 \times 10^{-6}$
8	2,9619	0,5000	$2,9619 \times 10^{-6}$
9	0,2762	0,2500	$2,7672 \times 10^{-7}$
10	0,1014	0,0625	$1,0147 \times 10^{-7}$
11	0,0304	0,0156	$3,0454 \times 10^{-8}$
12	0,0070	0,0078	$7,0851 \times 10^{-9}$
13	0,0063	0,0078	$6,3681 \times 10^{-9}$

Na Fig. 5.8 alteramos apenas o ruído do problema para ordem 10^{-2} com distribuição uniforme de probabilidades, onde observamos uma mudança em seu comportamento em relação a Fig. 5.7, apenas observou-se incremento de ruídos devido à quantização para o desajuste nos comprimentos de palavras (7 e 8 bits).

Na Tabela (5.3) observamos os valores para o desajuste experimental e de-

sajuste teórico, consideramos os valores encontrados satisfatório, uma observação interessante é que o excesso de erro médio quadrático para todos os comprimentos de palavras utilizados na tabela é muito menor em relação ao erro mínimo médio quadrático ($\xi_{min} = \sigma_k^2 = 10^{-4}$), conforme observamos na tabela para os comprimentos de palavras (7, 8, 9, 10, 11, 12 e 13 bits), dessa forma caracterizando uma previsão adequada do modelo teórico proposto.

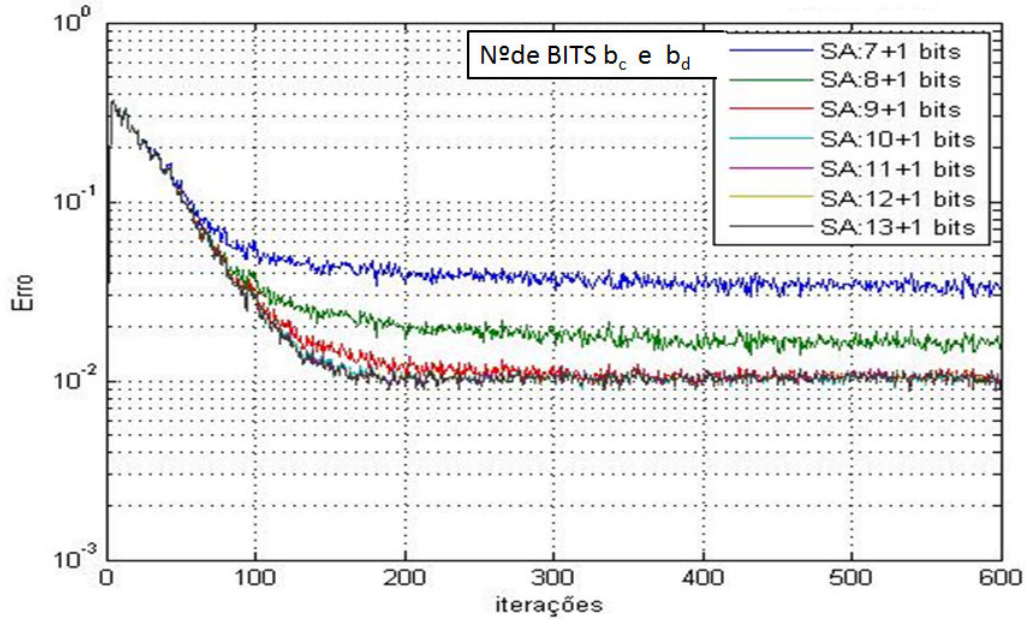


Figura 5.7: Curva de aprendizagem SA em precisão finita para $C = 0,0590$ em diferentes comprimentos de palavras

Tabela 5.3: Desajuste e MSE do SA em precisão finita para $b_d = 15bits$, $n_k = 10^{-2}$ e $C = 0,0590$

Nº de Bits b_c e b_d	Experimental	Teórico	Excesso MSE
7	0,0440	0,0625	$4,4046 \times 10^{-6}$
8	0,0109	0,0156	$1,0901 \times 10^{-6}$
9	0,0062	0,0078	$6,1948 \times 10^{-7}$
10	0,0055	0,0078	$5,5429 \times 10^{-7}$
11	0,0054	0,0078	$5,3892 \times 10^{-7}$
12	0,0055	0,0078	$5,4964 \times 10^{-7}$
13	0,0054	0,0078	$5,4266 \times 10^{-7}$

Na Fig. 5.8 alteramos apenas o ruído do problema para ordem 10^{-2} com distribuição uniforme de probabilidades, onde observamos uma mudança em seu com-

portamento em relação a Fig. 5.7, apenas observou-se incremento de ruídos devido à quantização para o desajuste nos comprimentos de palavras (7 e 8 bits) [Fonseca et al., (2016)].

Na Tabela (5.3) observamos os valores para o desajuste experimental e desajuste teórico, consideramos os valores encontrados satisfatório, uma observação interessante é que o excesso de erro médio quadrático para todos os comprimentos de palavras utilizados na tabela é muito menor em relação ao erro mínimo médio quadrático ($\xi_{min} = \sigma_k^2 = 10^{-4}$), conforme observamos na tabela para os comprimentos de palavras (7, 8, 9, 10, 11, 12 e 13 bits), dessa forma caracterizando uma previsão adequada do modelo teórico proposto.

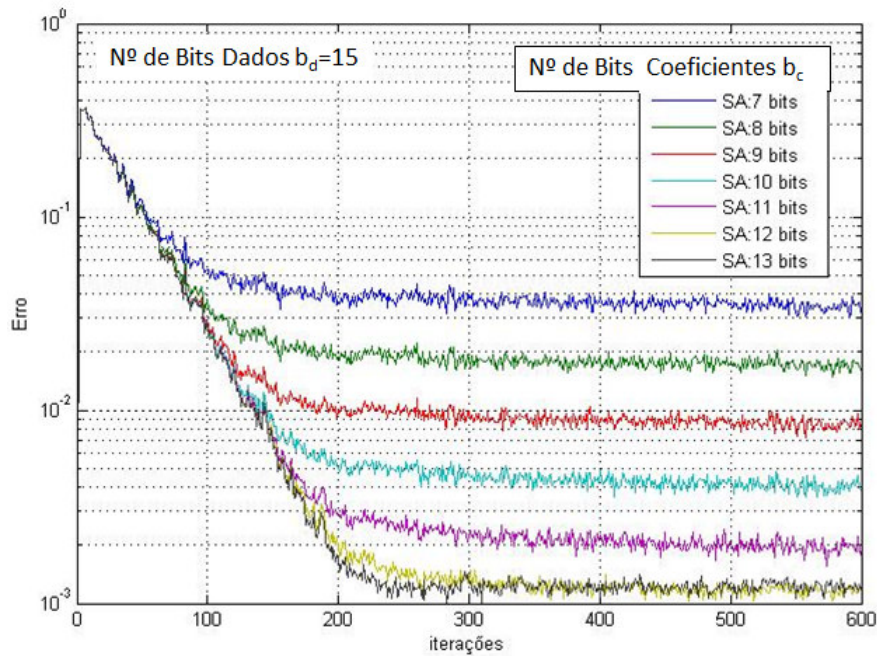


Figura 5.8: Curva de aprendizagem SA em precisão finita para $b_d = 15\text{bits}$, $n_k = 10^{-3}$ e $C = 0,0590$ em diferentes comprimentos de palavras

Na Fig. 5.9 e Tabela (5.4), deixamos fixo o número de bits para os dados b_d , neste caso 15bits , e variamos número de bits do sinal de entrada b_c para (7, 8, 9, 10 e 11 bits), onde observamos que não houve alteração em sua velocidade de convergência da sua trajetória ideal [Fonseca et al., (2016)].

Na Tabela (5.4) encontramos valores para o desajuste experimental e teórico semelhantes a Tabela (5.2), exceto para os comprimento de palavras (7 e 8 bits), onde também observamos que o excesso de erro médio quadrático está na mesma ordem do

Tabela 5.4: Desajuste e MSE do SA em precisão finita para $b_d = 15bits$, $n_k = 10^{-3}$ e $C = 0,0590$

Nº de Bits b_c	Experimental	Teorico	Excesso MSE
7	4,4654	1,9999	$4,4654 \times 10^{-6}$
8	1,1144	0,5000	$1,1145 \times 10^{-6}$
9	0,2570	0,2500	$2,5703 \times 10^{-7}$
10	0,0705	0,0625	$7,0529 \times 10^{-8}$
11	0,0187	0,0156	$1,8703 \times 10^{-8}$
12	0,0064	0,0078	$6,4395 \times 10^{-9}$
13	0,0068	0,0078	$6,7681 \times 10^{-9}$

erro mínimo médio quadrático ($\xi_{min} = \sigma_k^2 = 10^{-6}$).

6 Discussão

Observamos na Fig. 5.3, o resultado do algoritmo SA dado pela função $\tanh(e)$ e do algoritmo SA dado em série Taylor até o quinto grau, representado por (5.8), com erro de aproximação desprezível. Este resultado dessa aproximação com erro desprezível é importante, por que este será utilizado como referência na implementação do algoritmo SA dado na representação de precisão finita com aritmética de ponto fixo, devido a uma limitação dessa representação que utiliza apenas funções compostas por somas e multiplicações.

Na Fig. 5.5 comparamos o SA com o LMS em precisão finita com comprimento de palavra de 32 bits, onde mostra-se o comportamento aproximado aos algoritmos implementados em precisão infinita. Este resultado tem o objetivo de destacar a diferença de velocidade de convergência, para o mesmo desajuste entre os dois algoritmos. Assim podemos mostrar a principal característica do algoritmo SA em relação ao algoritmo LMS implementados em precisão finita.

Na Fig. 5.6 comparamos o SA em três comprimentos de palavras padrões (8, 16 e 32 bits). Este resultado é importante, por que mostra uma certa discrepância na trajetória de aprendizagem do algoritmo SA, onde observamos alguns fenômenos devido a quantização, como o aumento da velocidade de convergência nas iterações iniciais em relação a trajetória ideal, conjuntamente com um ruído devido a quantização incrementado na simulação com 8 bits, causando um certo incremento no desajuste em estado estacionário desse algoritmo.

Na Tabela (5.1), mostramos os valores numéricos para o desajuste e excesso de erro médio quadrático experimental encontrado para os comprimentos de palavras utilizados no exemplo da Fig. 5.6, mostrando assim um valor incremental considerável para o desajuste no comprimento de palavra de 8 bits. Estes resultados mostram que para os comprimentos de palavras de 16 e 32 bits apresenta um comportamento semelhantes e que não houve um incremento considerável de ruído de quantização, podendo ser desprezível, diferentemente para o caso de comprimento de palavra de 8 bits.

Os resultados encontrados na Fig. 5.7, onde consideramos tanto os comprimentos de palavras para os dados, quanto para os coeficientes iguais $b_d = b_c$, sugerem um aumento do nível de erro ou ruído adicional devido à quantização para alguns comprimentos de palavras (7, 8, 9, 10 e 11 bits), além de uma alteração na velocidade de convergência de sua trajetória ideal para os comprimentos de palavras (7, 8 e 9 bits).

Os resultados mostrados na Tabela (5.2), indicam os valores numéricos encontrados para o desajuste e excesso de erro médio quadrático experimental, assim como para o desajuste teórico proposto pelo nosso trabalho na subseção (4.2), onde encontramos (4.37) para o desajuste, onde podemos analisar o comportamento em estado estacionário em sua forma quantizada, modelando matematicamente o incremento do ruído devido a quantização em relação ao comprimento de palavra utilizado.

Para uma previsão adequada do desajuste de acordo com os pressupostos é importante observamos que para um tamanho de passo muito pequeno e o excesso de erro médio quadrático seja muito menor em relação ao erro mínimo médio quadrático ($\xi_{min} = \sigma_k^2 = 10^{-6}$), conforme observamos na tabela para os comprimentos de palavras (9, 10, 11, 12 e 13 bits).

Na Fig. 5.8 alteramos apenas o ruído do problema para ordem 10^{-2} com distribuição uniforme de probabilidades, onde observamos uma mudança em seu comportamento em relação a Fig. 5.7, apenas observou-se incremento de ruídos devido à quantização para o desajuste nos comprimentos de palavras (7 e 8 bits).

Na Tabela (5.3) observamos os valores para o desajuste experimental e desajuste teórico, consideramos os valores encontrados satisfatório, uma observação interessante é que o excesso de erro médio quadrático para todos os comprimentos de palavras utilizados na tabela é muito menor em relação ao erro mínimo médio quadrático ($\xi_{min} = \sigma_k^2 = 10^{-4}$), conforme observamos na tabela para os comprimentos de palavras (7, 8, 9, 10, 11, 12 e 13 bits), dessa forma caracterizando uma previsão adequada do modelo teórico proposto.

Outro resultado encontramos na Fig. 5.9, onde fixamos o comprimento de palavra para os dados b_d , no caso $15bits$, onde observamos que não houve alteração em sua velocidade de convergência da sua trajetória ideal. Para entender esse fenômeno desenvolvemos na subseção (4.1) um estudo de regime transitório do algoritmo SA em precisão finita. Podemos observar através de (4.23) uma devida explicação deve estar

relacionada à quantização dos sinais de entrada x_k que forma a matriz de autocorrelação $\mathbf{R}$, que modifica o valor da matriz de espalhamento que compõem o sinal de entrada, influenciando diretamente a velocidade de convergência do algoritmo.

Na Tabela (5.4) encontramos valores para o desajuste experimental e teórico semelhantes a Tabela (5.2), exceto para os comprimento de palavras (7 e 8 bits), onde também observamos que o excesso de erro médio quadrático está na mesma ordem do erro mínimo médio quadrático ($\xi_{min} = \sigma_k^2 = 10^{-6}$).

Nas Tabelas (5.2), (5.3) e (5.4) mostramos os resultados para o desajuste experimental e o desajuste teórico proposto pelo nosso trabalho, que dependem do problema envolvido, tal característica também pode ser encontrada em outros algoritmos adaptativos como o LMS. Os resultados encontrados nessas Tabelas demonstram uma aproximação com um erro aceitável dos resultados teóricos e os experimentais para o desajuste, validando neste caso o modelo proposto pelo nosso trabalho. Um desafio interessante do nosso trabalho deve estar relacionado ao entendimento da quantização do sinal de entrada com o comprimento de palavra utilizado, assim pode-se definir matematicamente o valor mínimo de comprimento de palavra para que não ocorra alteração da trajetória ideal da velocidade de convergência do algoritmo SA.

Nossa análise se aproxima de outras análises realizadas em outros algoritmos por outros autores, que utilizam algoritmos baseados na técnica do gradiente, como em [Caraiscos and Liu (1984)] que analisam o regime estacionário, especificamente o desajuste, do algoritmo LMS implementado em precisão finita, comparando resultados teóricos com os experimentais, onde verificou-se que o ruído de quantização influencia no resultado final para o desajuste, através de um incremento de um ruído de quantização, cujo valor deste depende necessariamente do tamanho de comprimento de palavra utilizado no problema. Concluiu que para um tamanho de passo muito pequeno e o excesso de erro médio quadrático muito maior que o erro mínimo médio quadrático, seus resultados teóricos prediziam os resultados experimentais e que o erro de quantização poderia ser desprezível com o aumento no comprimento de palavras.

Outra análise para o algoritmo LMS em precisão finita foi desenvolvida por Gupta (2000), este trabalho investiga um fenômeno de aumento da velocidade de convergência para alguns comprimentos de palavras específicos, desta forma este problema da velocidade de convergência tem influencia principalmente pelo comprimento de palavras

utilizado no sinal de entrada, ainda mostra resultados que definem o comprimento de palavra mínimo utilizado no sinal de entrada que não modifique a velocidade de convergência deste algoritmo.

7 Conclusões Finais e Perspectivas

Neste trabalho, analisamos o algoritmo SA em precisão finita utilizando aritmética de ponto fixo. Para isto derivamos expressões analíticas para o desajuste em precisão finita e realizamos uma aproximação em série de Taylor o que tornou possível realizar essa implementação. As expressões analíticas encontradas, mostram que a velocidade de convergência do algoritmo é controlada pela quantização do sinal de entrada que modifica a matriz de espalhamento do filtro adaptativo.

O modelo proposto para o desajuste teórico, desenvolvido neste trabalho, apresentou valores aproximados comparado com os valores dos resultados experimentais obtido através da simulação. Nesse modelo em estudo, para uma previsão adequada do desajuste pode ser necessário escolher um comprimento de palavra de forma que o excesso de erro quadrático médio seja muito menor que o erro mínimo quadrático médio. Assim, pode-se especificar uma relação entre o comprimento de palavra mínimo dos coeficientes para convergência com ruído de quantização desprezível para o algoritmo SA.

Os resultados preliminares encontrados sugerem uma estabilidade numérica para implementação desse algoritmo em precisão finita expandido em série de Taylor e a possível implementação em dispositivos de hardware desse algoritmo. Isso implica em novas alternativas para resolução de problemas de processamento de sinais encontrados em diversas áreas, como por exemplo, na medicina, militar e outros.

Os próximos rumos dessa pesquisa será continuarmos avaliando esta análise para a matriz de espalhamento do sinal de entrada, para diferentes sinais de entrada e ruídos. Como pesquisa futura sugerimos a implementação desse algoritmo em dispositivo de hardware.

Referências

- [Bershad and Bermudez,(1996)] Bershad, N. J.,& Bermudez, J., C., M., 1996. A Nonlinear analytical model for the quantized LMS algorithm - The power-of-two step size case. *IEEE Transactions on Sig. Proc.*,vol. SP-44, no. 11, pp. 2895-2900;
- [Caraiscos and Liu, (1984)] Caraiscos, C.,& Liu, B., 1984. A A roundoff error analysis of LMS adaptive algorithm. *IEEE Trans. Acoust., Speech, Signal Processing*,vol. ASSP-32, no. 1, pp. 34-41;
- [Diniz, (2013)] Diniz, P., 2013. *Adaptive filtering*, Rio de Janeiro.
- [Douglas and Meng, (1994)] Douglas, S., C.,& Meng, T., H., 1994. Stochastic gradient adaptation under general error criteria. *IEEE Transactions on Sig. Proc.*,Vol. 42, no. 6.
- [Gupta and Hero, (2000)] Gupta, R.,& Hero, A., O., 2000. Transient Behavior of fixed point LMS adaptation,. *University of Michigan, Ann Arbor*,MI 48109-2122.
- [Haykin, (2014)] Haykin, S., 2014. *Adaptive filter theory and Applications*, USA
- [Manolakis and Kogon, (2000)] Manolakis, V., K., & Kogon, S., M., 2000. Statistical and adaptive signal processing: spectral estimation, signal modeling, adaptive filtering and array processing. *McGraw Hill*, USA.
- [Maluenda et al., (2005)] Maluenda, Y., R., M.,& et al, 2005. Propriedades do algoritmo LMS operando em precisão finita. *XXII Simpósio Brasileiro de Tele.*,SBrT'05.
- [Meher and Park, (2014)] Meher, P., K.,& Park, S., Y., 2014. Area-delay-power efficient fixed-point LMS adaptive filter with low adaptation-delay. *IEEE Transactions on Very large scale int.(VLSI) Systems*,Vol. 22, no. 2.
- [Oppenheim et al., (1996)] Oppenheim, A. V., & et al, 1996. *Signals and Systems*, Prentice Hall, New Jersey.

- [Sayde, (2013)] Sayed, A. H., 2013. *Fundamentals of Adaptive Filtering*, IEEE Press, University of California.
- [Santana et al., (2006)] Santana, E., 2006. Estudo e desenvolvimento de uma família de algoritmos não lineares para filtragem adaptativa. In Olhoff, N. & Rozvany, G. I. N., eds, *UFMA*.
- [Windrow and Stearns, (1985)] Widrow, B., & Stearns, S., D., 1985. Adaptive signal processing, Prentice Hall, Englewood Cliffs. *Prentice Hall, Englewood Cliffs*.
- [Torre, (2004)] Torres, R. Coura., & p. 50, p 27., 2004. DSP e Suas Aplicações. *Prentice Hall, Englewood Cliffs*.
- [Tao et al., (1995)] Tao G., King S., Nabi-Bidhedi M., Ultrasonic wave propagation in dry and Brine-Saturated Sandstones as Function of Effective Stress: Laboratory measurements and Modeling. *Geophysical Prospecting*, vol. 43, p. 299-327, 1995.
- [Douglas and Meng, (1992)] S. C. Douglas e T. H.. Y. Meng, "The optimum scalar data nonlinearity in LMS adaptation for arbitrary i.i.d. inputs", *IEEE Trans. Signal Processing*, vol 40, n] 6, pp, 1566-1570, june 1992.
- [Wiener, (1949)] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Times Series*, with Engineering Applications, New York, Wiley 1949.
- [Walach and Widrow, (1984)] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm and its family" *IEEE Trans. Inform. Theory*, vol. IT-30, n°. 2, pp. 275-283, Mar. 1984.
- [Gibson and Gray, (1988)] J. d Gibson and S. D. Gray, "MVSE adaptive filtering subject to a constraint on MSE" *IEEE Trans. Circuits Syst.*, Sol 35, n°. 5, pp 603-608, May 1988.
- [Gersho, (1984)] A. Gersho, "Adaptive filtering with binary reinforcement" *IEEE Trans. Information Theory*, vol. IT-30, n°. 2, pp. 191-199, Mar. 1984.
- [Sherman, (1958)] S. Sherman, "Non-mean-square error criteria" *IRE Trans. Information Theory*, vol. IT4, n°. 3, pp. 125-126, 1958
- [Johnson and Rao, (1991)] D. H Johnson and P. S. Rao, "On the existence of Gaussian noise" in *Proc. Int. Conf. Acoust. , Speech, Signal Processing* (Toronto, Canda), 1991,

pp. 167301676.

[Tocci, (2013)] R. J Tocci, N. S. Widmer and G. L. Moss, "Sistemas Digitais ; princípios e aplicações, 11^a Edição, Pearson, 2013.

[Boroujeny, (2013)] B. F. Bouronjeny, "Adaptive Filters, Theory and Applications"Second Edition, USA, 2013.

[Padgett, (2009)] W. T. Padgett and D. V. Anderson, "Fixed-Point Signal Processing. Lecture 9", USA, 2009.

[Fonseca et al., (2016)] J. R.S. Fonseca, E. E. C. Santana, A. C. B. Cavalcante, R. C. S. Freire and A. K. Barros. "Finite Preciso Effect of Adaptative Algorithm Sigmoidal"Your paper 1570325509.[AMC2016]-Kota Kinabalu, Sabah -Malaysia. Dezembro, 2016

[Fonseca et al., (2017)] J. R. S. Fonseca, E. E. C. Santana, A. C. B. Cavalcante, R. C. S. Freire and A. K. Barros. "Análise do Efeito da Precisão Finita no Algoritmo Adaptativo Sigmoidal"X-EAMC- Petropolis-RJ,BRASIL. <http://www.eamc.lncc.br/> Fevereiro, 2017

Apêndices

```

%% UNIVERSIDADE FEDERAL DO MARANHÃO
%% MESTRADO DE ENGENHARIA DE ELETRICIDADE
%% LABORATORIO DE PROCESSAMENTO DA INFORMACAO BIOLOGICA
%% JOSÉ DE RIBAMAR SILVA FONSECA

%%          CODIGO : QUANTIZAÇÃO DO FILTRO ADAPTATIVO SIGMOIDAL

%% Referencias:
% Fonseca, J. R. S. et al. Dezembro 2016.

resetglobalfimath;
WordLength = 16; %numero de bits total
FractionalLength = 14; %numero de bits fracao
T1 =
numerictype('Signedness','signed','Slope',1,'Bias',0,'WordLength',WordLength,...
'FractionLength',FractionalLength);
F1 =
fimath('ProductMode','SpecifyPrecision','ProductWordLength',WordLength,...
.ProductFractionLength',FractionalLength,'SumMode','SpecifyPrecision',...
'SumWordLength',WordLength,'SumFractionLength',FractionalLength,'Overflow
Mode',... 'Saturate');

a1= 4;
A = eye(a1);
N1=fi(a1, T1, F1);

VEZ=100;
M=600;
alpha=3;
muL=1e-2;
muL1 = fi(muL, T1, F1);
J=1;
for i=1:VEZ,
    %Sinal de Entrada
    x = rand(1,M);
    x=maxize(stdize(x));
    x1= fi(x, T1, F1);

    %Ruído do Sinal
    ruido=0.001*stdize(rand(1,M));
    ruido1(i,:) = fi(ruido, T1, F1);ruido1(i,:) = ruido1(i,:)-
mean(ruido1(i,:));

    %Passo de Adaptação
    muT=0.0100;
    muT1=fi(muT, T1, F1);
    b1 = fi([0 0.2037 0.5926 0.2037], T1, F1);

    %mudar numero de bits
    D1 = filter(b1,1,x1)+ruido1(i,:);%sinal de entrada

    BIT1= 7; BIT2=8 ; BIT3= 9; BIT4=10 ; BIT5=11;  BIT6=12 ; BIT7=13;

    A1=fi(A, T1, F1);

```

```

X1=fdelay_pf(x1,A1);

[y1(i,:),e1(i,:)] = tlms_pontofixo1(X1,D1,muT1,alpha1, BIT1);
[y2(i,:),e2(i,:)] = tlms_pontofixo2(X1,D1,muT1,alpha1, BIT2);
[y3(i,:),e3(i,:)] = tlms_pontofixo3(X1,D1,muT1,alpha1, BIT3);
[y4(i,:),e4(i,:)] = tlms_pontofixo4(X1,D1,muT1,alpha1, BIT4);
[y5(i,:),e5(i,:)] = tlms_pontofixo5(X1,D1,muT1,alpha1, BIT5);
[y6(i,:),e6(i,:)] = tlms_pontofixo6(X1,D1,muT1,alpha1, BIT6);
[y7(i,:),e7(i,:)] = tlms_pontofixo7(X1,D1,muT1,alpha1, BIT7);
end

ems1= std(e1.data);

ems2= std(e2.data);
ems3= std(e3.data);
ems4= std(e4.data);
ems5= std(e5.data);
ems6= std(e6.data);
ems7= std(e7.data);

vruido= var(ruido);
Media_ruido = mean(vruido)

%Desajuste 1
vari1 = var(ems1(:,M-200:M-1));
vari2 = var(ems2(:,M-200:M-1));
vari3 = var(ems3(:,M-200:M-1));
vari4 = var(ems4(:,M-200:M-1));
vari5 = var(ems5(:,M-200:M-1));
vari6 = var(ems6(:,M-200:M-1));
vari7 = var(ems7(:,M-200:M-1));

JMCm1 = mean(vari1);
JMCm2 = mean(vari2);
JMCm3 = mean(vari3);
JMCm4 = mean(vari4);
JMCm5 = mean(vari5);
JMCm6 = mean(vari6);
JMCm7 = mean(vari7);

D_exper1 = (JMCm1 - mean(vruido))/mean(vruido)+1;
D_exper2 = (JMCm2 - mean(vruido))/mean(vruido)+1;
D_exper3 = (JMCm3 - mean(vruido))/mean(vruido)+1;
D_exper4 = (JMCm4 - mean(vruido))/mean(vruido)+1;
D_exper5 = (JMCm5 - mean(vruido))/mean(vruido)+1;
D_exper6 = (JMCm6 - mean(vruido))/mean(vruido)+1;
D_exper7 = (JMCm7 - mean(vruido))/mean(vruido)+1;

vb1bits1=((2^(-2*(BIT1)))/12)*0.0590);
vb1bits2=((2^(-2*(BIT2)))/12)*0.0590);
vb1bits3=((2^(-2*(BIT3)))/12)*0.0590);
vb1bits4=((2^(-2*(BIT4)))/12)*0.0590);
vb1bits5=((2^(-2*(BIT5)))/12)*0.0590);

```

```

vbits6=((2^(-2*(BIT6)))/12)*0.0590);
vbits7=((2^(-2*(BIT7)))/12)*0.0590);

D_teorico1 =
fi(((muT*(mean(tanh(alpha*ruido).^2)))+(vbits1)/muT)*trace(X1*X1')/M)/
(2*(mean(sech(alpha*ruido).^2))*mean(var(ruido))), T1, F1);
Figure,
t=1:M;semilogy([ems1;ems2;ems3;ems4;ems5;ems6;ems7]'); grid;
title('Curva de Aprendizagem Precisão finita Sigmoidal entrada:
15+1bits');
legend(['SA=',num2str(BIT1)], ['SA=',num2str(BIT2)],
['SA=',num2str(BIT3)], ['SA=',num2str(BIT4)], ['SA=',num2str(BIT5)],
['SA=',num2str(BIT6)], ['SA=',num2str(BIT7)]);
xlabel('iterações');
ylabel('Erro');

```