

Felipe Dennis de Resende Oliveira

**CONTRIBUIÇÕES PARA IMPLEMENTAÇÃO EFICIENTE DE
FILTROS VOLTERRA EXPLORANDO CARACTERÍSTICAS DE
*HARDWARE***

Dissertação submetida ao Programa de
Pós-Graduação em Engenharia
Elétrica da Universidade Federal de
Santa Catarina para a obtenção do
Grau de Mestre em Engenharia
Elétrica

Orientador: Prof. Dr. Eduardo Luiz
Ortiz Batista.

Coorientador: Prof. Dr. Rui Seara

Florianópolis
13 de Março de 2018

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Oliveira, Felipe Dennis de Resende
Contribuições para implementação eficiente de
filtros Volterra explorando características de
hardware / Felipe Dennis de Resende Oliveira ;
orientador, Eduardo Luiz Ortiz Batista,
coorientador, Rui Seara, 2018.
89 p.

Dissertação (mestrado) - Universidade Federal de
Santa Catarina, Centro Tecnológico, Programa de Pós
Graduação em Engenharia Elétrica, Florianópolis, 2018.

Inclui referências.

1. Engenharia Elétrica. 2. Filtros Volterra. 3.
Implementação em hardware. 4. Implementações de posto
reduzido. I. Batista, Eduardo Luiz Ortiz . II.
Seara, Rui. III. Universidade Federal de Santa
Catarina. Programa de Pós-Graduação em Engenharia
Elétrica. IV. Título.

Felipe Dennis de Resende Oliveira

**CONTRIBUIÇÕES PARA IMPLEMENTAÇÃO EFICIENTE DE
FILTROS VOLTERRA EXPLORANDO CARACTERÍSTICAS DE
HARDWARE**

Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, na área de concentração *Comunicações e Processamento de Sinais*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Florianópolis, 13 de Março de 2018



Prof. Marcelo Lobo Heldwein, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica



Prof. Eduardo Luiz Ortiz Batista, Dr.
Orientador

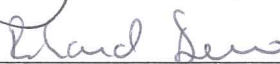


Prof. Rui Seara, Dr.
Coorientador

Banca Examinadora:



Prof. André Luís Dalcastagnê, Dr.



Prof. Richard Demo Souza, Dr.

Dedico este trabalho com todo carinho
à Maiara e aos meus pais.

AGRADECIMENTOS

Primeiramente, agradeço a Deus pela vida e por conduzir o caminho que trilhei até aqui. Agradeço também a todos que de alguma forma contribuíram para o desenvolvimento deste trabalho, em especial:

À Maiara, que é a fonte de onde retiro forças para enfrentar as dificuldades diárias e a razão pela qual busco o melhor da vida. Obrigado pelo companheirismo e por partilhar comigo os nossos sonhos.

Aos meus pais que, com muito afeto e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Aos meus irmãos: Bruno, Caio, Vanessa e Gabriel, pela amizade de sempre e para sempre.

Ao professor Eduardo Luiz Ortiz Batista, pela amizade e orientação, fundamentais para a realização deste trabalho.

Ao professor Rui Seara, pelas valiosas contribuições para este trabalho.

Aos membros da banca examinadora, que contribuíram de maneira significativa para o aprimoramento do meu trabalho.

Aos meus colegas do LINSE, pelo bom convívio durante todos esses anos.

RESUMO

Este trabalho de pesquisa apresenta uma nova estratégia de implementação de filtros Volterra com complexidade reduzida. Para tal, uma forma alternativa de representação de estruturas de posto reduzido é definida, permitindo, assim, que o fator de significância de cada um dos ramos dessas estruturas seja transferido à entrada do ramo correspondente. Com isso, torna-se possível não apenas remover os ramos menos significativos da estrutura, mas também explorar os diferentes graus de significância, visando definir tamanhos de palavra individuais a serem utilizados para implementação de cada ramo. Como resultado, são obtidas estruturas para implementação de filtros Volterra com reduções significativas de custo computacional, especialmente em implementações realizadas em FPGAs (*field-programmable gate arrays*) ou ASICs (*application-specific integrated circuits*). Também são apresentadas, neste trabalho, duas abordagens para avaliar a importância de cada ramo da estrutura de posto reduzido. Com essas abordagens, são definidos critérios que permitem identificar de maneira objetiva os ramos que não contribuem significativamente para a saída do sistema e que podem ser removidos. O custo computacional associado à estratégia de implementação proposta é avaliado através da implementação de três filtros Volterra, sendo dois com não linearidades de ordem quadrática e um de ordem cúbica. Os resultados obtidos com tais implementações confirmam a eficácia da estratégia proposta em definir estruturas de implementação para filtros Volterra com complexidade reduzida.

Palavras-chave: Filtros Volterra, implementação em *hardware*, implementações de posto reduzido.

ABSTRACT

This research work presents a new approach for the reduced-complexity implementation of Volterra filters. To this end, an alternative form to represent reduced-rank Volterra structures is defined, allowing to transfer the branch significance factor to the input of the corresponding branch. Thus, it becomes possible not only to remove the least significant branches of the structure, but also to exploit the different degrees of significance in order to define individual word lengths to be used for implementing each branch. As a result, structures for implementing Volterra filters with significant reductions of computational cost are obtained, especially in field-programmable gate arrays (FPGAs) or application-specific integrated circuits (ASICs). Two approaches to assess the importance of each branch of the reduced-rank structure are also presented in this work. These approaches allow defining objective criteria that allow identifying the branches that do not contribute significantly to the output of the system and thus can be removed. The computational cost associated with the proposed approach is evaluated by means of the implementation of three Volterra filters, two with quadratic order and one with cubic order. The results obtained in such implementations confirm the effectiveness of proposed approach in defining reduced-complexity structures to implement Volterra filters.

Keywords Volterra filters, hardware implementation, reduced-rank implementation.

LISTA DE FIGURAS

1.1	Representação em diagrama de blocos do sistema $S[x(t)]$. . .	21
2.1	Representação em blocos de um filtro Volterra de ordem P . . .	29
2.2	Estruturas de implementação dos <i>kernels</i> de (a) primeira ordem e de (b) segunda ordem.	30
2.3	Diagrama de blocos da implementação baseada em coordenadas diagonais de um <i>kernel</i> quadrático.	33
2.4	Diagrama de blocos de uma implementação baseada na decomposição genérica de um <i>kernel</i> Volterra de segunda ordem.	35
2.5	Diagrama de blocos de uma implementação baseada na SVD de um <i>kernel</i> Volterra de segunda ordem.	38
2.6	Diagrama de blocos de uma implementação com ramos paralelos baseada na SVD de um <i>kernel</i> Volterra de ordem p	39
3.1	Diagrama de blocos de uma forma alternativa de implementação da estrutura da Figura 2.5.	43
3.2	Diagrama de blocos de um dos ramos da estrutura proposta.	45
3.3	Somador sintetizado considerando a topologia <i>ripple-carry adder</i>	46
3.4	Representação simplificada de um somador com entradas de b_{sum} bits.	46
3.5	Esquema para implementação da multiplicação entre dois sinais A e B representados, respectivamente, por 4 e 3 bits.	47
3.6	Simbologias adotadas para representar as operações de multiplicação. (a) Símbolo do multiplicador genérico. (b) Símbolo do multiplicador por constante.	48
3.7	Estrutura de um filtro FIR (forma direta), utilizada para realizar a filtragem de um sinal conforme definido por (50).	49
3.8	Topologia de síntese do filtro FIR, a qual utiliza apenas uma unidade de multiplicação e acumulação.	50
5.1	Complexidade das implementações do <i>kernel</i> do Exemplo 1 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	62
5.2	Complexidade das implementações do <i>kernel</i> do Exemplo 1 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	63

5.3	Complexidade das implementações do <i>kernel</i> do Exemplo 1 implementado em FPGA e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	64
5.4	Complexidade das implementações do <i>kernel</i> do Exemplo 1 implementado em FPGA e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	65
5.5	Matriz de coeficientes do <i>kernel</i> quadrático obtido da modelagem do caminho do eco de rede em um ATA.	69
5.6	Complexidade das implementações do <i>kernel</i> do Exemplo 2 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	71
5.7	Complexidade das implementações do <i>kernel</i> do Exemplo 2 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	71
5.8	Complexidade das implementações do <i>kernel</i> do Exemplo 2 implementado em FPGA e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	72
5.9	Complexidade das implementações do <i>kernel</i> do Exemplo 2 implementado em FPGA e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	73
5.10	Diagrama de blocos para a implementação de um <i>kernel</i> de <i>p</i> -ésima ordem usando as decomposições de <i>kernels</i> com ordem elevada em <i>kernels</i> quadráticos.	75

5.11	Complexidade das implementações do <i>kernel</i> do Exemplo 3 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	78
5.12	Complexidade das implementações do <i>kernel</i> do Exemplo 3 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	78
5.13	Complexidade das implementações do <i>kernel</i> do Exemplo 3 implementado em FPGA e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	79
5.14	Complexidade das implementações do <i>kernel</i> do Exemplo 3 implementado em FPGA e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.	79

LISTA DE TABELAS

1	Custo computacional, em termos do número de somadores completos utilizados, das operações básicas necessárias para implementação de um filtro Volterra quadrático	50
2	Número de bits removidos dos ramos da estrutura de posto reduzido proposta para o caso do Exemplo 1	61
3	Quantidade de somadores completos utilizados na implementação do <i>kernel</i> do Exemplo 1 para as diferentes estruturas consideradas	62
4	Ocupação de blocos lógicos em FPGA com a implementação do <i>kernel</i> do Exemplo 1 para as diferentes estruturas consideradas	64
5	Capacidade de representação do filtro Volterra do Exemplo 1 em termos do NMSE	68
6	Número de bits removidos dos ramos da estrutura de posto reduzido proposta para o caso do Exemplo 2	70
7	Quantidade de somadores completos utilizados na implementação do <i>kernel</i> do Exemplo 2 para as diferentes estruturas consideradas	70
8	Ocupação de blocos lógicos em FPGA com a implementação do <i>kernel</i> do Exemplo 2 para as diferentes estruturas consideradas	72
9	Capacidade de representação do filtro Volterra do o Exemplo 2 em termos do NMSE	74
10	Quantidade de ramos implementados para os <i>kernels</i> quadráticos usados na implementação do filtro Volterra de terceira ordem	76
11	Valores de r_k em função dos <i>kernels</i> quadráticos obtidos da decomposição do filtro Volterra de terceira ordem do Exemplo 3	76
12	Quantidade de somadores completos utilizados na implementação do <i>kernel</i> do Exemplo 3 para as diferentes estruturas consideradas	77
13	Ocupação de blocos lógicos em FPGA com a implementação do <i>kernel</i> do Exemplo 3 para as diferentes estruturas consideradas	77
14	Capacidade de representação do filtro Volterra do Exemplo 3 em termos do NMSE	80

SUMÁRIO

1	Introdução	21
1.1	Sistemas Não Lineares	22
1.1.1	Filtros Polinomiais	22
1.2	Objetivos do Trabalho	24
1.3	Organização da Dissertação	24
2	Filtros Volterra	27
2.1	Complexidade do <i>Kernel</i>	28
2.2	Implementação com Redundância Removida	30
2.3	Implementações do <i>Kernel</i> Quadrático Baseadas em Decomposições Matriciais	31
2.3.1	Representação Matricial	32
2.3.2	Decomposição Matricial Genérica	34
2.3.3	Decomposição LU	35
2.3.4	Decomposição em Valores Singulares	36
2.3.5	Implementações de Posto Reduzido	37
2.4	Implementações de Filtros Volterra com Ordem Elevada	38
2.5	Conclusões	39
3	Estratégia para Implementar Estruturas de Posto Reduzido Eficientes Explorando Características do <i>Hardware</i>	41
3.1	Estratégia de Implementação Proposta	41
3.2	Aspectos de Implementação e Análise de Complexidade	45
3.2.1	Revisão de Topologias Básicas para Síntese Digital de Operações de Soma, Multiplicação e Filtragem	46
3.2.2	Análise de Complexidade para a Estratégia de Implementação Proposta	50
3.3	Conclusões	52
4	CrITÉrios para Remoção de Ramos em Estruturas de Posto Reduzido	53
4.1	Abordagem Numérica	53
4.2	Abordagem Energética	54
4.3	Conclusões	56
5	Resultados de Implementação	59
5.1	Exemplo 1	60
5.1.1	Descrição das Estruturas de Implementação	60
5.1.2	Resultados de Complexidade para as Implementações Consideradas	61

5.1.3	Análise da Complexidade das Implementações . . .	63
5.1.4	Capacidade de Representação das Estruturas Implementadas	67
5.2	Exemplo 2	68
5.2.1	Descrição das Estruturas de Implementação	68
5.2.2	Resultados de Complexidade para as Implementações Consideradas	69
5.2.3	Análise da Complexidade das Implementações . . .	70
5.2.4	Capacidade de Representação das Estruturas Implementadas	74
5.3	Exemplo 3	74
5.3.1	Descrição das Estruturas de Implementação	74
5.3.2	Resultados de Complexidade para as Implementações Consideradas	76
5.3.3	Análise da Complexidade das Implementações . . .	78
5.3.4	Capacidade de Representação das Estruturas Implementadas	80
5.4	Conclusões	81
6	Conclusões e Considerações Finais	83
6.1	Sumário e Discussão dos Resultados	83
6.2	Sugestões para Trabalhos Futuros	85

1 INTRODUÇÃO

Um sistema é comumente caracterizado por um operador que realiza transformações sobre um sinal de entrada x , obtendo como resultado um sinal de saída y . A relação matemática que descreve o comportamento de um sistema é

$$y = S[x], \quad (1)$$

onde $S[.]$ é o operador sistema o qual define a relação da entrada com a saída. É bem conhecido que os sinais carregam informações sobre fenômenos físicos e são funções de uma ou mais variáveis independentes. A distância percorrida por um objeto, por exemplo, pode ser caracterizada por um sinal em função do tempo, da velocidade e da aceleração. Contudo, mesmo que fenômenos físicos sejam funções de diversas variáveis, é comum considerar apenas o tempo como variável independente na modelagem de sistemas. Nesse contexto, pode-se reescrever (1) como

$$y(t) = S[x(t)], \quad (2)$$

onde $x(t)$ e $y(t)$ são, respectivamente, os sinais de entrada e saída do sistema em função do tempo. Também é comum representar sistemas através de diagramas de blocos. A Figura 1.1 ilustra o diagrama de blocos correspondente a (2).

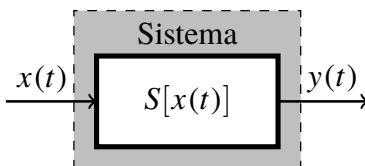


Figura 1.1: Representação em diagrama de blocos do sistema $S[x(t)]$.

Existem diversos modos para sintetizar determinado sistema de interesse. Especificamente no campo da Engenharia Elétrica, a transformação efetuada pelo operador do sistema é em geral definida por algum algoritmo sintetizado em sistema digital ou por um circuito eletrônico. Note que cada um desses modos de síntese possui natureza específica capaz de influenciar as características do processamento realizado pelo sistema. Os sistemas discretos, por exemplo, são dispositivos que usualmente operam com aritmética de ponto fixo e representam sua saída com valores discretizados. Por outro lado, os componentes eletrônicos são modelados por equações matemáticas que permitem representar suas saídas com valores contínuos. Pode-se, então, classificar os sistemas em diferentes categorias definidas a partir de suas características. Algumas das categorias usualmente encontradas em textos da

área de processamento de sinais são os sistemas [1]: contínuos ou discretos; lineares ou não lineares; variantes ou invariantes no tempo; causais ou não causais; e estáveis ou instáveis. Uma dentre essas categorias que é particularmente importante no contexto desta dissertação é a dos sistemas não lineares, cuja característica principal é a de não obedecerem ao princípio da superposição [1].

1.1 SISTEMAS NÃO LINEARES

Diversos problemas na área de processamento de sinais são resolvidos adequadamente utilizando sistemas lineares. Algumas das características que motivam o uso desse tipo de sistema são a simplicidade de implementação e o custo computacional relativamente baixo em comparação com as soluções não lineares. Contudo, em muitas aplicações práticas, fenômenos não lineares importantes estão envolvidos na geração ou processamento dos sinais, o que frequentemente torna o uso de sistemas não lineares indispensável para alcançar os níveis de desempenho desejados [2]. Alguns exemplos dessas aplicações são o controle ativo de ruído [3], o cancelamento de eco acústico [4] e a compensação das não linearidades existentes em sistemas de controle e comunicação digital [5].

A complexidade matemática associada a sistemas com características não lineares dificulta a elaboração de uma teoria geral que contemple os mais diferentes tipos de sistemas não lineares [6]. Para contornar esse problema, o estudo de sistemas não lineares é realizado separando os sistemas em classes definidas a partir de características específicas das não linearidades envolvidas. Assim, em aplicações práticas, um importante passo é escolher uma classe de sistemas não lineares que represente apropriadamente as características do problema em questão. Nesse contexto, os filtros polinomiais têm surgido como uma opção atrativa, uma vez que eles compõem uma classe de sistemas não lineares com capacidade universal de representação para sistemas discretos, causais e de memória finita [6], [7].

1.1.1 Filtros Polinomiais

A capacidade universal de representação associada aos filtros polinomiais é consequência direta do teorema da aproximação de Stone-Weierstrass, o qual estabelece que qualquer função contínua em um intervalo finito pode ser aproximada pela soma de um número finito de polinômios. Alguns dos diversos problemas reais que podem ser modelados adequadamente por meio de filtros polinomiais são [2]: i) a interação entre o movimento do barco e as ondas do oceano; ii) cancelamento de eco; iii) equalização de canais de comunicação; iv) predistorção de canais não lineares; v) modelagem de sinais da fala; e vi) detecção de bordas em processamento de imagens.

A relação entre a entrada e a saída de filtros polinomiais causais e discretos é dada por [2]

$$y(n) = \sum_{i=0}^P f_i \{x(n), x(n-1), \dots, x(n-N), y(n-1), \dots, y(n-M)\}, \quad (3)$$

onde $x(n)$ e $y(n)$ são, respectivamente, a entrada e a saída do filtro e $f_i\{\dots\}$ é a função polinomial de ordem i que atua sobre as amostras entre as chaves. Aqui, o limite superior do somatório P define a ordem do polinômio em questão. Note em (3) que os filtros polinomiais podem ser sistemas recursivos, isto é, sistemas cuja saída $y(n)$ depende de um polinômio relacionado com as amostras atrasadas dessa saída. Um problema recorrente associado a esse tipo de sistema, inclusive no contexto de filtros polinomiais, é que eles são suscetíveis à instabilidade no sentido BIBO (*bounded-input bounded-output*) [2]. Como não faz parte do escopo desta dissertação investigar as propriedades de estabilidade de sistemas, as discussões apresentadas limitam-se aos filtros polinomiais não recursivos, de modo que (3) pode ser reescrita como

$$y(n) = \sum_{i=0}^P f_i \{x(n), x(n-1), \dots, x(n-N)\}. \quad (4)$$

Os filtros polinomiais não recursivos podem ser caracterizados por uma relação de entrada e saída que toma a forma da expansão em série de Volterra truncada. Como consequência, tal expansão em série forma a base teórica dos filtros polinomiais e, além disso, herda importantes características dessa classe de sistemas não lineares [2]. Fréchet mostra em [8] que uma das características herdadas pelos filtros Volterra (filtros polinomiais representados pela expansão em série de Volterra truncada e de memória finita) é a capacidade de representar sistemas não lineares. Contudo, a representação adequada de sistemas não lineares tanto através de filtros polinomiais quanto de filtros Volterra exige um número elevado de coeficientes e, consequentemente, esse tipo de filtro possui alta complexidade de implementação. Visando contornar esse problema, nas últimas décadas, um grande esforço de pesquisa vem sendo direcionado ao desenvolvimento de diferentes abordagens para a implementação de filtros Volterra com custo computacional reduzido [2]. Nesse contexto, as abordagens denominadas de posto reduzido vêm suscitando grande interesse [9]-[14]. Essas abordagens são em geral baseadas na aplicação de decomposições matriciais ou tensoriais em representações estruturadas (na forma de matrizes ou tensores) dos coeficientes do filtro Volterra. Como resultado, estruturas de implementação compostas por ramos dispostos em paralelo são usualmente obtidas, sendo que cada um dos

ramos dessas estruturas tem um grau de significância diferente relacionado a um dos valores singulares da matriz ou tensor de coeficientes. Dessa forma, a partir da remoção de ramos menos significativos (relacionados aos menores valores singulares), torna-se possível obter implementações eficientes de filtros Volterra que apresentem reduzido custo computacional.

1.2 OBJETIVOS DO TRABALHO

O objetivo principal do presente trabalho é desenvolver implementações eficientes de filtros Volterra. Nesse contexto, uma nova estratégia de implementação baseada em estruturas de posto reduzido é discutida. A ideia central de tal abordagem é explorar os diferentes graus de significância dos ramos de uma estrutura de posto reduzido, não apenas para remover os ramos menos significativos, mas também para a especificação do correspondente *hardware* usado para implementação de cada ramo. Para tal, uma estrutura de posto reduzido convencionalmente utilizada para implementação de filtros Volterra é modificada de forma a possibilitar o uso de diferentes tamanhos de palavra na implementação dos diferentes ramos em função do grau de significância de cada ramo. Como resultado, implementações eficientes voltadas a plataformas de *hardware* específicas, tais como FPGAs (*field-programmable gate arrays*) e ASICs (*application-specific integrated circuits*), são obtidas. Em seguida, a abordagem proposta é validada a partir de estudos de caso, os quais envolvem uma avaliação da quantidade de somadores completos utilizados na implementação em *hardware* da estrutura proposta, além da quantidade de blocos lógicos de um FPGA consumidos na implementação dos filtros Volterra. É importante mencionar ainda que, por simplicidade, o foco deste trabalho está nas implementações de filtros Volterra de segunda ordem. Apesar disso, a abordagem proposta pode ser facilmente estendida para implementações de maior ordem, conforme demonstrado por resultados práticos apresentados neste trabalho. Além da nova abordagem de implementação proposta, uma contribuição envolvendo a definição de critérios para a escolha dos ramos da estrutura de posto reduzido que devem ser desprezados é apresentada neste trabalho.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

O restante desta dissertação é organizado conforme descrito a seguir. No Capítulo 2 são apresentados conceitos fundamentais dos filtros Volterra relacionados ao contexto do trabalho, como é o caso das implementações baseadas em decomposições matriciais. A estratégia proposta para implementação de filtros Volterra e uma descrição envolvendo os aspectos considerados para implementação da estrutura definida com tal estratégia são apresentados no Capítulo 3. No Capítulo 4 são definidos dois critérios para remoção dos ramos

das estruturas de posto reduzido, os quais são utilizados no Capítulo 5 para obtenção dos resultados das implementações realizadas. Essas implementações têm como principal objetivo avaliar a redução de complexidade obtida com a estratégia proposta através de uma análise comparativa com outras implementações usualmente discutidas na literatura. O Capítulo 5 também traz uma discussão acerca da capacidade das estruturas definidas pela estratégia de implementação proposta em representar precisamente alguns exemplos de filtros Volterras. Finalmente, no Capítulo 6 as conclusões desta dissertação e as propostas para trabalhos futuros são apresentadas.

2 FILTROS VOLTERRA

As expansões em séries têm se mostrado bastante eficazes para a representação de certos tipos de sistemas não lineares. Os sistemas sem memória, por exemplo, sob condições de operação apropriadas, são representados adequadamente pela expansão em série de Taylor [2]. A relação de entrada e saída dessa série centrada em zero é dada por

$$y(t) = \sum_{p=0}^{\infty} c_p x^p(t), \quad (5)$$

onde c_p é o p -ésimo coeficiente da série. No caso mais geral, em que se deseja representar sistemas não lineares com memória, a expansão em série de Volterra mostra-se uma ferramenta matemática mais adequada. Os sistemas não lineares com memória, quando contínuos, para serem representados adequadamente por uma expansão em série de Volterra, devem também ser [8]: invariantes no tempo; estáveis no sentido BIBO; de memória finita; e não apresentar discontinuidades na saída. Para representar sistemas não lineares discretos, a série de Volterra também se mostra uma ferramenta abrangente e as condições do sistema para sua representação adequada são: estável no sentido BIBO; causal; e de memória finita [6], [7]. Nesta dissertação, a discussão do uso da expansão em série de Volterra limita-se ao caso dos sistemas discretos, uma vez que o objetivo aqui é propor uma estrutura de implementação eficiente para esse tipo de sistema. Contudo, muitos dos resultados aqui apresentados são facilmente estendidos ao caso dos sistemas contínuos [2].

A caracterização de um sistema discreto não linear através da expansão em série de Volterra é realizada por meio de um conjunto de *kernels* (núcleos), cada um relacionado a uma certa ordem de não linearidade polinomial [2]. A saída $y(n)$ do sistema é então obtida somando as saídas dos *kernels* de diferentes ordens, isto é,

$$y(n) = y_0 + \sum_{p=1}^{\infty} y_p(n) \quad (6)$$

onde y_0 é um coeficiente constante e $y_p(n)$, a saída do *kernel* de ordem p . A relação matemática que define a saída dos *kernels* de ordem igual ou superior a 1 é

$$y_p(n) = \sum_{m_1=-\infty}^{\infty} \cdots \sum_{m_p=-\infty}^{\infty} h_p(m_1, \dots, m_p) \prod_{k=1}^p x(n - m_k), \quad (7)$$

com p representando a ordem do *kernel*, $x(n)$ o sinal de entrada do sistema e $h_p(m_1, \dots, m_p)$ os coeficientes de ordem p .

É possível observar de (6) que a saída do sistema $y(n)$ é definida considerando a contribuição de infinitos *kernels*. Como consequência, a representação de um sistema não linear no contexto discreto por meio da expansão em série de Volterra exige o truncamento do número de *kernels* usados, restringindo, para isso, a um valor finito P o limite superior do somatório em (6). Além disso, é possível observar em (7) que cada *kernel* possui infinitos coeficientes e, conseqüentemente, sua implementação requer: i) capacidade infinita de memória, ou ii) sistemas recursivos. Conforme dito anteriormente, este último tipo de sistema é desconsiderado no presente trabalho por questões de instabilidade. Sobre a capacidade infinita de memória, é bem conhecido que nenhum sistema discreto atual atende a tal requerimento. Contudo, em muitas aplicações práticas, o número total de coeficientes necessário para representar adequadamente uma dada não linearidade é finito, possibilitando, assim, limitar o tamanho de memória destinado a cada *kernel* a um valor finito N . Desse modo, para viabilizar a implementação da série de Volterra em sistemas discretos, a relação matemática entre a entrada e a saída dos filtros Volterra, bem como a dos seus *kernels* correspondentes, é definida, respectivamente, por

$$y(n) = y_0 + \sum_{p=1}^P y_p(n), \quad (8)$$

e

$$y_p(n) = \sum_{m_1=0}^{N-1} \cdots \sum_{m_p=0}^{N-1} h_p(m_1, \dots, m_p) \prod_{k=1}^p x(n - m_k). \quad (9)$$

Note de (8) que a saída de um dado filtro Volterra pode ser representada como a combinação de P sistemas em paralelo somadas ao termo constante. Tal representação é ilustrada na Figura 2.1 na forma de diagrama de blocos. O primeiro bloco da figura é caracterizado pelo *kernel* de primeira ordem (isto é, com $p = 1$) e é equivalente a um filtro FIR tratando-se, assim, de um *kernel* linear. Por outro lado, os demais blocos são caracterizados pelos *kernels* de ordem superior a 1 (com $p \geq 2$), carregando as não linearidades do filtro Volterra. É interessante observar da Figura 2.1 que a estruturação do filtro Volterra na forma de diagramas de blocos facilita discriminar os componentes do filtro responsáveis por cada ordem de não linearidade. Nesse contexto, pode-se trabalhar separadamente com os blocos com maior número de coeficientes e, até mesmo, desconsiderar determinados blocos dependendo da aplicação [15].

2.1 COMPLEXIDADE DO KERNEL

Mesmo considerando o truncamento da série de Volterra e o uso de uma memória com tamanho finito N , a complexidade computacional requerida

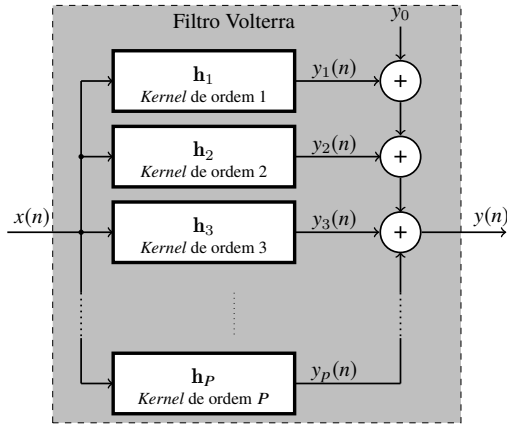


Figura 2.1: Representação em blocos de um filtro Volterra de ordem P .

para a implementação dos filtros Volterra é ainda bastante elevada. É evidente, por inspeção em (9), que o número de coeficientes associado a cada *kernel* cresce exponencialmente com a ordem e com o tamanho da memória. Especificamente para um *kernel* de ordem p , a quantidade de coeficientes requerida para implementação é dada por

$$N_p = N^p. \quad (10)$$

Assim, o número total de coeficientes de um filtro Volterra com P *kernels* é obtido por

$$N_T = \sum_{i=1}^P N_i = \frac{N_{P+1} - N_1}{N_1 - 1}. \quad (11)$$

A relação exponencial, evidenciada em (10) e (11), entre o número de coeficientes do *kernel* e o seu tamanho de memória é um obstáculo que frequentemente restringe a implementação prática de filtros Volterra. Tal restrição se deve ao fato de que o número de coeficientes e a complexidade de implementação do sistema correspondente são diretamente relacionados. No caso de sistemas de primeira ordem [$p = 1$ em (9)], por exemplo, ao se considerar como parâmetro de complexidade o número de multiplicações realizadas por amostra de saída, um filtro FIR com 3 coeficientes terá custo de implementação três vezes menor do que o custo associado a um filtro com 9 coeficientes. Ao se tratar da implementação de *kernels* com ordens superiores (maiores do que 1), o acréscimo em complexidade computacional é ainda mais elevado. Nesse caso, além do maior número de coeficientes que

os *kernels* de ordens superiores possuem para um mesmo tamanho de memória [veja (10)], são necessárias operações de multiplicação adicionais entre as amostras do sinal de entrada para o cálculo de saída do *kernel*. As estruturas de implementação de (9) para $N = 3$, considerando *kernels* de primeira e de segunda ordens, são ilustradas na forma de diagramas de blocos na Figura 2.2. O aumento excessivo em complexidade computacional necessário para a implementação de um *kernel* de segunda ordem, em comparação com um de primeira ordem, fica evidente ao se inspecionar essa figura. Note que são necessárias três multiplicações para calcular uma amostra de saída do *kernel* de primeira ordem. Já o *kernel* de segunda ordem, tem uma amostra de sua saída calculada a partir de 18 multiplicações; ou seja, em termos de multiplicações realizadas, para um tamanho de memória igual a três, o *kernel* de segunda ordem possui um custo computacional seis vezes superior ao de um *kernel* de primeira ordem.

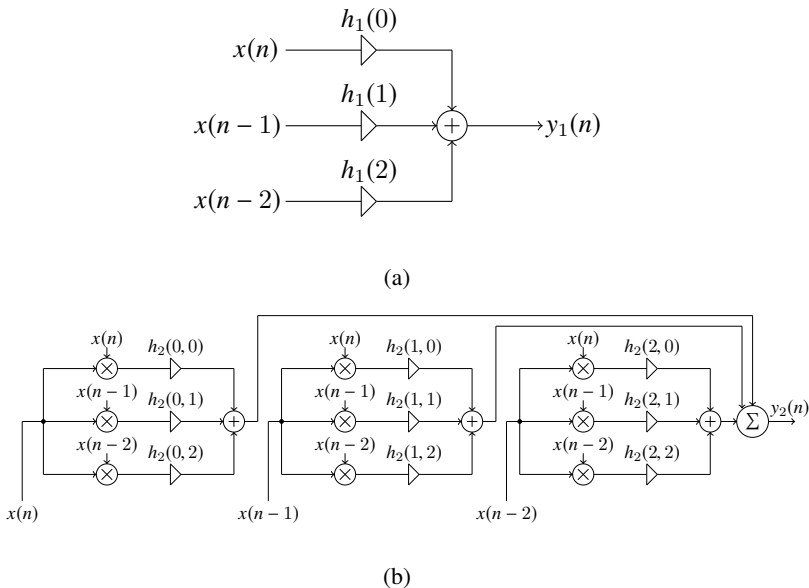


Figura 2.2: Estruturas de implementação dos *kernels* de (a) primeira ordem e de (b) segunda ordem.

2.2 IMPLEMENTAÇÃO COM REDUNDÂNCIA REMOVIDA

Devido ao excessivo custo de implementação associado ao filtro Volterra, diferentes formas para representar os *kernels* de ordens superiores a 1 são investigadas. Uma forma de representação conveniente, que apresenta redu-

ção no número de coeficientes sem perda de generalidade, é a estrutura com redundância removida. Pode-se observar, a partir de (7), que os coeficientes $h_p(m_1, \dots, m_p)$ com índices m_1, \dots, m_p permutados são multiplicadores de um mesmo produto cruzado de amostras do sinal de entrada para o cálculo de $y_p(n)$. No diagrama de blocos ilustrado na Figura 2.2(b), por exemplo, os coeficientes $h_2(0, 1)$ e $h_2(1, 0)$ do *kernel* de segunda ordem são ambos multiplicadores de $x(n)x(n-1)$ no cálculo de $y_2(n)$. Assim, combinando os coeficientes com índices permutados em um único coeficiente denotado por $\underline{h}_p(m_1, m_2, \dots, m_p)$ [por exemplo, $\underline{h}_2(0, 1) = h_2(0, 1) + h_2(1, 0)$], é possível obter uma implementação equivalente do filtro Volterra com número reduzido de coeficientes. Em tal implementação, conhecida como implementação com redundância removida ou implementação triangular, a relação de entrada e saída do *kernel* de ordem p é dada por

$$y_p(n) = \sum_{m_1=0}^{N-1} \cdots \sum_{m_p=m_p-1}^{N-1} \underline{h}_p(m_1, \dots, m_p) \prod_{k=1}^p x(n - m_k). \quad (12)$$

O número de coeficientes do *kernel* de ordem p da implementação com redundância removida é dado por

$$\underline{N}_p = \frac{(N + p - 1)!}{(N - 1)!p!}. \quad (13)$$

Mesmo que a estrutura com redundância removida possua menos coeficientes do que a do filtro Volterra padrão [definido por (9)], a redução da complexidade obtida é comumente insuficiente para viabilização do uso do filtro Volterra em certas situações práticas [16]. Como consequência, a aplicação de tal filtro muitas vezes é feita a partir de implementações mais eficientes em termos de complexidade, como é o caso das implementações de posto reduzido, as quais são desenvolvidas a partir de implementações baseadas em decomposições matriciais.

2.3 IMPLEMENTAÇÕES DO *KERNEL* QUADRÁTICO BASEADAS EM DECOMPOSIÇÕES MATRICIAIS

Por questões de simplicidade, a discussão envolvendo implementações de filtros Volterra com complexidade computacional reduzida é introduzida considerando apenas o *kernel* quadrático. Uma forma alternativa e mais adequada para representar a relação entrada e saída desse tipo de *kernel*, a qual geralmente leva a estruturas de implementação eficientes, é a representação matricial.

2.3.1 Representação Matricial

Na representação matricial, os índices m_1 e m_2 dos coeficientes do *kernel* quadrático são considerados como coordenadas de um plano cartesiano. Desse modo, tais coeficientes podem ser organizados na forma de uma matriz \mathbf{H}_2 e ordenados de acordo com o índice associado a cada coeficiente, isto é,

$$\mathbf{H}_2 = \begin{bmatrix} h_2(0,0) & h_2(0,1) & \cdots & h_2(0,N-1) \\ h_2(1,0) & h_2(1,1) & \cdots & h_2(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ h_2(N-1,0) & h_2(N-1,1) & \cdots & h_2(N-1,N-1) \end{bmatrix}. \quad (14)$$

A partir de (14) e da descrição das amostras de entrada na forma vetorial, definida como

$$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \dots \quad x(n-N+1)]^T, \quad (15)$$

a relação entrada e saída do *kernel* de segunda ordem, obtida fazendo $p = 2$ em (9), pode ser redefinida por

$$y_2(n) = \mathbf{x}^T(n) \mathbf{H}_2 \mathbf{x}(n). \quad (16)$$

Assim como a saída do *kernel* de segunda ordem padrão, obtido diretamente da expansão em série de Volterra, o *kernel* com redundância removida também pode ser representado na forma matricial. Nesse caso, a matriz de coeficientes é definida como

$$\underline{\mathbf{H}}_2 = \begin{bmatrix} \underline{h}_2(0,0) & \underline{h}_2(0,1) & \cdots & \underline{h}_2(0,N-1) \\ 0 & \underline{h}_2(1,1) & \cdots & \underline{h}_2(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \underline{h}_2(N-1,N-1) \end{bmatrix}, \quad (17)$$

onde

$$\underline{h}_2(i,j) = \begin{cases} h_2(i,j) + h_2(j,i) & \text{para } j > i \\ h_2(i,j) & \text{para } j = i \\ 0 & \text{para } j < i \end{cases} \quad (18)$$

Por meio de (17) e do vetor de entrada definido em (15), a saída do *kernel* de segunda ordem também pode ser representada por

$$y_2(n) = \mathbf{x}^T(n) \underline{\mathbf{H}}_2 \mathbf{x}(n). \quad (19)$$

Neste trabalho, utiliza-se a estrutura obtida por meio das implementações baseadas em coordenadas diagonais discutidas em [6] para representar (19).

A estrutura do *kernel* triangular resultante desse tipo de implementação é formada por N ramos em paralelo, conforme ilustrado na Figura 2.3. Cada um desses ramos é composto por um filtro FIR com sinal de entrada obtido pelo produto entre amostras atuais e atrasadas de $x(n)$. Além disso, o vetor de coeficientes do i -ésimo filtro FIR $\hat{\mathbf{h}}_{2,i}$ é definido como os elementos da i -ésima diagonal de \mathbf{H}_2 , isto é,

$$\hat{\mathbf{h}}_{2,i} = [h_2(0, i) \quad h_2(1, i + 1) \quad \cdots \quad h_2(N - 1 - i, N - 1)]^T. \quad (20)$$

Nesse contexto, a quantidade de coeficientes dos filtros FIR segue uma relação decrescente em função da posição do ramo. É importante ressaltar que não há perda de generalidade em se considerar a matriz triangular \mathbf{H}_2 no cálculo da saída do *kernel* de segunda ordem [2]. De modo geral, qualquer matriz genérica \mathbf{H}_2^* cujos coeficientes são definidos a partir de \mathbf{H}_2 , respeitando a seguinte relação

$$h_2^*(i, j) + h_2^*(j, i) = h_2(i, j) + h_2(j, i), \quad (21)$$

para $i, j = 0, 1, \dots, N - 1$, pode ser usada para calcular a relação entrada e saída definida por (16) sem perda de generalidade, isto é,

$$y_2(n) = \mathbf{x}^T(n) \mathbf{H}_2^* \mathbf{x}(n) = \mathbf{x}^T(n) \mathbf{H}_2 \mathbf{x}(n). \quad (22)$$

A possibilidade de representar a entrada e saída de qualquer *kernel* quadrático com a matriz genérica \mathbf{H}_2^* , respeitando as condições definidas em (21), permite a obtenção de implementações mais eficientes do *kernel* em questão. Para isso, a matriz \mathbf{H}_2^* é reorganizada de modo que suas características sejam adequadas para que estruturas de implementação eficientes sejam obtidas a partir de decomposições matriciais. Algumas das decomposições

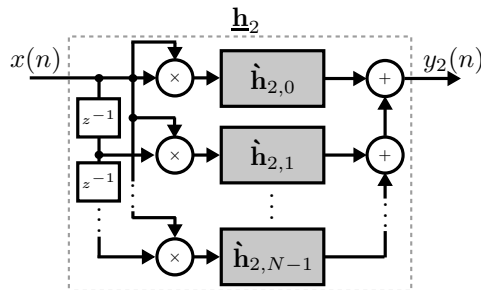


Figura 2.3: Diagrama de blocos da implementação baseada em coordenadas diagonais de um *kernel* quadrático.

matriciais usualmente consideradas no contexto de filtros Volterra são a decomposição LU (*lower-upper*) e a decomposição em valores singulares (SVD – *Singular Value Decomposition*).

2.3.2 Decomposição Matricial Genérica

É bem conhecido da álgebra linear que qualquer matriz \mathbf{A} com dimensões $M \times M$ pode ser decomposta na forma do somatório de P_A matrizes de posto unitário formadas pelo produto externo de dois vetores. Assim,

$$\mathbf{A} = \sum_{i=0}^{P_A-1} q_i \mathbf{r}_{e,i} \mathbf{r}_{d,i}^T, \quad (23)$$

onde P_A é o posto da matriz \mathbf{A} , $\mathbf{r}_{e,i}$ e $\mathbf{r}_{d,i}$ são vetores coluna com M elementos cada um e formadores da i -ésima matriz de posto unitário e, finalmente, q_i é um valor escalar real usado para multiplicar a matriz resultante do produto externo entre $\mathbf{r}_{e,i}$ e $\mathbf{r}_{d,i}^T$. Uma interessante característica das decomposições matriciais é que quando a matriz decomposta é simétrica, tem-se que $\mathbf{r}_{e,i} = \mathbf{r}_{d,i}$, reduzindo (23) a

$$\mathbf{A} = \sum_{i=0}^{P_A-1} q_i \mathbf{r}_i \mathbf{r}_i^T. \quad (24)$$

No contexto de filtros Volterra de segunda ordem, pode-se sempre considerar, sem perda de generalidade, o uso de matrizes simétricas para a aplicação de determinada decomposição matricial. Nos casos em que \mathbf{H}_2 não é simétrica, é possível aplicar a decomposição matricial em uma matriz $\bar{\mathbf{H}}_2$ alternativa e simétrica preservando a relação entrada e saída do filtro Volterra quadrático, desde que a relação definida em (21) seja respeitada. Para isso, os coeficientes de $\bar{\mathbf{H}}_2$ são definidos como

$$\bar{h}_2(i, j) = \frac{h_2(i, j) + h_2(j, i)}{2}, \quad (25)$$

com $[i, j \in \mathbb{N} \mid 0 \leq i, j \leq N - 1]$. Assim, aplicando a decomposição matricial genérica em $\bar{\mathbf{H}}_2$, tem-se que

$$\bar{\mathbf{H}}_2 = \sum_{i=0}^{P_{\bar{\mathbf{H}}_2}-1} q_i \mathbf{r}_i \mathbf{r}_i^T, \quad (26)$$

e, então, pode-se representar a relação entrada e saída do filtro Volterra qua-

drático substituindo (26) em (22), obtendo

$$\begin{aligned}
 y_2(n) &= \mathbf{x}^T(n) \sum_{i=0}^{P_{\bar{H}_2}-1} [q_i \mathbf{r}_i \mathbf{r}_i^T] \mathbf{x}(n) \\
 &= \sum_{i=0}^{P_{\bar{H}_2}-1} [q_i \mathbf{x}^T(n) \mathbf{r}_i \mathbf{r}_i^T \mathbf{x}(n)] \\
 &= \sum_{i=0}^{P_{\bar{H}_2}-1} q_i [\mathbf{x}^T(n) \mathbf{r}_i] [\mathbf{r}_i^T \mathbf{x}(n)] \\
 &= \sum_{i=0}^{P_{\bar{H}_2}-1} q_i [\mathbf{x}^T(n) \mathbf{r}_i]^2.
 \end{aligned} \tag{27}$$

Levando em conta que $\mathbf{x}^T(n) \mathbf{r}_i$ corresponde à relação de entrada e saída de um filtro FIR com vetor de coeficientes \mathbf{r}_i , é possível concluir, a partir de (27), que o *kernel* Volterra de segunda ordem pode ser implementado na forma da estrutura com ramos paralelos ilustrada na Figura 2.4. Cada ramo dessa estrutura é composto por um filtro FIR cuja saída é elevada ao quadrado e então multiplicada por um dos escalares obtidos a partir da decomposição de $\bar{\mathbf{H}}_2$.

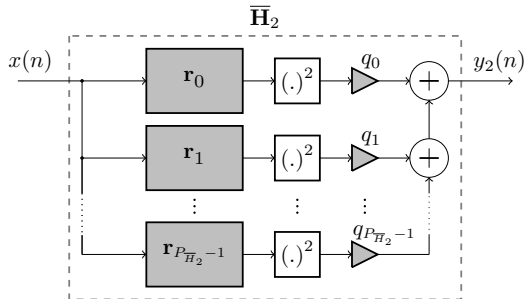


Figura 2.4: Diagrama de blocos de uma implementação baseada na decomposição genérica de um *kernel* Volterra de segunda ordem.

2.3.3 Decomposição LU

A decomposição LU é um tipo específico de decomposição matricial na qual o i -ésimo vetor usado na geração das matrizes de posto unitário contém i elementos iguais a zero. Especificamente, aplicando a decomposição LU na

matriz simétrica $\overline{\mathbf{H}}_2$, sob certas condições [2], obtém-se

$$\overline{\mathbf{H}}_2 = \sum_{i=0}^{P_{\overline{\mathbf{H}}_2}-1} d_i \mathbf{l}_i \mathbf{l}_i^T, \quad (28)$$

onde d_i é um escalar multiplicador da i -ésima matriz de posto unitário e \mathbf{l}_i é um vetor formado por

$$\mathbf{l}_i = [0 \cdots 0 \quad l_i(0) \quad l_i(1) \cdots l_i(N-i-1)]^T, \quad (29)$$

com N representando o tamanho de memória do *kernel* quadrático. A estrutura de implementação do filtro Volterra quadrático, obtida por meio da decomposição LU, é definida substituindo (28) em (22), o que resulta em

$$y_2(n) = \sum_{i=0}^{P_{\overline{\mathbf{H}}_2}-1} d_i [\mathbf{x}^T(n) \mathbf{l}_i]^2. \quad (30)$$

Assim como no caso da decomposição genérica, a estrutura de implementação do filtro Volterra quadrático, quando obtida por meio da decomposição LU, é constituída por $P_{\overline{\mathbf{H}}_2}$ ramos dispostos em paralelo. Contudo, neste último tipo de decomposição matricial, o filtro FIR do i -ésimo ramo possui $N-i$ coeficientes, ou seja, i coeficientes a menos do que o i -ésimo ramo da estrutura obtida considerando a decomposição matricial genérica. Desse modo, a complexidade de implementação do filtro Volterra quadrático com a estrutura definida por (30) é menor do que a definida por (16).

2.3.4 Decomposição em Valores Singulares

Uma importante decomposição matricial no contexto dos filtros Volterra é a decomposição SVD. Nessa decomposição, os escalares multiplicadores e os vetores de coeficientes dos filtros FIR associados aos ramos paralelos (veja Figura 2.4), são, respectivamente, os valores singulares e os vetores singulares da matriz sobre a qual a decomposição SVD foi aplicada. No caso da matriz simétrica $\overline{\mathbf{H}}_2$, a decomposição SVD resulta em

$$\overline{\mathbf{H}}_2 = \sum_{i=0}^{P_{\overline{\mathbf{H}}_2}-1} \lambda_i \mathbf{h}_{2,i} \mathbf{h}_{2,i}^T, \quad (31)$$

onde λ_i e $\mathbf{h}_{2,i}$ representam o i -ésimo valor singular e o i -ésimo vetor singular de $\overline{\mathbf{H}}_2$, respectivamente. Agora, substituindo (31) em (22), obtém-se [6]

$$y_2(n) = \sum_{i=0}^{P_{\overline{\mathbf{H}}_2}-1} \lambda_i [\mathbf{x}^T(n) \mathbf{h}_{2,i}]^2. \quad (32)$$

Comparando (27), (30) e (32) pode-se concluir que todas as decomposições matriciais consideradas levam a estruturas similares para implementação dos filtros Volterra quadráticos. Tais estruturas são constituídas por $P_{\overline{H}_2}$ ramos dispostos em paralelo, onde $P_{\overline{H}_2}$ representa o posto de \overline{H}_2 . Uma vez que o tamanho do posto da matriz de coeficientes define a quantidade de ramos dessa estrutura, *kernels* quadráticos com matrizes de posto significativamente inferior a seu tamanho de memória apresentam reduzido número de ramos e, como consequência, redução de complexidade computacional. Contudo, para as matrizes de coeficientes com posto completo, a complexidade computacional associada à estrutura de implementação com ramos paralelos é semelhante àquela obtida diretamente do *kernel* padrão do filtro Volterra quadrático [2]. A obtenção de estruturas de implementação com custo computacional reduzido, mesmo quando H_2 possui posto completo, envolve a aplicação de decomposições matriciais específicas, as quais geralmente levam a uma redução do número de coeficientes necessário para representar adequadamente o *kernel* em questão.

2.3.5 Implementações de Posto Reduzido

As implementações de posto reduzido de filtros Volterra têm sido consideradas com sucesso em muitos trabalhos de pesquisa encontrados na literatura [9]-[14]. Essas implementações são em geral baseadas na representação da relação de entrada e saída de um dado *kernel* em função de matrizes de coeficientes. Então, decomposições matriciais, tal como a decomposição SVD, são aplicadas em tais matrizes, levando a formas alternativas de representação da relação de entrada e saída do *kernel* considerado. Essas formas alternativas em geral resultam em estruturas de implementação compostas por ramos com diferentes graus de significância dispostos em paralelo. Assim, torna-se possível a obtenção de estruturas de implementação com complexidade computacional reduzida a partir da remoção dos ramos com menor grau de significância [9]-[14].

No caso das implementações de posto reduzido obtidas por meio da decomposição SVD, conforme definido em (32), é bem conhecido que os vetores singulares $\mathbf{h}_{2,i}$ possuem norma unitária. Como consequência, o grau de significância de cada ramo da estrutura é determinado pela magnitude do valor singular λ_i correspondente. Então, desconsiderando os ramos relacionados aos valores singulares de menor magnitude, torna-se possível a obtenção de uma estrutura de implementação eficiente com redução de custo computacional. Especificamente, pode-se truncar o somatório em (32) a um valor P_R , com $P_R < P_{\overline{H}_2}$, sem grandes perdas em capacidade de representação e

a saída do filtro Volterra quadrático é calculada por

$$y_2(n) = \sum_{i=0}^{P_R-1} \lambda_i [\mathbf{x}^T(n) \mathbf{h}_{2,i}]^2. \quad (33)$$

A estrutura de implementação de (33) é detalhada na forma de um diagrama de blocos ilustrado na Figura 2.5. Note que tal estrutura é bastante similar à definida pela decomposição matricial genérica, sendo a principal diferença o número de ramos implementados.

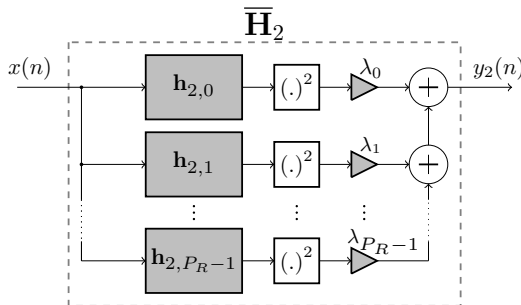


Figura 2.5: Diagrama de blocos de uma implementação baseada na SVD de um *kernel* Volterra de segunda ordem.

As demais abordagens de implementação com posto reduzido de filtros Volterra encontradas na literatura são obtidas associando diferentes formas de representação da relação de entrada e saída de um *kernel* com diferentes tipos de decomposições matriciais ou tensoriais [9]-[14]. De maneira geral, tais abordagens também resultam em estruturas de implementação compostas por ramos paralelos que apresentam diferentes graus de significância.

2.4 IMPLEMENTAÇÕES DE FILTROS VOLTERRA COM ORDEM ELE-VADA

Na literatura dos filtros Volterra, são tratadas diferentes maneiras de entender os resultados obtidos com o *kernel* quadrático para *kernels* de ordens superiores [13], [17]-[19]. Em [13], por exemplo, é discutida uma abordagem baseada na decomposição SVD para definir estruturas de implementação com posto reduzido de *kernels* homogêneos de ordem p . Tal abordagem usualmente considera definições não triviais de matrizes de coeficientes. De modo específico, o vetor de entrada e a matriz de coeficientes do filtro Volterra de ordem p são organizados de forma a representar precisamente a relação entrada e saída dada por (12) e então a decomposição SVD é aplicada a essa

matriz. Como resultado, uma estrutura de implementação com $P_{\overline{H}_p}$ ramos dispostos em paralelo é obtida, conforme ilustrado na Figura 2.6.

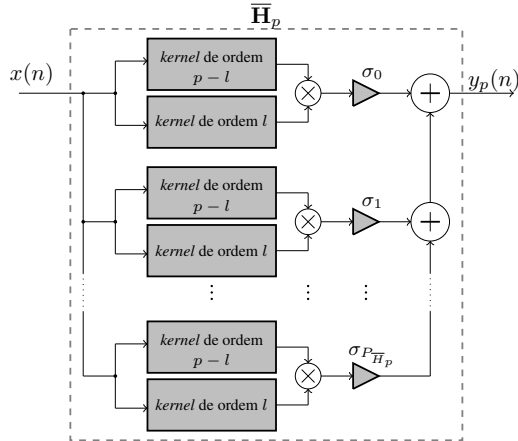


Figura 2.6: Diagrama de blocos de uma implementação com ramos paralelos baseada na SVD de um *kernel* Volterra de ordem p .

Observe que cada ramo dessa estrutura é composto por dois filtros Volterra, um de ordem l e outro de ordem $(p - l)$, além de dois multiplicadores para calcular o produto entre as saídas desses filtros e a multiplicação pelo respectivo valor singular σ_k do ramo.

Uma abordagem alternativa para obter implementações eficientes de filtros Volterra com ordens superiores a dois é tratada em [19]. Em tal abordagem, inicialmente é discutida uma forma para decompor os *kernels* com ordem elevada na forma de *kernels* quadráticos. Em seguida, a decomposição SVD é aplicada à matriz de coeficientes desses *kernels* quadráticos permitindo assim a obtenção de uma estrutura de ramos paralelos cuja significância de cada ramo é definida pelo respectivo valor singular. Como consequência, implementações eficientes de filtros Volterra com ordem elevada são obtidas desprezando os ramos com valor singular de menor magnitude.

2.5 CONCLUSÕES

Neste capítulo, foram apresentadas as principais características do filtro Volterra. Dentre elas, o elevado custo computacional associado às implementações de filtro Volterra destacou-se como uma característica indesejada. Buscando contornar esse problema, estruturas alternativas de implementação de filtros Volterra com número de coeficientes reduzido vêm sendo investigadas. Nesse contexto, as implementações derivadas de transformações matri-

ciais foram abordadas, visto que em alguns casos elas permitem a remoção de diversos coeficientes do filtro Volterra com uma perda em precisão aceitável. Em especial, foram discutidas as estruturas de posto reduzido, as quais derivam da decomposição SVD. Conforme comentado neste capítulo, as estruturas de posto reduzido mostram-se uma opção atrativa devido à sua capacidade de representar certas não linearidades com compromisso adequado entre complexidade e precisão.

3 ESTRATÉGIA PARA IMPLEMENTAR ESTRUTURAS DE POSTO REDUZIDO EFICIENTES EXPLORANDO CARACTERÍSTICAS DO *HARDWARE*

Diferentes estruturas para a implementação de filtros Volterra quadráticos foram discutidas no capítulo anterior. Dentre elas destacam-se as estruturas de posto reduzido, que, por meio da remoção dos ramos menos significativos, permitem a definição de estruturas com compromisso adequado entre capacidade de representação e complexidade computacional. Neste capítulo, é apresentada uma nova estratégia para a implementação de estruturas de posto reduzido, a qual leva a estruturas de implementação mais eficientes em termos de custo computacional. Além disso, também são definidas, ao final do capítulo, algumas das métricas consideradas para analisar a complexidade associada à estratégia de implementação proposta.

3.1 ESTRATÉGIA DE IMPLEMENTAÇÃO PROPOSTA

A estratégia de implementação proposta neste capítulo baseia-se em explorar os diferentes graus de significância dos ramos de uma implementação de posto reduzido, tanto para remoção de ramos não significativos quanto para a escolha das características do *hardware* usado para implementar cada ramo. Nesse contexto, algumas hipóteses iniciais são consideradas em relação às características do *hardware* que será utilizado para implementação do filtro Volterra:

- A plataforma digital alvo para implementação do filtro utiliza representação em ponto fixo.
- O tamanho de palavra disponível nessa plataforma é de n_b bits, sendo $(n_b - 1)$ bits dedicados à mantissa e 1 bit de sinal.
- A faixa dinâmica de representação de valores numéricos é $[-1, 1)$.
- Amostras do sinal fora da faixa dinâmica são truncadas ao valor representável mais próximo (isto é, em caso de *overflow*, ocorre saturação).
- Em função da faixa de representação utilizada, os valores singulares da matriz de coeficientes tem módulo menor ou igual a 1.

É importante comentar que essas hipóteses não levam à perda de generalidade. De maneira geral, um desenvolvimento similar ao da estratégia de implementação proposta pode ser aplicado em plataformas de *hardware* com diferentes características. Como resultado, são definidas estruturas de implementação eficientes do filtro Volterra para as características específicas do *hardware* considerado.

A próxima etapa para o desenvolvimento da estratégia proposta consiste em reorganizar a estrutura da Figura 2.5 com a finalidade de facilitar a exploração dos diferentes graus de significância dos diferentes ramos. Para tal, (33) é reescrita como

$$y_2(n) = \sum_{k=0}^{P_R-1} \text{sign}(\lambda_k) \left[\sqrt{|\lambda_k|} \mathbf{x}^T(n) \mathbf{h}_{2,k} \right]^2 \quad (34)$$

com

$$\text{sign}(\lambda_k) = \begin{cases} 1 & \text{se } \lambda_k \geq 0 \\ -1 & \text{se } \lambda_k < 0 \end{cases}. \quad (35)$$

A partir de (34), nota-se que a estrutura da Figura 2.5 pode ser reorganizada na forma ilustrada na Figura 3.1. Observe que, na entrada do ramo de índice k da estrutura da Figura 3.1, o sinal de entrada $x(n)$ é multiplicado pela raiz quadrada do valor singular λ_k . O sinal resultante, dado por

$$x_k(n) = \sqrt{|\lambda_k|} \cdot x(n), \quad (36)$$

possui potência tipicamente menor do que a potência de $x(n)$ (uma vez que $|\lambda_k| \leq 1$), sendo essa redução de potência mais expressiva nos ramos menos significantes da estrutura (relacionados aos menores valores de λ_k). Como consequência, torna-se possível reduzir a faixa dinâmica utilizada para a implementação de certos ramos sem qualquer impacto na capacidade de representação do sistema. No caso de implementações em FPGAs ou ASICs, tal redução de faixa dinâmica possibilita uma diminuição do tamanho de palavra utilizado, reduzindo assim a complexidade do *hardware*. Nesse contexto, a abordagem aqui proposta consiste em reduzir a faixa dinâmica (e consequentemente o tamanho de palavra) de cada um dos ramos individualmente, usando a probabilidade de *overflow* de $x_k(n)$ como critério para tal. Mais especificamente, a faixa dinâmica é reduzida mantendo a probabilidade de *overflow* de $x_k(n)$ menor ou igual à probabilidade de *overflow* de $x(n)$, considerando este último como um sinal de entrada do tipo ruído gaussiano branco.

Para uma faixa dinâmica de $[-1, 1]$, a probabilidade de *overflow* de um sinal $x(n)$ branco gaussiano com variância σ_x^2 é dada por [20]

$$P[|x(n)| > 1] = \int_{-\infty}^{-1} \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{1}{2\sigma_x^2}x^2(n)} dx + \int_1^{\infty} \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{1}{2\sigma_x^2}x^2(n)} dx. \quad (37)$$

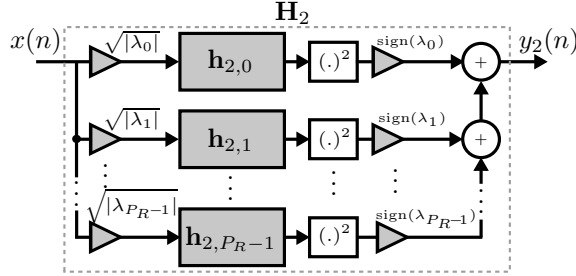


Figura 3.1: Diagrama de blocos de uma forma alternativa de implementação da estrutura da Figura 2.5.

Como a função densidade de probabilidade de $x(n)$ é uma função par, (37) pode ser reescrita como

$$P[|x(n)| > 1] = 2 \cdot \int_{-\infty}^{-1} \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{1}{2\sigma_x^2}x^2(n)} dx. \quad (38)$$

É bem conhecido que (38) representa duas vezes a função de distribuição acumulada (FDA) da variável aleatória $x(n)$ em -1 . Quando a variável aleatória desse tipo de função possui distribuição gaussiana, é comum reescrever sua FDA em função de uma variável $y(n)$ alternativa, onde $y(n)$ possui média zero e variância unitária. Para isso, substitui-se $x(n) = y(n)\sigma_x$ em (38), obtendo

$$P[|x(n)| > 1] = 2 \cdot \int_{-\infty}^{-\frac{1}{\sigma_x}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2(n)} dy \quad (39)$$

como resultado. A FDA representada por (39) é usualmente denotada por $\Phi(y)$, de tal modo que pode-se representar a equação em questão como [20]

$$P[|x(n)| > 1] = 2\Phi\left(-\frac{1}{\sigma_x}\right). \quad (40)$$

Deseja-se agora definir uma faixa dinâmica $[-\alpha_k, \alpha_k]$ para o ramo de índice k de tal forma que a probabilidade de *overflow* de $x_k(n)$ seja menor ou igual à dada em (40), isto é,

$$P[|x_k(n)| > \alpha_k] \leq 2\Phi\left(-\frac{1}{\sigma_x}\right). \quad (41)$$

Fazendo uma análise similar a (37)-(40) para obter $P[|x_k(n)| > \alpha_k]$ e substituindo o valor resultante em (41), obtém-se

$$\Phi\left(-\frac{\alpha_k}{\sqrt{|\lambda_k|\sigma_x}}\right) \leq \Phi\left(-\frac{1}{\sigma_x}\right). \quad (42)$$

Considerando então (42) e ainda que $\Phi(x)$ é uma função crescente, é possível concluir que (41) é satisfeita somente se

$$\alpha_k \geq \sqrt{|\lambda_k|}. \quad (43)$$

Em outras palavras, a faixa dinâmica considerada dentro de um ramo precisa ser de pelo menos $[-\sqrt{|\lambda_k|}, \sqrt{|\lambda_k|}]$ para que a probabilidade de *overflow* de $x_k(n)$ seja menor ou igual do que aquela para $x(n)$ com uma faixa dinâmica de $[-1, 1)$.

A ideia agora é usar o limite estabelecido por (43) para escolha do tamanho de palavra a ser usado na implementação de cada ramo. Nesse contexto, é bem conhecido que, para cada bit removido do tamanho de palavra original, tem-se uma redução pela metade (divisão por 2) nos limites da faixa dinâmica obtida. Como o desejado é reduzir o limite da faixa dinâmica de 1 para α_k visando atender (43) na igualdade, tem-se uma razão de redução de faixa dinâmica dada por $\alpha_k/1$ e, assim, o número de bits a ser removido pode ser obtido resolvendo a seguinte equação:

$$2^{-r_k} = \alpha_k/1 \quad (44)$$

com r_k representando o número de bits a ser removido do ramo de índice k . Considerando (43) e ainda que r_k só pode assumir valores inteiros, a seguinte expressão é obtida para r_k a partir de (44):

$$r_k = \lfloor \log_2(1/\alpha_k) \rfloor \quad (45)$$

com $\lfloor \cdot \rfloor$ representando a operação de truncamento. Finalmente, no contexto da abordagem proposta, o número de bits a ser utilizado para implementação do ramo de índice k da estrutura da Figura 3.1 é dado por

$$n_{b,k} = n_b - r_k. \quad (46)$$

Nota-se, a partir de (45), que r_k é uma função decrescente de α_k . Assim, quanto menor o grau de significância de um dado ramo (isto é, quanto menor o valor de $\sqrt{|\lambda_k|}$ e α_k), maior é o número de bits removido e menor é a quantidade de bits usada na implementação de tal ramo. Além disso, como consequência da operação de truncamento presente em (45), observa-se que diferentes valores de α_k irão resultar em uma mesma redução de número de bits. Mais especificamente, qualquer valor de α_k no intervalo $[2^{-r_k}, 2^{-(r_k-1)})$ irá resultar em um mesmo valor r_k . Por exemplo, para qualquer α_k no intervalo $[0,5; 0,25)$, tem-se $r_k = 1$. Assim, visando simplificar ainda mais o *hardware* usado para implementação da estrutura da Figura 3.1, a proposta neste

trabalho é substituir o ganho $\sqrt{|\lambda_k|}$ de entrada do ramo de índice k por 2^{-r_k} , substituindo também o vetor de coeficientes do filtro FIR por

$$\mathbf{h}'_{2,k} = \frac{\sqrt{|\lambda_k|}}{2^{-r_k}} \mathbf{h}_{2,k}. \quad (47)$$

Como resultado, o multiplicador da entrada de tal ramo pode ser substituído por uma operação (mais simples) de deslocamento, reduzindo o custo computacional de implementação sem alterar as características de transferência do ramo. É importante ressaltar que essa alteração resulta em uma probabilidade de *overflow* de $x_k(n)$ igual a $\Phi(-1/\sigma_x)$, ou seja, igual à probabilidade de *overflow* de $x(n)$. O diagrama de blocos da forma de implementação proposta é ilustrado na Figura 3.2, com $\gg r_k$ representando a operação de deslocamento de r_k bits para a direita com extensão de sinal. O número de bits utilizado para implementação de cada parte da estrutura também é indicado na Figura 3.2.

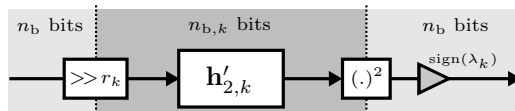


Figura 3.2: Diagrama de blocos de um dos ramos da estrutura proposta.

É importante destacar ainda que, apesar de a estratégia de implementação proposta ser desenvolvida a partir de uma estrutura de posto reduzido de segunda ordem (ilustrada Figura 2.5), ela pode ser facilmente aplicada a *kernels* com ordens superiores. Para tal, a decomposição em *kernels* de segunda ordem apresentada em [19] é utilizada e a estratégia aqui proposta é aplicada a cada um dos *kernels* de segunda ordem resultantes.

3.2 ASPECTOS DE IMPLEMENTAÇÃO E ANÁLISE DE COMPLEXIDADE

O objetivo nesta seção é fazer um detalhamento de alguns aspectos de implementação da estrutura proposta de forma a permitir uma análise mais precisa da complexidade computacional requerida para sua implementação. De maneira geral, são necessárias para tal implementação estruturas que calculem operações de soma, multiplicação e filtragem. Do ponto de vista de implementação digital, essas operações têm como elemento básico comum o somador completo [21]. Nesse contexto, as topologias de síntese escolhidas para a realização de tais operações influenciam diretamente na quantidade de somadores completos considerados na implementação das três estruturas. As topologias utilizadas neste trabalho de pesquisa são em geral baseadas na simplicidade. Contudo, não faz parte do escopo deste trabalho investigar quais

topologias são mais simples e resultam em filtros Volterra com o menor custo de implementação.

3.2.1 Revisão de Topologias Básicas para Síntese Digital de Operações de Soma, Multiplicação e Filtragem

As operações de soma entre dois sinais genéricos de b_{sum} bits, A e B , podem ser realizadas utilizando a topologia de síntese *ripple-carry adder*. Essa topologia, detalhada em nível de somadores completos na Figura 3.3, apresenta um custo total de b_{sum} somadores completos.

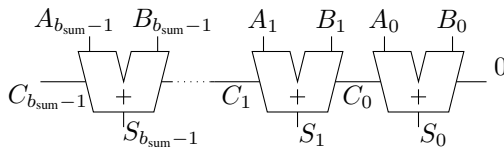


Figura 3.3: Somador sintetizado considerando a topologia *ripple-carry adder*.

Observe que, para representar o resultado da soma, a saída de um somador deve ter precisão de $b_{\text{sum}} + 1$ bits, sendo b_{sum} bits correspondentes à saída S e 1 bit ao *carry out* do último somador completo ($C_{b_{\text{sum}}-1}$) ilustrado na Figura 3.3. Neste trabalho, a saída dos somadores é representada com o mesmo tamanho de palavra de sua entrada, sendo a operação de saturação utilizada para tratar os casos em que o resultado da soma não puder ser representado com b_{sum} bits. Na Figura 3.4 é ilustrada a forma adotada neste trabalho para representar um somador com entradas de b_{sum} bits.

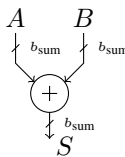


Figura 3.4: Representação simplificada de um somador com entradas de b_{sum} bits.

O cálculo das multiplicações é realizado pelo método da soma dos produtos parciais do multiplicando deslocado à esquerda, discutida em [21]. Tal método é bastante similar à operação tradicional de multiplicação entre dois números na base decimal; contudo, envolve apenas operações binárias. Especificamente, os dois sinais binários usados para o cálculo são definidos como multiplicador e multiplicando, onde este último é representado por A com b_{mnd} bits e o primeiro por B com precisão de b_{mdr} bits. A multiplicação é então efetuada por b_{mdr} estágios de soma do sinal A deslocado sucessivamente à esquerda. Por exemplo, se $b_{\text{mnd}} = 4$ e $b_{\text{mdr}} = 3$, a multiplicação entre A e

B é realizada como segue

$$\begin{array}{r}
 \begin{array}{cccc}
 & A_3 & A_2 & A_1 & A_0 \\
 & \times & B_2 & B_1 & B_0 \\
 \hline
 & A_3B_0 & A_2B_0 & A_1B_0 & A_0B_0 \\
 + & & A_3B_1 & A_2B_1 & A_1B_1 & A_0B_1 \\
 + & A_3B_2 & A_2B_2 & A_1B_2 & A_0B_2 & \\
 \hline
 P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
 \end{array} \\
 \end{array} \quad (48)$$

onde P é o resultado da multiplicação representado por 7 bits, ou seja, $b_{\text{mnd}} + b_{\text{mnr}}$ bits. Pode-se, portanto, implementar (48) em um sistema digital utilizando portas lógicas AND e somadores completos, conforme ilustrado na Figura 3.5. Observe nessa figura que são utilizados oito somadores completos para calcular a multiplicação entre os sinais A e B . Genericamente, o custo total de uma multiplicação com topologia similar à ilustrada na Figura 3.5 é de $b_{\text{mnd}}(b_{\text{mnr}} - 1)$ somadores completos. Assim como na operação de soma, o resultado da multiplicação P deve ser representado com uma quantidade de bits maior do que aqueles de suas entradas para evitar o *overflow*. Neste trabalho é adotado, exceto quando for especificado o contrário, que após cada multiplicação o resultado é truncado para b_{mnd} bits.

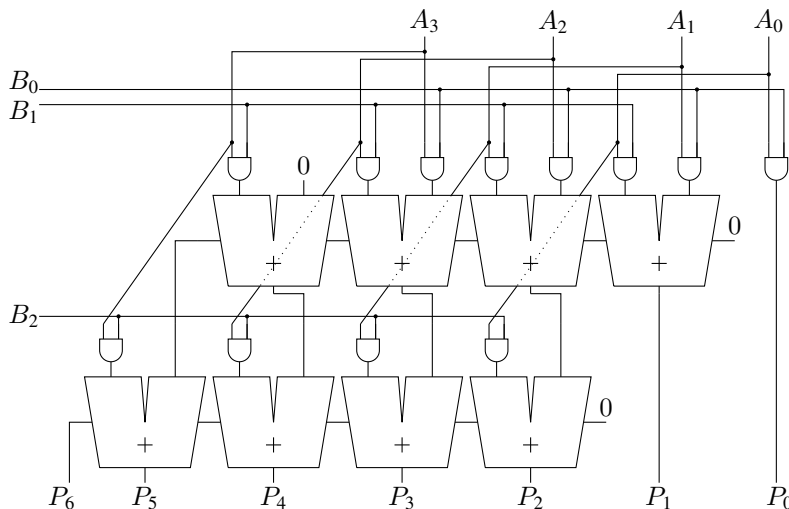


Figura 3.5: Esquema para implementação da multiplicação entre dois sinais A e B representados, respectivamente, por 4 e 3 bits.

O esquema de implementação da multiplicação ilustrada na Figura 3.5 calcula apenas o produto entre sinais positivos. Para considerar também o cálculo de sinais negativos são acrescentadas lógicas combinacionais nas en-

tradas e na saída de cada multiplicador. Essas lógicas combinacionais são projetadas para que, quando necessário, calculem o complemento de dois dos sinais antes e/ou após a multiplicação. O custo total para a síntese dessa lógica é de $2(b_{mnd} + b_{mdr})$ somadores completos.

Um caso específico de multiplicação que é bastante recorrente em implementações de filtros Volterra é a multiplicação por uma constante. Nesse caso, é possível reduzir o número de somadores completos utilizados em função da quantidade de bits iguais a zero da constante, quando esta é representada em base binária. Para isso, a constante é definida como o multiplicador da multiplicação de modo que o número de produtos parciais do multiplicando é fixo e geralmente reduzido. No exemplo anterior, se o bit B_2 for uma constante igual a zero, pode-se reescrever (48) como

$$\begin{array}{rcccccc}
 & & & A_3 & A_2 & A_1 & A_0 \\
 & & & \times & 0 & B_1 & B_0 \\
 \hline
 & & & A_3B_0 & A_2B_0 & A_1B_0 & A_0B_0 \\
 + & & A_3B_1 & A_2B_1 & A_1B_1 & A_0B_1 & \\
 + & 0 & 0 & 0 & 0 & & \\
 \hline
 P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
 \end{array} \tag{49}$$

e é possível desconsiderar no cálculo de (49) 4 dos 8 somadores completos utilizados na implementação ilustrada na Figura 3.5. De modo geral, a quantidade total de somadores utilizados em uma operação por constante é igual a $b_{mnd}(b_{mdr} - k - 1)$, onde k é a quantidade de bits iguais a zero da constante. Para diferenciar a multiplicação genérica da multiplicação por uma constante são adotadas para elas, neste trabalho, simbologias distintas, conforme ilustrado na Figura 3.6. A simbologia usada para representar a multiplicação genérica é ilustrada na Figura 3.6(a), enquanto a multiplicação por constante é representada conforme ilustrado na Figura 3.6(b).

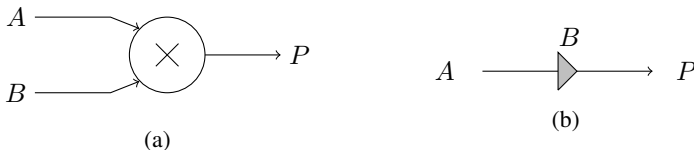


Figura 3.6: Simbologias adotadas para representar as operações de multiplicação. (a) Símbolo do multiplicador genérico. (b) Símbolo do multiplicador por constante.

Finalmente, para realizar a operação de filtragem são utilizados filtros FIR, os quais possuem a seguinte relação de entrada e saída

$$y(n) = \sum_{i=0}^{N_F-1} x(n - i)h_F(i), \tag{50}$$

onde N_F é o número de coeficientes do filtro, $h_F(i)$ seu i -ésimo coeficiente e $x(n)$ é o sinal de entrada. Considera-se neste trabalho, que o sinal de entrada dos filtros FIR são representados com b_s bits e seus coeficientes com b_c bits. Uma possível topologia de síntese em sistema digital que realize (50) é ilustrada na Figura 3.7. Tal topologia é a bem conhecida forma direta [22], que efetua o cálculo de N_F multiplicações por constante e $N_F - 1$ somas. Nesse contexto, o número de somadores completos utilizados com as N_F multiplicações é

$$\sum_{i=0}^{N_F-1} [b_s(b_c - k_{h_F,i} - 1) + 2(b_s + b_c)], \quad (51)$$

onde $k_{h_F,i}$ é o número de bits igual a zero do i -ésimo coeficiente do filtro FIR representado em base binária. Para reduzir o erro de quantização e a probabilidade de *overflow* na operação de filtragem, o resultado das N_F multiplicações é representado com precisão de $b_c + b_s$ bits e apenas a saída $y(n)$ é arredondada para b_s bits. Como consequência, os $N_F - 1$ somadores (ilustrados na Figura 3.7) recebem em suas entradas sinais com precisão de $b_c + b_s$ bits, exigindo um custo computacional de $(N_F - 1)(b_c + b_s)$ somadores completos para sua implementação.

A estrutura de síntese para os filtros FIR ilustrada na Figura 3.7 apresenta elevada complexidade de implementação se comparada com a complexidade dos somadores e multiplicadores. Uma forma alternativa de calcular (50), contornando esse problema, é utilizar para tal apenas uma unidade de multiplicação e acumulação (MAC). O diagrama de blocos dessa forma alternativa de implementação dos filtros FIR é ilustrada na Figura 3.8. Os blocos Vetor de entrada e Vetor de coeficientes são sintetizados na forma de memórias no sistema digital. A saída dessas memórias é controlada por uma lógica combinacional que representa um custo total de $2 \lceil \log_2(N_F) \rceil$ somadores completos. Além dessa lógica combinacional de controle, também são sintetizados um multiplicador genérico com entradas de b_c e b_s bits e um somador cujas entradas têm precisão de $b_s + b_c$ bits. Portanto, o custo

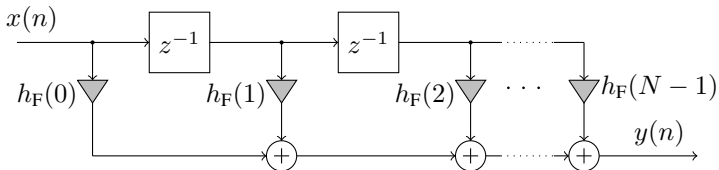


Figura 3.7: Estrutura de um filtro FIR (forma direta), utilizada para realizar a filtragem de um sinal conforme definido por (50).

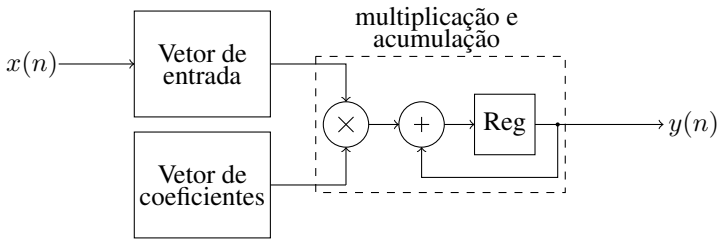


Figura 3.8: Topologia de síntese do filtro FIR, a qual utiliza apenas uma unidade de multiplicação e acumulação.

computacional dessa estrutura em termos de somadores completos é igual a $b_s(b_c - 1) + 2(b_c + b_s) + (b_s + b_c) + 2 \lfloor \log_2(N_F) \rfloor$. As informações referentes à quantidade de somadores completos considerados para a realização das operações de soma, multiplicação e filtragem são apresentadas de forma resumida na Tabela 1.

Tabela 1: Custo computacional, em termos do número de somadores completos utilizados, das operações básicas necessárias para implementação de um filtro Volterra quadrático

	Somadores completos requeridos
Somador	b_{sum}
Multiplicador	$b_{\text{mnd}}(b_{\text{mdr}} - 1) + 2(b_{\text{mnd}} + b_{\text{mdr}})$
Multiplicador (constante)	$b_{\text{mnd}}(b_{\text{mdr}} - k - 1) + 2(b_{\text{mnd}} + b_{\text{mdr}})$
FIR - Forma Direta	$\sum_{i=0}^{N_F-1} [b_s(b_c - k_{h,i} + 2) + 3b_c] - (b_c + b_s)$
FIR - Unidade MAC	$b_c b_s + 3b_c + 2b_s + 2 \lfloor \log_2(N_F) \rfloor$

3.2.2 Análise de Complexidade para a Estratégia de Implementação Proposta

O objetivo principal desta seção é desenvolver expressões que permitam comparar a complexidade de implementação da estrutura proposta (veja Figura 3.2) com as das estruturas Volterra triangular (veja Figura 2.3) e Volterra de posto reduzido convencional (veja Figura 2.5). Essa comparação será realizada em termos de somadores completos necessários para implementação das diferentes estruturas; portanto, baseia-se nas informações da Tabela 1. Além disso, são consideradas, para o desenvolvimento das expressões, as mesmas hipóteses para as características do *hardware* alvo listadas no início deste capítulo. Assim, o tamanho de palavra considerado na implementação das estruturas é igual a n_b .

As implementações do *kernel* triangular, que são baseadas no diagrama de blocos da Figura 2.3, são formadas por uma estrutura com N ramos em paralelo, sendo que cada ramo efetua o cálculo de uma operação de multiplicação e uma de filtragem. Além dessas duas operações, as implementações do *kernel* triangular necessitam de $N - 1$ somadores para calcular a soma da saída de todos ramos. Desse modo e considerando os filtros FIR implementados na forma direta, o número total de somadores completos requeridos para as implementações do *kernel* triangular é

$$C_T = \sum_{j=0}^{N-1} \left\{ n_b(n_b + 2) + \left[\sum_{i=0}^{N-1-j} n_b(n_b - k_{h_j,i} + 5) \right] \right\} - n_b, \quad (52)$$

onde $k_{h_j,i}$ corresponde à quantidade de bits iguais a zero do i -ésimo elemento do vetor de coeficientes do j -ésimo FIR ($\hat{\mathbf{h}}_{j,2}$). O custo dessa mesma estrutura ilustrada na Figura 2.3 quando considerada uma topologia com apenas um multiplicador e acumulador para a implementação do FIR é de

$$C_T = N [2n_b^2 + 9n_b + 2 \lceil \log_2(N) \rceil] - n_b \quad (53)$$

somadores completos.

Conforme discutido anteriormente e ilustrado na Figura 2.5, a estrutura de posto reduzido convencional é composta por P_R ramos em paralelo. Cada um desses ramos realiza uma operação de filtragem e duas multiplicações: uma genérica e outra multiplicação por constante. Além disso, a saída da estrutura $y(n)$ é obtida através de $N - 1$ somas nas saídas de todos os seus ramos. Em todas essas operações, os coeficientes do filtro FIR e a constante que representa o valor singular dos ramos são representadas com n_b bits. Portanto, são utilizados

$$C_{P_R} = \sum_{j=0}^{P_H-1} \left\{ \left[\sum_{i=0}^{N-1} n_b(n_b - k_{h_j,i} + 5) \right] + n_b(2n_b + 5 - k_{\lambda_j}) \right\} - n_b \quad (54)$$

somadores completos se a estrutura de posto reduzido convencional tiver seus filtros sintetizados na forma direta e

$$C_{P_R} = \sum_{j=0}^{P_H-1} [3n_b^2 + 12n_b + 2 \lceil \log_2(N) \rceil - k_{\lambda_j} n_b] - n_b \quad (55)$$

no caso de os filtros FIR serem sintetizados com apenas uma unidade de multiplicação e acumulação. Em (54) e (55), a variável k_{λ_j} representa o número

de bits iguais a zero do valor singular do j -ésimo ramo da estrutura de posto reduzido.

Finalmente, a estrutura de posto reduzido proposta, a qual é formada com P_R ramos paralelos dispostos conforme ilustrado pela Figura 3.2, envolve a implementação de uma operação de filtragem e uma multiplicação genérica por ramo. Além disso, também são sintetizados $N - 1$ somadores para o cálculo da saída $y(n)$. Os coeficientes dos filtros FIR são todos representados com precisão de n_b bits. Como consequência da redução da faixa dinâmica obtida a partir da estratégia de implementação proposta, o número de bits dos sinais de entrada de todos os blocos do j -ésimo ramo da estrutura é dado por (46), isto é, $n_{b,j}$ bits. Nesse contexto, a quantidade de somadores completos requeridos para implementação da estrutura obtida com a abordagem proposta e considerando a forma direta para a síntese do filtro é dada por

$$C_P = \sum_{j=0}^{P_H-1} \left\{ \left[\sum_{i=0}^{N-1} n_{b,j}(n_b - k_{h_j,i} + 2) + 3n_b \right] + n_{b,j}n_b + 2n_b \right\} - n_b. \quad (56)$$

No caso em que o filtro é sintetizado com apenas uma unidade de multiplicação e acumulação, a quantidade de somadores completos requeridos para a implementação da estrutura obtida com a abordagem proposta é

$$C_P = \sum_{j=0}^{P_H-1} [2n_{b,j}n_b + 3n_{b,j} + 6n_b + 2 \lfloor \log_2(N) \rfloor] - n_b. \quad (57)$$

3.3 CONCLUSÕES

Neste capítulo, foi apresentada a estratégia de implementação proposta. Durante o desenvolvimento dessa estratégia buscou-se reduzir o número de bits usado na implementação de um dado ramo da estrutura de posto reduzido, explorando, para isso, a importância (significância) do ramo correspondente. Na sequência, foram apresentados os aspectos de implementação considerados para avaliar a complexidade associada à estrutura proposta. Mais especificamente, foram definidas topologias básicas para síntese digital de soma, multiplicação e filtragem consideradas neste trabalho. O custo dessas topologias de síntese, em termos de somadores completos, foi então considerado para se definir expressões analíticas envolvendo a complexidade das estruturas triangular, de posto reduzido convencional e de posto reduzido proposta.

4 CRITÉRIOS PARA REMOÇÃO DE RAMOS EM ESTRUTURAS DE POSTO REDUZIDO

Um passo importante na implementação dos filtros Volterra utilizando estruturas de posto reduzido é a escolha dos ramos que devem ser desprezados. Tal escolha é baseada no grau de significância do ramo e objetiva atender determinadas métricas de desempenho e de complexidade computacional. Os diferentes trabalhos de pesquisa que abordam ou implementam estruturas de posto reduzido não descrevem critérios que possam ser usados para a remoção dos ramos. De modo geral, alguns dos ramos são considerados com significância desprezível arbitrariamente e a avaliação de desempenho e de custo computacional é realizada por meio de comparações entre os resultados obtidos com a estrutura ideal com os obtidos considerando a estrutura de posto reduzido. Nesse contexto, é relevante definir critérios que permitam identificar de forma objetiva os ramos da estrutura que não contribuem significativamente para a saída do sistema e que, portanto, podem ser desprezados. Neste capítulo, são propostas duas abordagens para a definição de critérios de remoção dos ramos, sendo elas a abordagem numérica e a abordagem energética.

4.1 ABORDAGEM NUMÉRICA

A abordagem numérica é baseada na determinação da estrutura de posto reduzido com a melhor capacidade de representação possível considerando as características do *hardware* alvo. Nesse sentido, pretende-se identificar e permitir a remoção de apenas os ramos cuja contribuição para a saída do sistema for nula. A identificação de tais ramos é baseada na magnitude do valor singular de cada ramo e no tamanho de palavra presente no sistema digital alvo. Note que, de acordo com as hipóteses assumidas no Capítulo 3, sinais no sistema digital alvo com magnitude inferior a 2^{-n_b} são arredondados para zero. Assim, os ramos com saída inferior a tal valor podem ser removidos sem afetar a capacidade de representação do sistema. Para a estrutura de posto reduzido ilustrada na Figura 2.5, a condição que determina os ramos com saída nula é

$$|\lambda_k| \leq 2^{-n_b}. \quad (58)$$

Quando (58) for satisfeita, o valor armazenado no sistema digital para representar o valor singular do k -ésimo ramo será zero e a saída de tal ramo será sempre nula. Agora, para a estrutura de posto reduzido proposta, a qual possui ramos dispostos conforme ilustrado na Figura 3.2, utiliza-se o sinal de saída de cada ramo [veja (34)] para a definição do critério de remoção. Mais especificamente, é razoável considerar que o k -ésimo ramo terá contribuição nula para saída da estrutura se

$$\left[\sqrt{|\lambda_k|} \mathbf{x}^T(n) \mathbf{h}_{2,k} \right]^2 < 2^{-n_b} \quad (59)$$

for satisfeita independentemente das amostras do vetor \mathbf{x} . Sabendo que os sinais no *hardware* alvo são limitados ao intervalo $[-1, 1]$, pode-se afirmar que (59) é satisfeita quando

$$\left[\sqrt{|\lambda_k|} \sum_{i=0}^{N-1} h_{2,k}(i) \right]^2 < 2^{-n_b} \quad (60)$$

for válido, onde $h_{2,k}(i)$ denota o i -ésimo coeficiente do filtro FIR implementado no ramo k da estrutura de posto reduzido proposta. Assim, a condição definida em (60) é utilizada como critério para remoção dos ramos da estrutura de posto reduzido proposta.

Um aspecto interessante do critério para remoção dos ramos estabelecida pela abordagem numérica proposta é que, a partir de (58) e (60), pode-se determinar a estrutura de posto reduzido com a melhor capacidade de representação possível para as características do *hardware* alvo. Contudo, o desenvolvimento dessa abordagem desconsidera a contribuição relativa de cada ramo para a saída do filtro Volterra e o critério de remoção definido é fortemente dependente do tamanho da palavra disponível no sistema digital alvo. Assim, os critérios de remoção estabelecidos pela abordagem numérica geralmente permitem a implementação de ramos com grau de significância desprezível, resultando em um aumento desnecessário na complexidade computacional do filtro Volterra. Por exemplo, considere a implementação de um *kernel* quadrático em um sistema digital com n_b igual a 16 bits. Considere também que o *kernel* em questão possa ser representado com dois ramos da estrutura ilustrada na Figura 2.5, onde a magnitude dos valores singulares desses ramos são $1 - 2^{-15}$ e 2^{-15} . Nesse caso, o critério definido por (58) permite a implementação de ambos os ramos da estrutura, mesmo que o valor singular do ramo de maior significância seja aproximadamente 2^{15} vezes maior do que o valor singular do ramo de menor significância. Assim, observa-se que, em certos casos, a abordagem numérica será ineficiente em manter o compromisso entre capacidade de representação e o custo computacional ao definir a estrutura de implementação do filtro Volterra. Com o objetivo contornar esse problema e permitir a implementação de apenas ramos com contribuição significativa para a saída do sistema, uma abordagem baseada na energia dos sinais de saída de cada ramo é apresentada a seguir.

4.2 ABORDAGEM ENERGÉTICA

Outra abordagem aqui discutida para definir critérios de remoção dos ramos da estrutura de posto reduzido é a abordagem energética. Tal abordagem é baseada no uso de uma relação entre a energia do sinal na saída dos ramos da estrutura de posto reduzido e a energia na saída do ramo de maior significância como métrica para identificar os ramos que devem ser removidos. A

análise energética discutida é focada na estrutura de posto reduzido formada com ramos dispostos como ilustrado na Figura 3.2; contudo, ela é facilmente estendida para as demais estruturas consideradas. As mesmas hipóteses sobre as características do sinal de entrada assumidas no Capítulo 3 são consideradas no desenvolvimento da abordagem energética. Nesse sentido, cada ramo da estrutura recebe como entrada um sinal do tipo ruído branco gaussiano de variância σ_x^2 e média igual a zero. Note na Figura 3.2 que, ao receber o sinal de entrada, a primeira operação realizada pelo k -ésimo ramo da estrutura de posto reduzido é um deslocamento à direita de r_k vezes. Uma vez que cada deslocamento à direita corresponde a divisão do sinal de entrada por dois, a energia do sinal de interesse deslocado r_k vezes é dada por [20]

$$\sigma_{x_{1,k}}^2 = \left(\frac{1}{2^{r_k}} \right)^2 \sigma_x^2. \quad (61)$$

Após a operação de deslocamento, o sinal resultante é enviado para o filtro FIR de coeficientes $\mathbf{h}'_{2,k}$. Considerando um sinal gaussiano como entrada do filtro, pode-se definir a variância na saída do k -ésimo filtro FIR por [20]

$$\sigma_{x_{2,k}}^2 = \sigma_{x_{1,k}}^2 \sum_{i=0}^{N-1} h_{2,k}'^2(i), \quad (62)$$

onde $h(i)'_{2,k}$ é o i -ésimo elemento do vetor de coeficientes $\mathbf{h}'_{2,k}$. Substituindo (47) em (62), tem-se

$$\sigma_{x_{2,k}}^2 = \sigma_{x_{1,k}}^2 \sum_{i=0}^{N-1} \left(\frac{\sqrt{|\lambda_k|}}{2^{-r_k}} h_{2,k}(i) \right)^2, \quad (63)$$

onde $h_{2,k}(i)$ é o i -ésimo componente de um vetor de tamanho N com norma unitária. Dessa forma, utilizando (61) para definir o valor de $\sigma_{x_{1,k}}$ em (63), pode-se expressar a energia do sinal de interesse na saída do filtro FIR do k -ésimo ramo como

$$\sigma_{x_{2,k}}^2 = |\lambda_k| \sigma_x^2. \quad (64)$$

Finalmente, para definir a energia na saída de cada ramo avalia-se a variância da função densidade de probabilidade (PDF – *probability density function*) que representa os sinais após a operação quadrática. Nessa avaliação, a PDF na saída de cada ramo é obtida por uma transformação de variáveis aleatórias, na qual a função quadrática é aplicada em um sinal gaussiano de média zero e variância $\sigma_{x_{2,k}}^2$. A PDF resultante dessa transformação é [20]

$$p_{x_{3,k}}(x) = \begin{cases} \frac{1}{\sqrt{2\pi x \sigma_{x_{2,k}}^2}} \exp\left(-\frac{x}{2\sigma_{x_{2,k}}^2}\right) & \text{se } x \geq 0 \\ 0 & \text{se } x < 0. \end{cases} \quad (65)$$

A variância e a média da PDF representada por (65) podem ser obtidas por meio do operador valor esperado [20]. Especificamente, tal operador aplicado diretamente em (65) resulta na média da função, a qual é definida em termos da variância do sinal de entrada como $\sigma_{x_{2,k}}^2$. Já quando o operador valor esperado é aplicado ao quadrado de $p_{x_{3,k}}(x)$ e subtraída sua média $\sigma_{x_{2,k}}^2$, tem-se como resultado a variância da PDF correspondente, definida por $2(\sigma_{x_{2,k}}^2)^2$. Portanto, a energia na saída dos ramos da estrutura de posto reduzido é definida como

$$\sigma_{x_{3,k}}^2 = 2(|\lambda_k| \sigma_x^2)^2. \quad (66)$$

A partir de (66) pode-se estabelecer um critério de remoção dos ramos consistente com o grau de significância correspondente. O critério aqui estabelecido considera que todo o ramo com energia ζ vezes inferior à energia do ramo de maior significância pode ser desconsiderado. Assim, os ramos que satisfizerem

$$\sigma_{x_{3,k}}^2 \zeta < \sigma_{x_{3,max}}^2 \quad (67)$$

são desconsiderados na saída do sistema, onde $\sigma_{x_{3,max}}^2$ é a energia do ramo de maior significância. Substituindo (66) em (67) pode-se obter como critério para remoção do k -ésimo ramo a seguinte relação

$$\frac{\lambda_{max}^2}{\lambda_k^2} > \zeta, \quad (68)$$

onde λ_{max}^2 é o valor singular da estrutura de posto reduzido de maior magnitude. Desse modo, o critério para remoção dos ramos estabelecido pela abordagem energética considera apenas informações referentes à significância do ramo para definir a estrutura de posto reduzido final.

É relevante comentar que o valor de ζ deve ser definido considerando os requisitos de desempenho desejados para a aplicação específica em que o filtro Volterra é utilizado. Por exemplo, em aplicações envolvendo a identificação de informações pelo sistema nervoso auditivo central do ser humano, estudos mostram que, para manter a inteligibilidade da informação, a energia do sinal de interesse deve ser pelo menos 10 vezes maior do que a energia do ruído [23]. Nesse sentido, é coerente afirmar que, para essas aplicações, ζ deve ser definido com valores iguais ou superiores a 10.

4.3 CONCLUSÕES

Neste capítulo, foram definidas duas abordagens para definição dos ramos removidos de uma estrutura de posto reduzido. A primeira apresentada é a abordagem numérica, que permite definir estruturas de posto reduzido com a

melhor capacidade de representação de um dado filtro Volterra considerando as características do *hardware* alvo. A segunda abordagem é a energética. O critério definido por tal abordagem leva em conta uma análise da razão entre a energia na saída do ramo de maior significância sobre os demais ramos da estrutura de posto reduzido.

5 RESULTADOS DE IMPLEMENTAÇÃO

O objetivo central neste capítulo é avaliar a complexidade associada à estratégia de implementação de filtros Volterra proposta, como também a sua capacidade em representar precisamente *kernels* não lineares. Para isso, realiza-se uma análise comparativa entre filtros Volterra implementados com três estruturas distintas, sendo elas: i) a triangular (veja Figura 2.3); ii) a de posto reduzido convencional, definida a partir do diagrama de blocos da Figura 2.5; e iii) a de posto reduzido proposta, que possui ramos dispostos conforme ilustrado na Figura 3.2. É importante mencionar que são assumidos cenários variados para a implementação dessas estruturas. Uma dessas variações é a forma de síntese dos filtros FIR. As diferentes estruturas de filtro Volterra implementadas consideram tanto os filtros FIR na forma direta (veja Figura 3.7), quanto os que utilizam uma unidade de multiplicação e acumulação (veja Figura 3.8). Considera-se também que as implementações do filtro Volterra são efetuadas em *hardwares* com tamanho de palavra de 8, 12 e 16 bits. Além disso, são consideradas implementações das estruturas de posto reduzido com ramos removidos tanto pelos critérios da abordagem numérica quanto pelos da abordagem energética. Para os critérios definidos por esta última abordagem, o valor de ζ [veja (68)] é escolhido considerando um critério conservador para aplicações envolvendo a identificação de informações pelo sistema nervoso auditivo central. Mais especificamente, ζ é definido igual a 100, de modo que tal critério permitirá a definição de estruturas de posto reduzido com capacidade de representação adequada para as aplicações consideradas.

A complexidade das diferentes estruturas de implementação consideradas são avaliadas por meio de duas métricas. Essas métricas têm como objetivo quantificar o custo computacional relativo às estruturas triangular, de posto reduzido convencional e de posto reduzido proposta, quando utilizadas para implementação de alguns exemplos de filtros Volterra. A primeira métrica envolve a quantidade de somadores completos necessários em implementações de filtros Volterra. Especificamente, o número de somadores completos exigido nessas implementações é calculado utilizando (52)-(57) e, então, a complexidade das estruturas é avaliada. A segunda métrica adotada para a avaliação da complexidade corresponde à área ocupada de um FPGA em termos de número de blocos lógicos básicos utilizados. O FPGA alvo das implementações foi o 5CGXFC7C7F23C8 da família Cyclone V da Altera Corporation e a plataforma de projeto utilizada foi o Software Quartus II.

A capacidade de representação correspondente às estruturas de implementação consideradas é avaliada por meio do erro quadrático médio norma-

lizado (NMSE - *Normalized Mean Squared Error*), o qual é definido como

$$e_{NMSE} = \frac{\sum_{i=0}^{n-1} [y(i) - y'(i)]^2}{\sum_{i=0}^{n-1} y(i)^2}. \quad (69)$$

Em (69), n representa a quantidade de amostras utilizada na avaliação do NMSE, y denota a saída idealizada do filtro Volterra e y' , em contraste a y , corresponde à saída do filtro Volterra implementado com precisão finita de n_b bits e por meio de uma das estruturas de implementação consideradas.

Nas próximas seções, são apresentados três exemplos de filtros Volterra implementados neste trabalho, bem como os resultados obtidos em complexidade e capacidade de representação com essas implementações. Para facilitar a compreensão dos resultados, o restante do capítulo é organizado de acordo com o seguinte padrão. As discussões envolvendo as implementações dos três filtros Volterra utilizados (como exemplos) são realizadas em diferentes seções. Cada uma dessas seções é subdividida em mais quatro seções. Na primeira, as estruturas de implementação do filtro Volterra são descritas. A segunda contém os resultados de custo computacional considerando todos os casos em que o filtro Volterra é implementado. Tais resultados são analisados na terceira seção. Finalmente, a quarta seção traz uma discussão acerca da capacidade em representar precisamente o filtro Volterra do exemplo utilizando as estruturas triangular, de posto reduzido convencional e de posto reduzido proposta.

5.1 EXEMPLO 1

5.1.1 Descrição das Estruturas de Implementação

O primeiro exemplo de filtro Volterra considerado neste trabalho consiste na implementação do filtro digital quadrático definido em [9] e escalado por um termo constante K . O filtro em questão possui tamanho de memória igual a 21 e 441 coeficientes, sendo estes últimos obtidos a partir de

$$h_2(i, j) = K \frac{1, 5}{2\pi[1, 5^2 + (i - 10)^2 + (j - 10)^2]^2} \quad (70)$$

com $i, j = 0, 1, \dots, 20$ e K igual ao inverso da norma de Frobenious da matriz de coeficientes do filtro. O fator de escala K é utilizado com o objetivo de reduzir o erro na saída das estruturas implementadas, o qual é oriundo da quantização dos coeficientes do filtro. Para $K = 1$, muitos dos coeficientes definidos em (70) tornam-se zero quando quantizados e o erro dessa quantização é bastante significativa na saída das estruturas implementadas, principalmente nos casos em que o *hardware* alvo possui precisão de 8 ou 12 bits.

Tabela 2: Número de bits removidos dos ramos da estrutura de posto reduzido proposta para o caso do Exemplo 1

Ramos em ordem crescente de significância	r_k
1	0
2	1
3	2
4	3
5	4
6	6

O filtro Volterra deste exemplo é implementado com as estruturas triangular, de posto reduzido convencional e de posto reduzido proposta. No caso da implementação triangular, a saída do filtro Volterra é calculada utilizando a implementação baseada em coordenadas diagonais. A estrutura obtida com tal implementação é similar à ilustrada na Figura 2.3, sendo, especificamente, composta por 21 ramos em paralelo, onde o i -ésimo ramo possui um filtro FIR com os coeficientes definidos por (20). Com respeito às estruturas de posto reduzido convencional e proposta, os ramos removidos são determinados utilizando os critérios de remoção derivados das abordagens numérica e energética discutidas no Capítulo 4. Para este exemplo, os critérios definidos pela abordagem numérica permitem a implementação dos três ramos mais significativos da estrutura de posto reduzido em um *hardware* com tamanho de palavra igual a 8 bits ($n_b = 8$). Nos casos em que $n_b = 12$ e $n_b = 16$, o critério definido permite a implementação de cinco e seis dos ramos mais significativos da estrutura, respectivamente. Finalmente, se utilizado o critério de remoção definido pela abordagem energética com $\zeta^{-1} = 1\%$, são implementados apenas os dois ramos mais significativos das estruturas de posto reduzido. Considerando a estrutura de posto reduzido proposta, em que a quantidade de bits utilizada na implementação de cada ramo varia de acordo com a magnitude do valor singular correspondente [veja (43)-(46)], a redução do número de bits obtida para os seis ramos mais significativos da estrutura é dada na Tabela 2.

5.1.2 Resultados de Complexidade para as Implementações Consideradas

A complexidade das implementações do Exemplo 1 em termos do número de somadores completos utilizados é detalhada na Tabela 3 e ilustrada graficamente nas Figuras 5.1 e 5.2. A primeira dessas figuras contém os resultados de complexidade das estruturas triangular, de posto reduzido convencional e de posto reduzido proposta, considerando que os filtros FIR são

Tabela 3: Quantidade de somadores completos utilizados na implementação do *kernel* do Exemplo 1 para as diferentes estruturas consideradas

Estrutura de implementação	Topologia do filtro FIR	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	Forma Direta Unidade MAC	X	5624	14508	27952
		X	3730	7506	12626
Posto Reduzido Convencional	Forma Direta	Númérica	3768	11400	22784
		Energética	2672	4932	8352
	Unidade MAC	Númérica	694	2222	4332
		Energética	476	932	1540
Posto Reduzido Proposto	Forma Direta	Númérica	3440	9917	18178
		Energética	2549	4854	7962
	Unidade MAC	Númérica	471	1512	3038
		Energética	330	678	1154

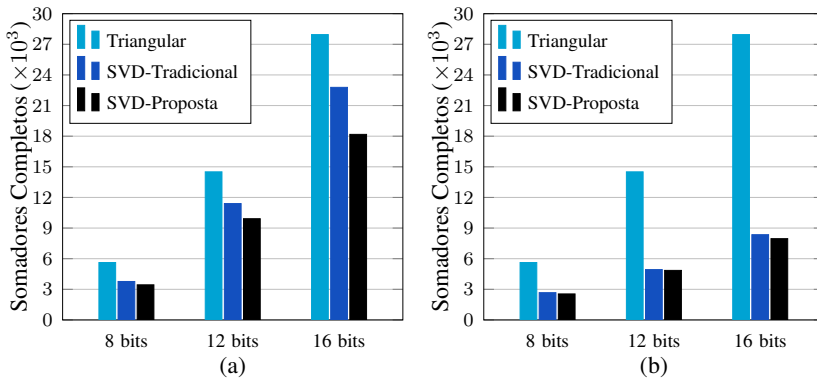


Figura 5.1: Complexidade das implementações do *kernel* do Exemplo 1 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

implementados na forma direta. Além disso, na Figura 5.1(a) considera-se que as estruturas de posto reduzido tiveram seus ramos removidos pelos critérios da abordagem numérica. Em contraste, os resultados da Figura 5.1(b) foram obtidos das estruturas de posto reduzido cujos ramos implementados são definidos pela abordagem energética. As implementações que consideram o uso de uma unidade de multiplicação e acumulação para síntese dos filtros FIR têm seus resultados apresentados na Figura 5.2. Mais especificamente, nas Figuras 5.2(a) e 5.2(b) considera-se que os ramos removidos das estruturas de posto reduzido são definidos, respectivamente, pelas abordagens numérica e energética.

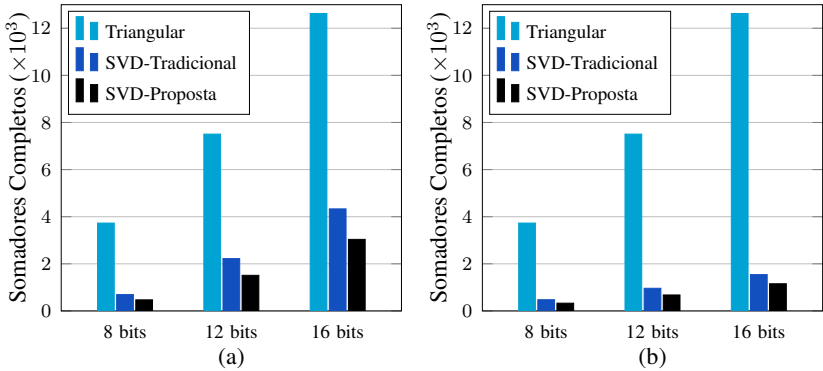


Figura 5.2: Complexidade das implementações do *kernel* do Exemplo 1 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

Os resultados de complexidade considerando o número de blocos lógicos básicos utilizado nas implementações em FPGA da estrutura proposta são mostrados na Tabela 4 e nas Figuras 5.3 e 5.4. O esquema de apresentação dos resultados em tais figuras é semelhante aquele adotado para os casos em que a complexidade é avaliada em termos do número de somadores completos. Isto é, os resultados das implementações com filtro FIR na forma direta e com a unidade de multiplicação e acumulação são apresentados, respectivamente, nas Figuras 5.3 e 5.4. A complexidade das estruturas de posto reduzido cujos ramos removidos são definidos pela abordagem numérica é apresentada nas Figuras 5.3(a) e 5.4(a) e os casos em que a abordagem energética é considerada para tal têm seus resultados em complexidade ilustrados nas Figuras 5.3(b) e 5.4(b).

5.1.3 Análise da Complexidade das Implementações

A partir dos resultados apresentados na Tabela 3, observa-se que as estruturas de posto reduzido têm complexidade, em termos de somadores completos utilizados, menor do que a das implementações baseadas em coordenadas diagonais, se forem comparadas a cenários em implementação semelhantes (mesma topologia do filtro FIR e valores iguais de n_b). A proporção da redução de complexidade depende do cenário no qual as estruturas são implementadas. Comparando as Figuras 5.1(a) e 5.1(b) [ou 5.2(a) e 5.2(b)], por exemplo, é evidente que a relação da complexidade entre as estruturas de posto reduzido e a triangular é inferior quando a abordagem numérica é con-

Tabela 4: Ocupação de blocos lógicos em FPGA com a implementação do *kernel* do Exemplo 1 para as diferentes estruturas consideradas

Estrutura de implementação	Topologia do filtro FIR	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	Forma Direta	X	6496	15149	24948
	Unidade MAC	X	3929	7613	12864
Posto Reduzido Convencional	Forma Direta	Númerica	2618	8914	16104
		Energética	1677	3060	5068
	Unidade MAC	Númerica	595	1870	3878
		Energética	408	715	1263
Posto Reduzido Proposto	Forma Direta	Númerica	2358	7618	14302
		Energética	1564	2390	4267
	Unidade MAC	Númerica	464	1373	2717
		Energética	329	539	953

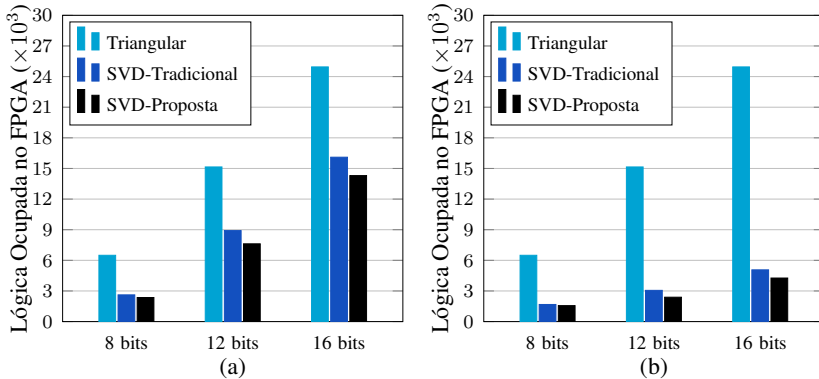


Figura 5.3: Complexidade das implementações do *kernel* do Exemplo 1 implementado em FPGA e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

siderada para remoção dos ramos da estrutura. Tal fato ocorre pois para o filtro Volterra com coeficientes definidos por (70), o número de ramos removidos da estrutura de posto reduzido é superior quando utilizada a abordagem energética com $\zeta^{-1} = 1\%$. O tamanho de palavra disponível no *hardware* alvo também é um fator bastante impactante na complexidade de implementação das estruturas consideradas neste trabalho. Além de ser diretamente relacionado com a quantidade de somadores completos utilizados por operação, o valor de n_b também influencia na definição dos ramos implementados da estrutura de posto reduzido se considerada a abordagem numérica para tal. Nesse contexto, quanto maior o tamanho da palavra, mais ramos da estrutura

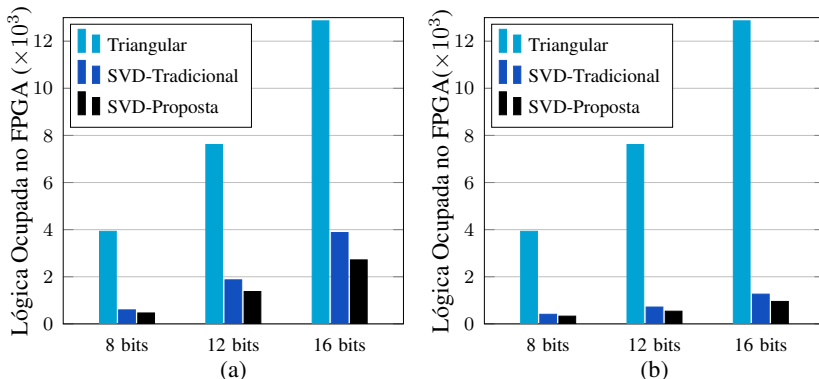


Figura 5.4: Complexidade das implementações do *kernel* do Exemplo 1 implementado em FPGA e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

de posto reduzido são implementados. Como consequência, o ganho proporcional em complexidade de implementação obtido ao se utilizar as estruturas de posto reduzido é menor, conforme pode ser observado nas Figuras 5.1(a) e 5.2(a). As implementações baseadas em coordenadas diagonais também têm sua complexidade fortemente influenciada pelo valor de n_b . De modo geral, mesmo com o escalamento de todos coeficientes por K , os elementos das extremidades da matriz têm magnitude insuficiente para serem representados no *hardware* (principalmente os que possuem precisão de 8 bits). Assim, os filtros FIR cuja implementação é baseada nos elementos presentes na matriz triangular possuem diversos de seus coeficientes com amplitude insignificante, sendo que em muitos casos esses coeficientes são quantizados para zero. Nesse contexto, muitas das multiplicações envolvidas na filtragem podem ser evitadas e, quando necessárias, podem ser implementadas com custo bastante reduzido em termos de somadores completos se considerado o multiplicador por constante para tal (o custo da multiplicação por uma constante é mostrado na Tabela 1). Observa-se, portanto, na Figura 5.1(b) que o ganho proporcional em complexidade de implementação ao se utilizar as estruturas de posto reduzido é superior para valores maiores de n_b . Tal fato é observado porque para n_b pequeno, muitas das operações de filtragem realizadas na estrutura triangular podem ser desprezadas, reduzindo, assim, o custo computacional dessa estrutura. Nos casos em que a filtragem é efetuada com apenas uma unidade de multiplicação e acumulação, mesmo que poucos coeficientes do filtro tenham magnitude significativa para serem representa-

dos no *hardware* alvo, toda a unidade de multiplicação e acumulação deve ser implementada. Assim, a variação de n_b não apresenta grandes impactos se consideradas comparações com as estruturas de posto reduzido para um número fixo de ramos [veja a Figura 5.2(b)].

Com respeito às duas estruturas de posto reduzido consideradas, é possível concluir, a partir da Tabela 3 e das Figuras 5.1 e 5.2, que a estratégia de implementação proposta resulta em estruturas com menor complexidade. O ganho em complexidade obtido com a abordagem proposta é dependente do cenário no qual as duas estruturas de posto reduzido são comparadas. De maneira geral, quanto mais bits são reduzidos por ramo com a estratégia proposta, maior é o ganho em custo computacional obtido. Por meio de uma análise comparativa entre as Figuras 5.1(a) e 5.1(b) observa-se que a estratégia de implementação proposta resulta em reduções de complexidade proporcionalmente inferiores, em comparação com a de posto reduzido convencional, quando utilizada a abordagem energética na definição dos ramos removidos. Tal fato ocorre pois a abordagem energética permite que apenas dois dos ramos mais significativos da estrutura sejam implementados. Desses dois ramos, apenas o que possui menor valor singular pode ser representado com número de bits reduzido, sendo essa redução igual a 1 bit (veja Tabela 2). No caso dos resultados da Figura 5.1(a), que consideram a abordagem numérica para definir os ramos removidos da estrutura, são implementados 3, 5 e 6 dos ramos mais significativos para *hardware* com precisão de, respectivamente, 8, 12 e 16 bits. Dessa forma e considerando as informações da Tabela 2, é possível concluir que um número maior de bits por ramo é reduzido quando considerada a abordagem numérica. Como consequência, tal abordagem proporciona ganhos superiores de complexidade para a estrutura de posto reduzido proposta em relação à estrutura de posto reduzido convencional. Ainda considerando a complexidade das estruturas de posto reduzido, conclusões similares as extraídas das Figuras 5.1(a) e 5.1(b) são obtidas ao confrontar os resultados apresentados nas Figuras 5.2(a) e 5.2(b). Contudo, comparando as Figuras 5.1 e 5.2, as quais são obtidas a partir de estruturas com diferentes tipos de filtros FIR, verifica-se uma redução de ganho (proporcional) em complexidade obtido com a estrutura proposta, em relação à estrutura de posto reduzido convencional, quando essas são implementadas com filtros FIR na forma direta. É importante lembrar que, além da redução de bits por ramo, as estruturas derivadas da estratégia de implementação proposta contém uma multiplicação por ramo a menos do que as estruturas de posto reduzido convencionais. A redução em complexidade obtida com a retirada dessa multiplicação é proporcionalmente superior nos cenários em que os filtros FIR possuem menor complexidade. Nesse sentido, o ganho de complexidade obtido com as estruturas propostas é superior (em proporção)

quando os filtros FIR são implementados com a unidade de multiplicação e acumulação (pois este tipo de filtro possui menor complexidade), conforme pode ser observado comparando as Figuras 5.1 e 5.2.

Os resultados de complexidade das implementações em FPGA carregam essencialmente as mesmas informações que os resultados obtidos quando considerado o número de somadores completos. Nesse sentido, as discussões anteriores envolvendo a complexidade em termos do número de somadores completos, podem ser estendidas para os resultados associados às implementações em FPGA mostrados na Tabela 4 e nas Figuras 5.3 e 5.4.

5.1.4 Capacidade de Representação das Estruturas Implementadas

A capacidade das estruturas implementadas em representar o sistema não linear deste exemplo é avaliada em termos do NMSE, conforme definido em (69). O sinal utilizado como entrada nessa avaliação é um ruído gaussiano branco de variância igual a 0,01. O NMSE obtido para as estruturas implementadas neste trabalho é apresentado na Tabela 5 com valores em escala decibel (dB). A partir de tais resultados, observa-se que as implementações do filtro Volterra baseadas nas estruturas de posto reduzido são mais precisas do que as implementações baseadas em coordenadas diagonais, quando considerada a abordagem numérica para remoção dos ramos. A imprecisão associada a esse último tipo de implementação está relacionada com o erro acumulado da quantização dos coeficientes da matriz triangular. No caso das estruturas de posto reduzido, a quantidade de coeficientes é bastante reduzida e, como consequência, o impacto do erro de quantização não é tão relevante.

Com respeito aos resultados de precisão associados às estruturas de posto reduzido e considerando o contexto específico de plataformas de *hardware* com 8 bits, observa-se, de acordo com a Tabela 5, que os valores de NMSE obtidos usando ambos os critérios de remoção de ramos são bastante similares. Desse modo, é válido supor que a abordagem numérica foi ineficiente ao permitir a implementação de um ramo com contribuição desprezível até mesmo em casos que se deseje primar pela precisão de representação do filtro Volterra implementado. Em contraste, para as plataformas de *hardware* com 12 e 16 bits, as estruturas definidas pela abordagem numérica são significativamente mais precisas do que as estruturas definidas pela abordagem energética. Observa-se também com as informações mostradas na Tabela 5, que as estruturas de posto reduzido convencionais e de posto reduzido proposta apresentam valores de NMSE semelhantes quando comparadas nos mesmos cenários de implementação. Dessa maneira e considerando os resultados apresentados na análise da complexidade, pode-se concluir que para o *kernel* do Exemplo 1, a estrutura de implementação proposta, dentre as estruturas de implementação do filtro Volterra consideradas, é a que apresenta os

melhores resultados tanto em capacidade de representação quanto em complexidade.

Tabela 5: Capacidade de representação do filtro Volterra do Exemplo 1 em termos do NMSE

Estrutura de implementação	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	X	-21, 85 dB	-43, 32 dB	-66, 89 dB
Posto Reduzido	Numérica	-27, 97 dB	-50, 03 dB	-72, 46 dB
Convencional	Energética	-27, 28 dB	-32, 31 dB	-32, 28 dB
Posto Reduzido	Numérica	-27, 84 dB	-50, 08 dB	-72, 49 dB
Proposto	Energética	-27, 39 dB	-32, 35 dB	-32, 28 dB

5.2 EXEMPLO 2

5.2.1 Descrição das Estruturas de Implementação

No Exemplo 2, é implementado um filtro Volterra de ordem quadrática cujos coeficientes foram extraídos de um problema real. O problema em questão envolveu a modelagem do caminho do eco de rede em um adaptador de telefone analógico (*analog telephone adapter* - ATA) usado em sistemas VoIP (*voice over internet protocol*)[6]. O *kernel* obtido com tal modelagem possui tamanho de memória igual a 25 e 625 coeficientes. A Figura 5.5 ilustra a distribuição dos coeficientes desse *kernel* com respeito à posição correspondente na matriz simétrica. É importante mencionar que o caminho do eco de rede do ATA foi modelado usando um filtro adaptativo com sinal de fala (não estacionário) como entrada. Nesse contexto, o ponto mínimo da superfície do erro quadrático médio varia com o tempo, dificultando, assim, o processo de rastreamento dos coeficientes ótimos [24]. Desse modo, o desajuste do erro pode apresentar um valor considerável e uma quantidade significativa de ruído será adicionada aos coeficientes adaptativos.

A implementação triangular do filtro Volterra para este exemplo é baseada na estrutura da Figura 2.3 e possui um total de 25 ramos implementados. Também são implementados com 25 ramos as estruturas de posto reduzido cuja remoção dos ramos é baseada nos critérios definidos na abordagem numérica. Já as estruturas que consideram a abordagem energética para definição dos ramos removidos são compostas por 19 de seus ramos mais significativos para $\zeta^{-1} = 1\%$. Note que em proporção ao posto completo da matriz de coeficientes, a quantidade de ramos removidos das estruturas do Exemplo 2 é muito menor do que no caso do Exemplo 1. Essa redução é consequência da importância dos ramos menos significativos da estrutura, que possivelmente tem origem no ruído de adaptação dos coeficientes. Nesse sentido, no processo de modelagem de sistemas não lineares visando à aplicação de filtros Volterra, a

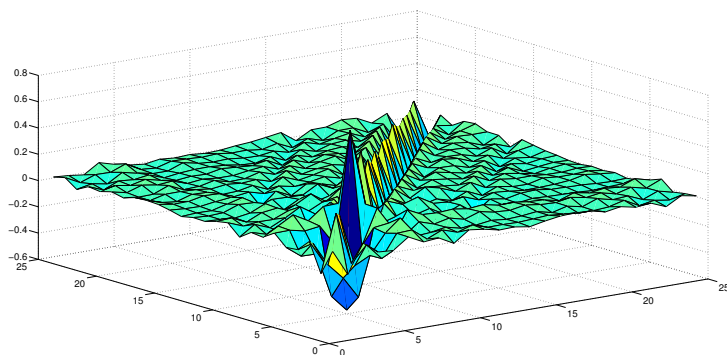


Figura 5.5: Matriz de coeficientes do *kernel* quadrático obtido da modelagem do caminho do eco de rede em um ATA.

redução do erro de adaptação é um aspecto que pode contribuir para redução de complexidade do sistema modelado. Com respeito especificamente à estrutura de implementação proposta, a quantidade de bits removida por ramo $[r_k$ de (45)] é dada na Tabela 6.

5.2.2 Resultados de Complexidade para as Implementações Consideradas

Os resultados de complexidade obtidos com as implementações do filtro Volterra do Exemplo 2 são apresentadas de forma similar aos resultados do Exemplo 1. A Tabela 7 e as Figuras 5.6 e 5.7 contêm as informações de complexidade em termos do número de somadores completos utilizados. Os resultados considerando a área ocupada em FPGA como métrica de complexidade são apresentados na Tabela 8 e nas Figuras 5.8 e 5.9. Nessa tabela, são destacados em **negrito** as implementações que requerem um número maior de blocos lógicos básicos do que o disponível no FPGA alvo (56480). Note, portanto, que devido à quantidade excessiva de ramos implementados nas estruturas de posto reduzido, o FPGA alvo não possui blocos lógicos suficientes para alguns dos cenários considerados. As estruturas de posto reduzido cujos ramos são removidos pela abordagem numérica têm os resultados de complexidade ilustrados nas Figuras 5.6(a)-5.9(a). Em contraste, as Figuras 5.6(b)-5.9(b) contêm a complexidade das estruturas em que o critério de remoção dos ramos considerado é o definido na abordagem energética. Finalmente, os resultados de complexidade das estruturas em que a forma direta é utilizada na implementação dos filtros FIR são apresentados nas Figuras 5.6 e 5.8 e as estruturas do filtro Volterra com filtros FIR implementados com a unidade de

Tabela 6: Número de bits removidos dos ramos da estrutura de posto reduzido proposta para o caso do Exemplo 2

Ramos em ordem crescente de significância	r_k	Ramos em ordem crescente de significância	r_k
1	0	15	1
2	0	16	1
3	0	17	1
4	0	18	1
5	0	19	1
6	0	20	1
7	0	21	1
8	0	22	1
9	0	23	2
10	0	24	2
11	1	25	3
12	1		
13	1		

multiplicação e acumulação têm seus resultados de complexidade ilustrados nas Figuras 5.7 e 5.9.

Tabela 7: Quantidade de somadores completos utilizados na implementação do *kernel* do Exemplo 2 para as diferentes estruturas consideradas

Estrutura de implementação	Topologia do filtro FIR	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	Forma Direta	X	13568	30588	53840
	Unidade MAC	X	4442	8938	15034
Posto Reduzido Convencional	Forma Direta	Númérica	27864	59796	102048
		Energética	22192	45696	77328
	Unidade MAC	Númérica	5048	11434	19050
		Energética	4334	8662	14414
Posto Reduzido Proposto	Forma Direta	Númérica	23354	52296	91216
		Energética	19330	41176	71412
	Unidade MAC	Númérica	3702	8091	13939
		Energética	3049	6354	10858

5.2.3 Análise da Complexidade das Implementações

Alguns dos resultados de complexidade obtidos com as implementações no Exemplo 2 carregam informações correspondentes às tratadas no Exemplo 1, como é o caso da variação da complexidade em função do tamanho de palavra da plataforma de *hardware*. Seria inconsistente nesses casos a realização de uma análise dos resultados de complexidade semelhante àquela rea-

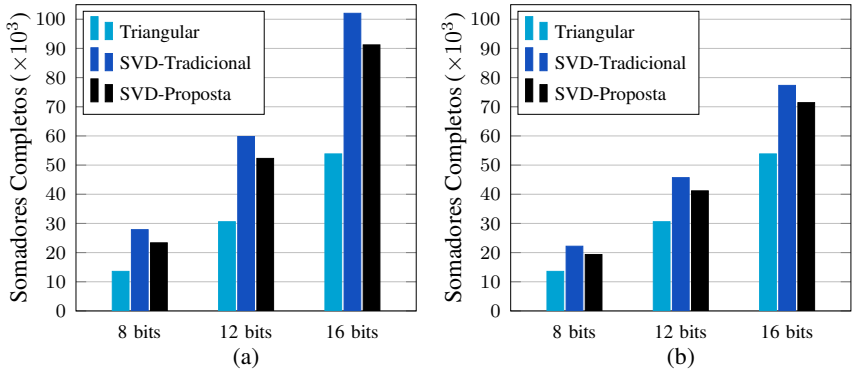


Figura 5.6: Complexidade das implementações do *kernel* do Exemplo 2 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

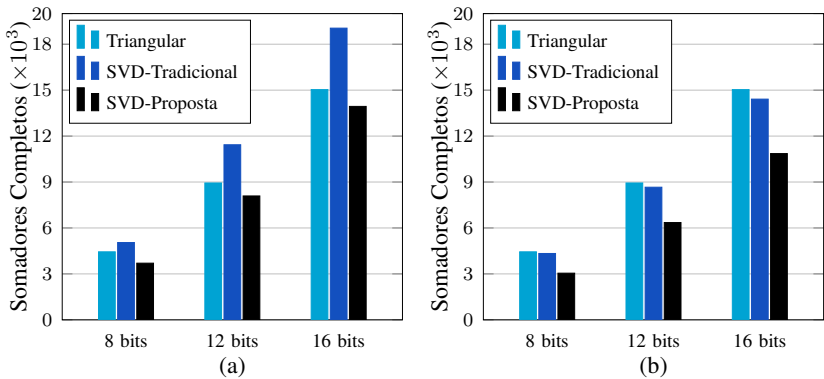


Figura 5.7: Complexidade das implementações do *kernel* do Exemplo 2 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

lizada para o Exemplo 1, visto a ausência de novas informações úteis. Nesse sentido, as discussões aqui apresentadas são focadas nos resultados que são essencialmente diferentes daqueles apresentados no Exemplo 1.

Um dos cenários de implementação que traz em seus resultados novas informações úteis envolve uma comparação de complexidade entre as estruturas de posto reduzido com as estruturas triangulares. Note nas Tabelas 7 e

Tabela 8: Ocupação de blocos lógicos em FPGA com a implementação do *kernel* do Exemplo 2 para as diferentes estruturas consideradas

Estrutura de implementação	Topologia do filtro FIR	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	Forma Direta	X	16115	29337	45470
	Unidade MAC	X	4681	9071	15349
Posto Reduzido Convencional	Forma Direta	Númérica	32492	60692	109347
		Energética	24805	43167	83192
	Unidade MAC	Númérica	4977	9776	16706
		Energética	3851	7577	12803
Posto Reduzido Proposto	Forma Direta	Númérica	30718	57201	103413
		Energética	23856	41046	79719
	Unidade MAC	Númérica	4124	8067	13850
		Energética	3258	6338	10808

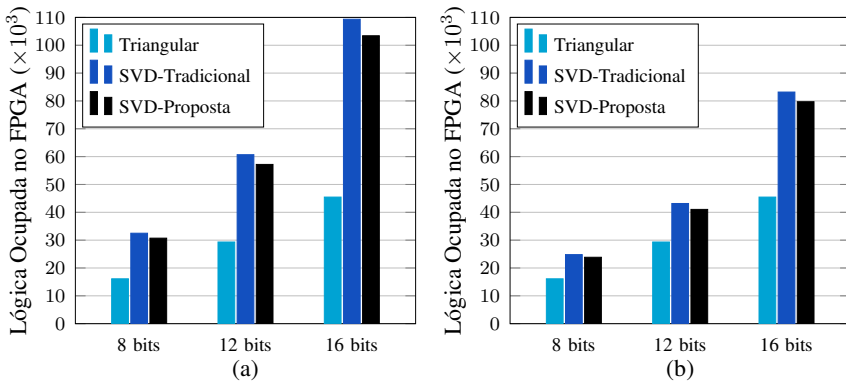


Figura 5.8: Complexidade das implementações do *kernel* do Exemplo 2 implementado em FPGA e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

8 (ou nas Figuras 5.6-5.9) que, muitos dos resultados obtidos para esta última estrutura apresentam complexidade menor do que os obtidos com a primeira. Mais especificamente, é vantajoso em termos de complexidade utilizar a estrutura triangular sempre que os filtros FIR forem implementados na forma direta (veja as Figuras 5.6 e 5.8). Observa-se tal fato pois poucos ramos da estrutura de posto reduzido associada ao *kernel* obtido da modelagem do ATA possuem significância desprezível. Como consequência, apenas os 6 ramos menos significativos são removidos com o critério de remoção estabelecido na abordagem energética e, portanto, são necessários 475 coeficientes (19 filtros FIR com tamanho de memória igual a 25) para implementar o filtro

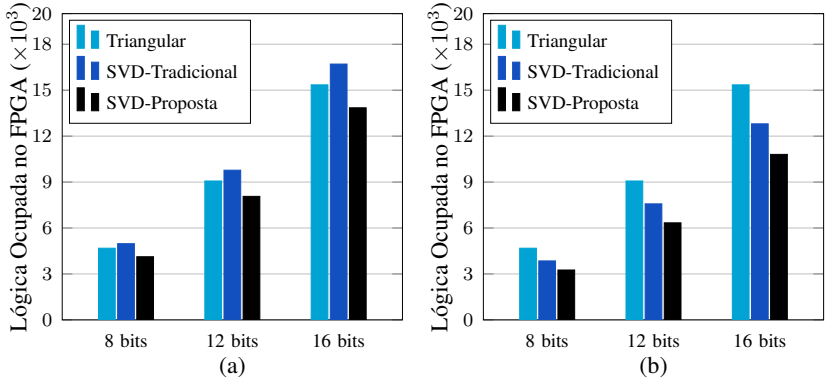


Figura 5.9: Complexidade das implementações do *kernel* do Exemplo 2 implementado em FPGA e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

Volterra. A quantidade de coeficientes requerida ao se considerar os critérios da abordagem numérica é ainda maior, chegando a um total de 625 (nesse caso todos os ramos da estrutura devem ser implementados). Em contraste, as implementações baseadas na estrutura triangular necessitam de 325 coeficientes [veja (13)]. Essa diferença em quantidade de coeficientes não apresenta grandes impactos no cenário em que os filtros FIR são implementados com uma unidade de multiplicação e acumulação, pois, em todos os casos, são consumidos apenas os somadores completos necessários para a implementação da unidade de multiplicação e acumulação por ramo. A vantagem em se utilizar a estratégia de implementação proposta nesse caso é que, mesmo no pior caso com os 25 ramos implementados, a redução do número de bits por ramo faz com que tal abordagem seja a mais eficiente para implementar o filtro Volterra. Contudo, quando a forma direta é considerada na implementação dos filtros FIR, a diferença em quantidade de coeficientes das estruturas é um fator determinante para que as estruturas triangulares tenham menor complexidade do que as de posto reduzido, conforme ilustrado nas Figuras 5.6 e 5.8. Com respeito às comparações apenas entre as estruturas de posto reduzido, observa-se que, assim como no Exemplo 1, os resultados de complexidade da estratégia de implementação proposta são significativamente inferiores aos da estrutura de posto reduzido convencional.

5.2.4 Capacidade de Representação das Estruturas Implementadas

Os valores de NMSE relativos às implementações do filtro Volterra do Exemplo 2 são apresentados na Tabela 9 em uma escala em dB. O sinal de entrada utilizado para a aquisição dos resultados apresentados foi um ruído branco gaussiano de variância igual a 0,01. Observe que, para uma determinada quantidade de bits, a estrutura triangular possui valores de NMSE semelhantes às estruturas definidas por ambas abordagens de posto reduzido e com os ramos removidos considerando a abordagem numérica. Mais especificamente, o NMSE mínimo possível para essas implementações de 8, 12 e 16 bits é da ordem de, respectivamente, -27 dB, -51 dB e -75 dB. Nos casos em que se utiliza a abordagem energética para remover os ramos das estruturas de posto reduzido, o NMSE mínimo fica limitado a valores de aproximadamente -29 dB. Os resultados da Tabela 9 indicam que, nos mesmos cenários de avaliação, a estrutura de implementação proposta apresenta resultados equivalentes aos obtidos com a estrutura de posto reduzido convencional. Portanto, as mesmas características de complexidade e capacidade de representação observadas para essas duas estruturas de posto reduzido no Exemplo 1 são também verificadas no Exemplo 2.

Tabela 9: Capacidade de representação do filtro Volterra do o Exemplo 2 em termos do NMSE

Estrutura de implementação	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	X	$-26,89$ dB	$-50,78$ dB	$-74,82$ dB
Posto Reduzido	Numérica	$-28,02$ dB	$-51,24$ dB	$-75,12$ dB
Convencional	Energética	$-25,83$ dB	$-29,13$ dB	$-29,15$ dB
Posto Reduzido	Numérica	$-27,24$ dB	$-51,29$ dB	$-75,18$ dB
Proposto	Energética	$-26,04$ dB	$-29,16$ dB	$-29,15$ dB

5.3 EXEMPLO 3

5.3.1 Descrição das Estruturas de Implementação

No Exemplo 3, é realizada a implementação de um filtro Volterra de terceira ordem com tamanho de memória N igual a 10. Os coeficientes desse filtro são adquiridos da modelagem de um circuito limitador usado em pedais de efeito de áudio [25]. O circuito limitador em questão foi modelado com um filtro adaptativo utilizando como entrada um ruído gaussiano branco, que é um sinal estacionário. Nesse contexto, é válido supor que o desajuste do erro dos coeficientes é inferior ao dos coeficientes modelados do ATA (Exemplo 2). Como consequência, o aumento da importância dos ramos menos significativos das estruturas de posto reduzido devido ao ruído de adaptação não será tão relevante quanto no Exemplo 2.

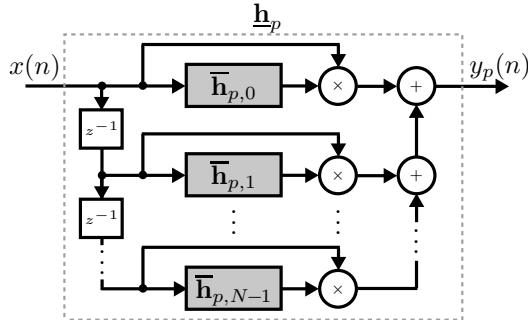


Figura 5.10: Diagrama de blocos para a implementação de um *kernel* de p -ésima ordem usando as decomposições de *kernels* com ordem elevada em *kernels* quadráticos.

As implementações do filtro Volterra de terceira ordem realizadas neste trabalho são baseadas no procedimento de decomposição de *kernels* com ordem elevada em *kernels* quadráticos discutida em [19]. Como resultado, obtém-se uma estrutura com ramos em paralelo semelhante à ilustrada no diagrama de blocos da Figura 5.10. Para este exemplo em especial, o i -ésimo *kernel* resultante da decomposição, representado na figura por $\bar{h}_{p,i}$, possui ordem quadrática ($p = 2$) e tamanho de memória igual a $10 - i$. Desse modo, a complexidade obtida com as implementações do filtro Volterra de terceira ordem pode ser definida pelo custo associado aos 10 *kernels* quadráticos, somado ao custo das operações de multiplicação e de soma efetuadas por ramo.

Nas implementações triangulares, o i -ésimo *kernel* quadrático ilustrado na Figura 5.10 é composto por $10 - i$ ramos em paralelo, onde o filtro FIR do j -ésimo ramo possui $10 - i - j$ coeficientes. No caso das estruturas de posto reduzido, a quantidade de ramos implementados é dependente da abordagem escolhida para remoção dos ramos e, obviamente, do nível de significância dos ramos. É relevante comentar que a importância de determinado ramo é avaliada considerando apenas o *kernel* quadrático do qual ele foi obtido; isto é, a avaliação da importância dos ramos do i -ésimo *kernel* é realizada considerando apenas os valores singulares obtidos da decomposição SVD sobre $\bar{h}_{p,i}$. A quantidade de ramos implementada para cada *kernel* quadrático decomposto e considerando ambas as abordagens para remoção dos ramos (com $\zeta^{-1} = 1\%$) é apresentada de forma resumida na Tabela 10. Observe que quase todos os ramos da estrutura são implementados se considerados os critérios da abordagem numérica. Nesse sentido, de maneira semelhante ao Exemplo 2, as implementações de posto reduzido baseadas nesses critérios devem apresentar complexidade correspondente (ou superior) aos resultados de complexidade associados à estrutura triangular. Em contraste, uma quan-

mente, as Tabelas 12 e 13 contêm, respectivamente, o custo de implementação em termos do: i) número de somadores completos utilizados; e da ii) quantidade de blocos lógicos básicos utilizada do FPGA alvo. O custo de implementação de i) é ilustrado graficamente nas Figuras 5.11 e 5.12, sendo que na primeira figura os filtros FIR são sintetizados na forma direta e na segunda, com a unidade de multiplicação e acumulação. Por outro lado, as Figuras 5.13 e 5.14 carregam os resultados de complexidade obtidos das implementações em FPGA e, analogamente ao caso considerando os somadores completos, na primeira figura os filtros FIR são sintetizados na forma direta e na segunda, com a unidade de multiplicação e acumulação. Finalmente, as estruturas de posto reduzido que tiveram seus ramos removidos de acordo com as abordagens numérica e energética têm seus resultados apresentados, respectivamente, nas Figuras 5.11(a)- 5.14(a) e 5.11-5.14(b).

Tabela 12: Quantidade de somadores completos utilizados na implementação do *kernel* do Exemplo 3 para as diferentes estruturas consideradas

Estrutura de implementação	Topologia do filtro FIR	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	Forma Direta	X	17704	37020	63312
	Unidade MAC	X	10313	21129	35785
Posto Reduzido Convencional	Forma Direta	Numérica	19952	51600	89088
		Energética	13448	27660	46864
	Unidade MAC	Numérica	10749	26877	45964
		Energética	8071	16207	27127
Posto Reduzido Proposto	Forma Direta	Numérica	15440	40668	72128
		Energética	10600	22572	38992
	Unidade MAC	Numérica	7757	19449	33868
		Energética	5831	12031	20471

Tabela 13: Ocupação de blocos lógicos em FPGA com a implementação do *kernel* do Exemplo 3 para as diferentes estruturas consideradas

Estrutura de implementação	Topologia do filtro FIR	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	Forma Direta	X	13922	27130	55327
	Unidade MAC	X	10197	20384	32753
Posto Reduzido Convencional	Forma Direta	Numérica	18301	39806	72358
		Energética	11175	19775	35360
	Unidade MAC	Numérica	9011	22422	44235
		Energética	6518	12816	25430
Posto Reduzido Proposto	Forma Direta	Numérica	14425	32721	60032
		Energética	9144	16402	32841
	Unidade MAC	Numérica	6192	15326	30255
		Energética	4565	9358	16611

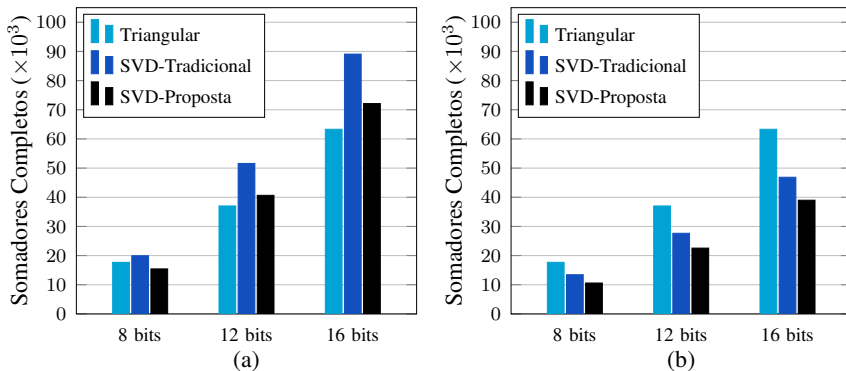


Figura 5.11: Complexidade das implementações do *kernel* do Exemplo 3 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

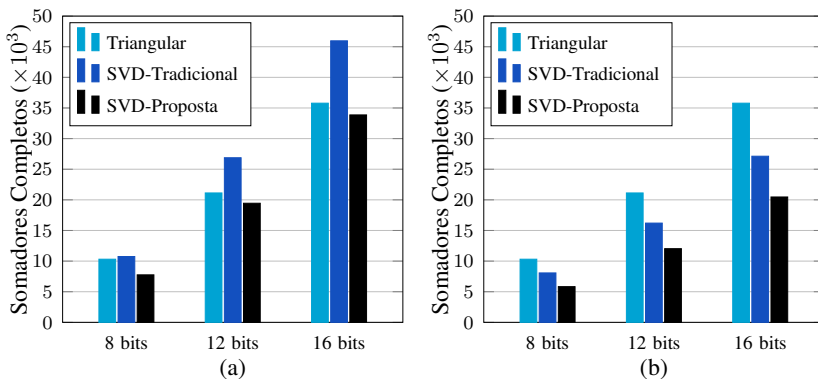


Figura 5.12: Complexidade das implementações do *kernel* do Exemplo 3 em termos da quantidade de somadores utilizados e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

5.3.3 Análise da Complexidade das Implementações

A partir dos resultados mostrados nas Tabelas 12 e 13 e ilustrados nas Figuras 5.11-5.14 conclui-se que as implementações baseadas na estrutura de posto reduzido proposta são as que apresentam a menor complexidade de implementação na maioria dos cenários. Em contraste com os resultados do

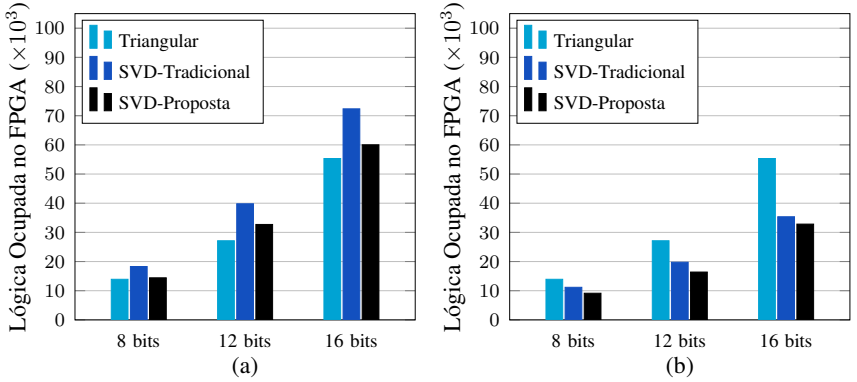


Figura 5.13: Complexidade das implementações do *kernel* do Exemplo 3 implementado em FPGA e com os filtros FIR sintetizados na forma direta. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

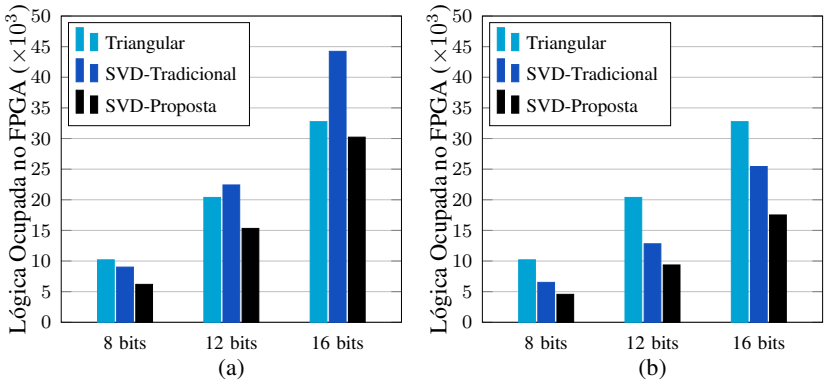


Figura 5.14: Complexidade das implementações do *kernel* do Exemplo 3 implementado em FPGA e com os filtros FIR sintetizados com uma unidade de multiplicação e acumulação. Os resultados apresentados consideram que os ramos removidos da estrutura de posto reduzido são definidos pela (a) abordagem numérica e (b) pela abordagem energética.

Exemplo 2, uma quantidade importante dos ramos da estrutura é considerada com significância desprezível se forem utilizados os critérios da abordagem energética com $\zeta^{-1} = 1\%$. Mais especificamente, para esses critérios, são removidos 25 dos 55 ramos (45%) das estruturas de posto reduzido que representam o filtro Volterra de terceira ordem, conforme apresentado na Tabela 10. Os cenários em que os resultados de complexidade das estruturas

triangulares são inferiores às de posto reduzido consideram a utilização da abordagem numérica para remoção dos ramos da estrutura de posto reduzido e a topologia de síntese dos filtros FIR na forma direta. Note, portanto, que mesmo considerando estruturas com a máxima capacidade de representação para um dado tamanho de palavra, a estrutura de posto reduzido proposta ainda apresenta complexidade inferior à triangular se os filtros FIR são implementados com a unidade de multiplicação e acumulação. As implementações do Exemplo 3 também permitem uma interessante análise de complexidade envolvendo ambas estruturas de posto reduzido consideradas. Em contraste aos Exemplos 1 e 2, o ganho em utilizar a abordagem de implementação proposta é relativamente significativa mesmo nos casos em que os filtros são implementados na forma direta. Esse acréscimo do ganho obtido com a estrutura de posto reduzido proposta é consequência da redução total de bits alcançada. Nos casos em que se considera a abordagem numérica para remoção dos ramos, por exemplo, a economia obtida pode alcançar um total de 124 bits de um total de 880 bits (14%).

5.3.4 Capacidade de Representação das Estruturas Implementadas

O NMSE dos filtros Volterra de terceira ordem implementados é apresentado na Tabela 14 em escala dB. Note que o valor mínimo do NMSE obtido com as estruturas implementadas em plataformas de *hardware* com precisão de 8, 12 e 16 bits é da ordem de, respectivamente, -22 dB, -43 dB e -68 dB. Com respeito às estruturas de posto reduzido, observa-se que o valor do NMSE para as estruturas cujos ramos removidos são escolhidos pelos critérios da abordagem energética estão na faixa de -25 dB. Essas estruturas apresentam um erro ligeiramente superior para os cenários considerando *hardwares* com precisão de 8 bits (-22 dB). Contudo, é consistente supor que nesse caso, o erro de quantização tem maior importância na saída do sistema do que o erro gerado na remoção dos ramos da estrutura de posto reduzido. Os resultados considerando as estruturas de posto reduzido em que a escolha dos ramos removidos foi baseada na abordagem numérica são semelhantes aos observados para as estruturas de implementação triangular.

Tabela 14: Capacidade de representação do filtro Volterra do Exemplo 3 em termos do NMSE

Estrutura de implementação	Abordagem de remoção dos ramos	8 bits	12 bits	16 bits
Triangular	X	$-20,59$ dB	$-44,07$ dB	$-68,24$ dB
Posto Reduzido Convencional	Numérica	$-22,74$ dB	$-43,15$ dB	$-67,24$ dB
Posto Reduzido Proposto	Energética	$-21,07$ dB	$-24,50$ dB	$-24,54$ dB
	Numérica	$-23,01$ dB	$-43,27$ dB	$-68,84$ dB
	Energética	$-21,19$ dB	$-24,47$ dB	$-24,54$ dB

5.4 CONCLUSÕES

Neste capítulo, foram apresentados os resultados de complexidade de implementação e de capacidade de representação de filtros Volterra implementados com as estruturas triangular, de posto reduzido convencional e de posto reduzido proposta. As implementações consideraram três *kernels* não lineares como exemplos, sendo dois destes quadráticos e um de terceira ordem. Além disso, foram considerados diversos cenários para a implementação dos filtros Volterra com o objetivo de avaliar as estruturas em diferentes contextos de operação do filtro. Com os resultados obtidos nas implementações do Exemplo 1, foi possível observar que para os *kernels* com uma quantidade relativamente baixa de ramos significativos, a utilização de estruturas de posto reduzido é vantajosa tanto em termos de complexidade de implementação quanto em termos de precisão de representação da não linearidade. No Exemplo 2, observou-se uma desvantagem significativa em termos de complexidade de implementação ao utilizar as estruturas de posto reduzido na representação de *kernels* adquiridos de ambientes ruidosos. Nesse caso, poucos dos ramos das estruturas atenderam aos critérios adotados para remoção e, como consequência, apresentaram custo de implementação superior ao das estruturas triangulares em diversos cenários. Contudo, é possível obter implementações mais eficientes de filtros Volterra no caso do Exemplo 2 se forem adotados outros parâmetros para a remoção dos ramos. Por exemplo, caso a abordagem energética fosse utilizada com $\zeta^{-1} = 5\%$, a quantidade de ramos removidos seria igual a 11, resultando em um total de 121 coeficientes para implementação do filtro Volterra (quantidade quase três vezes inferior ao número de coeficientes requerido nas implementações triangulares). No terceiro exemplo, foi apresentada a implementação de um filtro Volterra com não linearidade de terceira ordem. Nesse caso, as estruturas de posto reduzido apresentaram reduções significativas em complexidade de implementação, quando comparadas às estruturas triangulares, se considerada a abordagem energética para remoção dos ramos. Em todas as implementações realizadas, as estruturas de posto reduzido convencional e proposta apresentaram resultados semelhantes de capacidade de representação se avaliadas nos mesmos cenários. Por outro lado, ao comparar a complexidade das implementações baseadas nessas duas estruturas, observa-se que em todos os casos o uso da estratégia de implementação proposta é vantajoso.

6 CONCLUSÕES E CONSIDERAÇÕES FINAIS

Neste trabalho, uma estratégia para implementação de filtros Volterra com complexidade reduzida foi apresentada. As discussões envolvidas no desenvolvimento de tal estratégia e os principais resultados obtidos são apresentados na próxima seção. Ao final deste capítulo, são sugeridas algumas propostas de trabalhos futuros.

6.1 SUMÁRIO E DISCUSSÃO DOS RESULTADOS

No primeiro capítulo desta dissertação, uma breve análise abrangendo alguns aspectos de sistemas não lineares foi apresentada. Os pontos abordados em tal análise serviram como fundamento para orientar os esforços durante o desenvolvimento do trabalho para a busca de novas estruturas de implementação dos filtros Volterra com complexidade reduzida. Na sequência, no Capítulo 2, uma discussão acerca das principais características dos filtros Volterra foi realizada. Dentre essas características, destacaram-se o elevado número de coeficientes requerido para a representação de não linearidades com os filtros Volterra e, como consequência, o alto custo associado às implementações desses filtros. Considerando que essas são características indesejadas, pois limitam o uso dos filtros Volterra, algumas das estratégias comumente utilizadas para contornar o problema em questão também foram apresentadas. Especificamente, foram tratadas as estruturas triangular e de posto reduzido convencional que são utilizadas para representar os filtros Volterra com quantidade menor de coeficientes, sendo esta última estrutura representada na forma de ramos em paralelo. No contexto das estruturas de posto reduzido, foi destacada a possibilidade de implementar filtros Volterra desconsiderando os ramos menos significativos da estrutura às expensas de uma perda de precisão muitas vezes insignificante.

A estratégia de implementação proposta foi discutida no Capítulo 3. Tal estratégia teve como objetivo explorar os diferentes graus de significância dos ramos de uma estrutura de posto reduzido para definir o tamanho de palavra usado nas implementações de cada ramo. Para isso, inicialmente a estrutura de posto reduzido convencional foi reorganizada e, então, a probabilidade de *overflow* de cada ramo, em função da importância correspondente ao ramo, foi analisada. Na sequência, o número de bits removido por ramo foi definido mantendo a probabilidade de *overflow* na entrada de cada ramo igual ou menor do que a probabilidade de *overflow* na entrada da estrutura. Também foram abordados no Capítulo 3 os aspectos de implementação das estruturas consideradas neste trabalho, visando definir uma métrica para avaliação de complexidade da estratégia de implementação proposta. Nesse sentido, foram revisadas as topologias básicas para síntese digital de operações de soma, multiplicação e filtragem consideradas neste trabalho. A partir disso,

expressões analíticas para avaliar a complexidade em termos do número de somadores completos utilizados nas implementações das estruturas triangular, de posto reduzido convencional e de posto reduzido proposto foram determinadas. Tais expressões foram utilizadas no Capítulo 5 para a análise de complexidade das estruturas implementadas.

No Capítulo 4, foram apresentadas as abordagens para a definição de critérios para remoção dos ramos de estruturas de posto reduzido, as quais foram denominadas abordagem numérica e abordagem energética. Os critérios definidos pela primeira abordagem buscam a implementação de um dado filtro Volterra com a estrutura de posto reduzido de melhor capacidade de representação possível, considerando as características do *hardware* alvo. Já na segunda abordagem (a energética) são removidos os ramos cuja razão entre a energia do sinal em sua saída e a energia na saída do ramo de maior significância seja inferior a um valor de limiar. Visto que essa razão entre energias depende exclusivamente dos valores singulares associados aos ramos da estrutura de posto reduzido, a condição de implementação dos ramos apresentada está intimamente relacionada ao grau de significância relativo aos ramos avaliados.

Finalmente, no Capítulo 5, os resultados de complexidade e capacidade de representação da estrutura obtida com a estratégia de implementação proposta foram avaliados. Tal avaliação é realizada por meio de uma análise comparativa entre os resultados de implementação associados à estrutura proposta e os obtidos com as estruturas triangular e de posto reduzido convencional. Para a aquisição de tais resultados, são considerados três filtros Volterra como exemplos, os quais representam não linearidades com diferentes características. Além disso, diversos cenários de implementação foram considerados para cada um desses exemplos. De modo geral, nos casos em que as estruturas triangulares exigiram uma quantidade de coeficientes significativamente maior do que às de posto reduzido, o erro associado àquela primeira estrutura foi maior em termos do NMSE se considerada as estruturas de posto reduzido com a capacidade máxima de representação possível. Nos outros casos, essas estruturas apresentaram capacidade de representação bastante similar. Com respeito à complexidade, a estrutura obtida com a abordagem de implementação proposta foi a que obteve os melhores resultados. O ganho de complexidade ao utilizar a estrutura proposta varia proporcionalmente com o cenário em que a implementação é considerada. Para o Exemplo 1, em que muitos dos ramos das estruturas de posto reduzido foram removidos para ambos os critérios de remoção apresentados neste trabalho, a redução em complexidade obtida com a estrutura proposta em comparação com as demais foi bastante significativa. Em contraste, no Exemplo 2, em que poucos ramos das estruturas de posto reduzido satisfizeram os critérios de remoção dos ra-

mos adotados neste trabalho, a complexidade das estruturas triangulares e da proposta foram similares nos diversos casos considerados.

6.2 SUGESTÕES PARA TRABALHOS FUTUROS

Algumas sugestões para a continuidade deste trabalho são elencadas a seguir:

- Desenvolver uma nova estratégia de implementação de filtros Volterra com a mesma filosofia da apresentada neste trabalho e considerando um sinal de entrada genérico para o seu desenvolvimento.
- Estender o desenvolvimento da estratégia de implementação proposta considerando outras estruturas de implementação de filtros Volterra.
- Verificar a possibilidade de aplicar a estratégia de implementação proposta em *kernels* Volterra recursivos.
- Realizar um estudo aprofundado da energia na saída da estrutura de posto reduzido para o sinal de interesse como também para os ruídos de quantização e decorrente da remoção dos ramos.

REFERÊNCIAS

- [1] B. P. Lathi, *Linear Systems and Signals*, 2nd ed. Oxford, UK: Oxford University Press, 2009.
- [2] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*. New York: John Wiley & Sons, Inc., 2000.
- [3] L. Tan and J. Jiang, “Adaptive Volterra filters for active control of nonlinear noise processes,” *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1667–1676, Aug. 2001.
- [4] A. Stenger, L. Trautmann, and R. Rabenstein, “Nonlinear acoustic echo cancellation with second-order adaptive Volterra filters,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Phoenix, USA, Mar. 1999, pp. 877–880.
- [5] D. Zhou and V. De Brunner, “Novel adaptive nonlinear predistorters based on the direct learning algorithm,” *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 120–133, Jan. 2007.
- [6] E. L. O. Batista and R. Seara, “A novel reduced-rank approach for implementing Volterra filters,” in *Proc. Eur. Signal Process. Conf.*, Budapest, Hungary, Aug. 2016, pp. 1778–1782.
- [7] A. Carini and G. L. Sicuranza, “Even mirror Fourier nonlinear filters,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, Canada, May 2013, pp. 5608–5612.
- [8] M. Fréchet, *Sur les fonctionnelles continues*. Gauthier-Villars, 1910.
- [9] H.-H. Chiang, C. L. Nikias, and A. N. Venetsanopoulos, “Efficient implementations of quadratic digital filters,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 6, pp. 1511–1528, Dec. 1986.
- [10] Y. Lou, C. L. Nikias, and A. N. Venetsanopoulos, “Efficient VLSI array processing structures for adaptive quadratic digital filters,” *Circuits, Syst. Signal Process.*, vol. 7, no. 2, Jun. 1988.
- [11] S. Marsi and G. L. Sicuranza, “On reduced-complexity approximations of quadratic filters,” in *Conf. Rec. 27th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, Pacific Grove, CA, Nov. 1993, pp. 1026–1030.
- [12] R. D. Nowak and B. D. Van Veen, “Tensor product basis approximations for Volterra filters,” *IEEE Trans. Signal Process.*, vol. 44, no. 1, pp. 36–50, Jan. 1996.

- [13] T. M. Panicker, V. J. Mathews, and G. L. Sicuranza, "Adaptive parallel-cascade truncated Volterra filters," *IEEE Trans. Signal Process.*, vol. 46, no. 10, pp. 2664–2673, 1998.
- [14] G. Favier, A. Y. Kibangou, and T. Bouilloc, "Nonlinear system modeling and identification using Volterra-PARAFAC models," *Int. J. Adapt. Control Signal Process.*, vol. 26, no. 1, pp. 30–53, Jan. 2012.
- [15] E. L. O. Batista, "Filtros volterra adaptativos: estruturas interpoladas e modelos estocásticos," Ph.D. dissertation, Universidade Federal de Santa Catarina, 2009.
- [16] E. L. O. Batista, O. J. Tobias, and R. Seara, "A sparse-interpolated scheme for implementing adaptive Volterra filters," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2022–2035, Apr. 2010.
- [17] G. Palm and T. Poggio, "The volterra representation and the wiener expansion: validity and pitfalls," *SIAM Journal on Applied Mathematics*, vol. 33, no. 2, pp. 195–216, 1977.
- [18] T. M. Panicker and V. Mathew, "Parallel-cascade realizations and approximations of truncated volterra systems," *IEEE Transactions on Signal Processing*, vol. 46, no. 10, pp. 2829–2832, 1998.
- [19] E. L. O. Batista and R. Seara, "A reduced-rank approach for implementing higher-order Volterra filters," *EURASIP Journal on Advances in Signal Process.*, vol. 118, no. 1, pp. 1–8, Nov. 2016.
- [20] S. Kay, *Intuitive Probability and Random Processes Using MATLAB*. New York: John Wiley & Sons, Inc., 2006.
- [21] D. Harris and S. Harris, *Digital design and computer architecture*. Morgan Kaufmann, 2010.
- [22] D. G. Manolakis and V. K. Ingle, *Applied digital signal processing: theory and practice*. Cambridge University Press, 2011.
- [23] S. do Couto Mendes, F. Cristina Alves Branco-Barreiro, and S. Frota, "Limiar diferencial de mascaramento: valores de referência em adultos," vol. 22, 01 2017.
- [24] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and nonstationary learning characteristics of the lms adaptive filter," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151–1162, 1976.

- [25] G. C. K. da Silva, “Contribuições para a implementação digital de um pedal de efeito de áudio do tipo overdrive,” Ph.D. dissertation, Universidade Federal de Santa Catarina, 2016.