FERNANDO GONÇALVES DE ALMEIDA NETO

ANÁLISE DE FILTROS DIGITAIS IMPLEMENTADOS EM ARITMÉTICA DE PONTO FIXO USANDO CADEIAS DE MARKOV

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

São Paulo
2011

FERNANDO GONÇALVES DE ALMEIDA NETO

ANÁLISE DE FILTROS DIGITAIS IMPLEMENTADOS EM ARITMÉTICA DE PONTO FIXO USANDO CADEIAS DE MARKOV

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de Concentração:

Sistemas Eletrônicos

Orientador:

Prof. Dr. Vítor H. Nascimento

São Paulo
2011

Aos meus pais, Rosângela e Fernando, por todo amor, apoio e incentivo ao longo de toda minha formação.

AGRADECIMENTOS

Ao amigo e orientador Prof. Vítor, que me acompanha e orienta desde o início da minha formação como Engenheiro.

À Amanda, sempre carinhosa e compreensiva nos momentos de maior dificuldade.

À minha irmã Renata, pelo carinho e incentivo nos momentos de tensão.

Aos amigos Murilo, Vítor, Wilder e Wesley, com os quais tive frutíferas discussões e que partilharam comigo o dia-a-dia como alunos de mestrado

Aos amigos Gleison, Mirele, Felipe e Fernando, que desde a graduação me incentivaram a realizar meu mestrado.

Ao amigo Alexandre, que sempre esteve presente com bons conselhos nos momentos de maior pressão.

Aos meus amigos Danilo e Ivo, pelo apoio desde o início deste trabalho.

Ao meu primo Eduardo, que há muito tempo me mostrou que a Engenharia seria o meu caminho ideal.

RESUMO

Uma forma de se reduzir o custo (em termos tanto de área de chip quanto de consumo de energia) de algoritmos de processamento de sinais é empregar aritmética de ponto fixo, usando o menor número de bits possível para se representar as variáveis e coeficientes necessários. Com isso, consegue-se reduzir a complexidade do hardware, levando a economias de energia e de área de chip em circuitos dedicados. A escolha do nível de quantização a que cada variável deve ser submetida depende de se conhecer o efeito da quantização de cada variável nas saídas do sistema, o que pode ser conseguido através de simulações (em geral lentas) ou por métodos analíticos. Este documento propõe avanços a uma nova metodologia de análise de algoritmos para processamento digital de sinais implementados em aritmética de ponto fixo, usando modelos baseados em cadeias de Markov.

As contribuições desta dissertação são as seguintes:

Filtros IIR de primeira e de segunda ordem são analisados via cadeia de Markov, pressupondo que a entrada possui uma função densidade de probabilidade conhecida. O modelo é desenvolvido de forma geral, de forma que pode ser considerada uma função de densidade de probabilidade qualquer. A saída dos filtros é usada para definir os estados da cadeia.

O modelo via cadeia de Markov para o coeficiente do algoritmo LMS unidimensional é estendido para entrada correlacionada. Nesse caso, os estados passam a ser descritos em termos do coeficiente e do da entrada anterior. Um exemplo assumido função de densidade de probabilidade de entrada gaussiana para o filtro adaptativo é apresentado.

ABSTRACT

The implementation cost of signal processing algorithms may be reduced by using fixed-point arithmetic with the smallest possible word-length for each variable or parameter. This allows the designer to reduce hardware complexity, leading to economy of energy and chip area in dedicated circuits. The choice of word-length depends on the determination of the effect at the output of the quantization of each variable, which may be obtained through simulations (generally slow) or through analytical methods. This document proposes new advances to a new analysis method for digital signal processing algorithms implemented in fixed-point arithmetic, based on Markov chain models.

Our contributions are the following:

A Markov chain model is used to study first and second order IIR filters for an known input density probability function. The model is general and can be applied for any probability function. We use the output of the filters to define the states of the Markov chain.

The unidimensional LMS Markov chain model is extended to correlated input. The states are defined by a pair considering the coefficient and the previous input and an example assuming Gaussian-distributed input is presented.

SUMÁRIO

Lista de Figuras **vii**

Lista de Tabelas **x**

1 Introdução **11**

2 Probabilidades e Cadeias de Markov **14**

2.1 Propriedades básicas de probabilidades 14

2.1.1 Independência de eventos 15

2.2 Probabilidade condicionada 15

2.2.1 Teorema da probabilidade total 15

2.2.2 Teorema de Bayes 17

2.3 Variáveis aleatórias e distribuições de probabilidade 17

2.4 Função de probabilidade acumulada 18

2.5 Função de densidade de probabilidade 21

2.6 Função densidade de probabilidade de somas 22

2.7 Funções de probabilidade consideradas no escopo deste texto . . . 23

2.7.1 Variável aleatória uniforme 23

2.7.2 Variável aleatória normal ou gaussiana 24

2.7.3 Probabilidade de entrada do filtro digital 25

2.8 Funções de probabilidade com duas variáveis aleatórias 26
2.9 Funções de probabilidade condicionada 28
2.10 Definição de esperança matemática e variância 29
2.10.1 Esperança matemática 29
2.10.2 Variância 29
2.11 Cadeias de Markov de tempo discreto 29
2.11.1 Cálculo da probabilidade depois de n passos 31
2.11.1.1 Classificação dos estados e definição de classes . . 32
2.11.1.2 Comportamento em regime estacionário 35

3 Filtros digitais fixos e efeitos de precisão finita 40

3.1 Filtros digitais 40
3.1.1 Formas diretas 41
3.1.2 Realização em cascata 41
3.1.3 Realização paralela 43
3.2 Efeitos de precisão finita 44
3.2.1 Sistema binário 45
3.2.2 Representação em ponto fixo 46
3.2.2.1 Representação em sinal-módulo 46
3.2.2.2 Representação em complemento-a-um 46
3.2.2.3 Representação em complemento-a-dois 47
3.2.3 Representação em ponto flutuante 47

3.2.4 Quantização 48

3.2.4.1 *Overflow* e saturação 51

3.2.5 Escalamento da entrada 53

3.2.6 Quantizações de coeficientes 57

3.2.7 Ciclos-limite 59

3.2.7.1 Ciclos-limite granulares 59

3.2.7.2 Ciclos-limite por *overflow* 59

3.2.7.3 Evitando ciclos-limite 60

4 Filtros digitais com acumulador de precisão simples 64

4.1 Implementação com acumulador de precisão simples 64

4.2 Probabilidade de saturação usando cadeias de Markov 67

4.2.1 Probabilidades em um filtro de segunda ordem 71

4.2.1.1 Densidade de probabilidade da entrada 72

4.2.1.2 Cálculo da probabilidade de entrada 77

4.2.1.3 Cálculo da probabilidade de entrada 79

4.2.1.4 Probabilidade condicionada de $y(n)$ 81

4.2.2 Probabilidades em um filtro de primeira ordem 83

4.3 Probabilidade de *overflow* usando cadeias de Markov 84

4.3.1 Probabilidades em um filtro de segunda ordem 88

4.3.1.1 Densidade de probabilidade da entrada 88

4.3.1.2 Cálculo da probabilidade de entrada 89

4.3.1.3 Cálculo da probabilidade de entrada 91
4.3.1.4 Probabilidade condicionada de $y(n)$ 91
4.3.2 Probabilidades em um filtro de primeira ordem 92
4.4 Análise usando não-linearidade de saturação 93
4.4.1 Escalamento da entrada 93
4.4.2 Comparação com o modelo linear 97
4.5 Análise usando não-linearidade de *overflow* 101
4.5.1 Comparação com o modelo linear 101
4.6 Identificação de ciclos-limite de entrada zero 103
4.6.1 Ciclos-limite de entrada nula 104
4.6.2 Encontrando os ciclos-limite de entrada nula 108
4.7 Cadeias de Markov para a análise de filtros em cascata 112
4.7.1 Implementação de filtros passa-tudo 115
4.7.2 Exemplos com filtros passa-tudo 115
4.7.2.1 Cálculo com o modelo linear 116
4.7.2.2 Exemplo 1 120
4.7.2.3 Exemplo 2 123
4.7.2.4 Exemplo 3 125
4.7.3 Exemplos com filtros passa-baixa 127
4.7.3.1 Modelo linear para filtro passa-baixa 130
4.7.3.2 Exemplo 1 131
4.7.3.3 Exemplo 2 133

4.7.4 Comentários finais da análise com filtros concatenados . . 134

5 Filtros digitais com acumulador de precisão dupla 137

5.1 Probabilidades do filtro de segunda ordem: saturação 138

5.2 Probabilidades do filtro de primeira ordem: saturação 140

5.3 Probabilidades do filtro de segunda ordem: *overflow* 141

5.4 Probabilidades do filtro de primeira ordem: *overflow* 142

5.5 Simulações 142

5.5.1 Exemplo considerando a não-linearidade de saturação . . . 143

5.5.2 Exemplo considerando a exceção de *overflow* 145

6 Efeitos de precisão finita no algoritmo LMS 148

6.1 O algoritmo LMS e a precisão finita 148

6.2 Modelo em precisão finita 150

6.2.1 Função de probabilidade correlacionada 151

6.2.2 Cálculo da probabilidade de $d_Q(n)$ 153

6.2.3 Cálculo da probabilidade de $e_Q(n)$ 153

6.2.4 Cálculo da probabilidade de $y_Q(n)$ 154

6.2.5 Cálculo da probabilidade de $w(n+1)$ 155

6.3 Teste do modelo com cadeias de Markov 155

6.3.1 O *overflow* no LMS 155

6.3.2 Comparação do MSE 157

6.3.3 Simulações 160

6.3.4 Comparação da estimativa do coeficiente 162

7 Conclusões **164**

Referências **168**

Anexo A – Artigo para o ITS2010 **170**

LISTA DE FIGURAS

1 Diagrama de Venn 16

2 Mapeamento de variável aleatória 18

3 Probabilidade acumulada da variável aleatória $\boldsymbol{X}$ 20

4 Densidade de probabilidade da variável aleatória $\boldsymbol{X}$ 22

5 Densidade de probabilidade antes e depois da amostragem 26

6 Diagrama de transição de estados 31

7 Diagrama de transição de estados mostrando estados recorrentes . 33

8 Cadeias de Markov composta somente por classes recorrentes . . . 34

9 Cadeias de Markov com classes recorrentes e estados transientes . 34

10 Classes recorrentes de periodicidade 3 35

11 Classe única de recorrência aperiódica 37

12 Filtro IIR implementado na forma direta I 42

13 Filtro IIR implementado na forma direta II 42

14 Cascata de filtros IIR na forma direta II 43

15 Implementação paralela de filtros IIR na forma direta II 44

16 Arredondamento em uma implementação de 3 bits 49

17 Truncamento em uma implementação de 3 bits 50

18 Modelo do erro de quantização após multiplicador 50

19 Exceção de *overflow* para um sinal de 3 bits 53

20 Exceção de saturação 53

21 Escalamento da entrada de um filtro digital 54

22 Filtro digital com quantizadores 60

23 Filtro digital com quantizadores 61

24 Não-linearidades de precisão finita em filtro de segunda ordem . . 66

25 Não-linearidades de precisão finita em filtro de primeira ordem . . 66

26 Não-linearidade de saturação para implementação de 2 bits 73

27 Exemplo de saturação usando estados adicionais 74

28 Média e variância calculadas por cadeias de Markov 99

29 Média e variância calculadas por cadeias de Markov 100

30 Média e variância da saída calculadas com de cadeias de Markov . 103

31 Saída do filtro digital do exemplo, para entrada nula 110

32 Filtro passa-tudo com pólos pequenos, usando saturação 120

33 Filtro passa-tudo com pólos pequenos, usando *overflow* 121

34 Filtro passa-tudo com pólos próximos da circ. unitária: saturação 123

35 Filtro passa-tudo com pólos próximos da circ. unitária: *overflow* . 124

36 Filtro com pólo próximo e distante da circunferência unitária . . . 126

37 Filtro com pólo próximo e distante da circunferência unitária . . . 127

38 Filtro com pólo próximo e distante da circunferência unitária . . . 128

39 Filtro com pólo próximo e distante da circunferência unitária . . . 129

40 Filtro com pólo próximo de zero 132

41 Filtro com pólo próximo de zero 132

42 Filtro com pólo próximo da circunferência unitária 134

43 Filtro com pólo próximo da circunferência unitária 135

44 Não-linearidades de precisão finita em filtro de segunda ordem . . 137

45 Não-linearidades de precisão finita em filtro de primeira ordem . . 138

46 Variância e média via Markov e via modelo linearizado 144

47 Variância e média via Markov e via modelo linearizado 147

48 LMS com blocos modelando não-linearidades 151

49 *Overflow* em implementação do LMS 156

50 LMS com modelo de fontes de ruído de quantização 157

51 MSE da abordagem linearizada, via Markov e implementada . . . 161

52 Estimativa do coeficiente do filtro adaptativo 163

LISTA DE TABELAS

1 Representação binária para valores entre −1 e 1 52

2 Média e variância, considerando saturação 122

3 Média e variância, considerando *overflow* 122

4 Média e variância, considerando saturação 124

5 Média e variância, considerando *overflow* 125

6 Média para os três modelos, considerando saturação 125

7 Variância para os três modelos, considerando saturação 126

8 Média para os três modelos, considerando *overflow* 127

9 Variância para os três modelos, considerando *overflow* 128

10 Média e variância, considerando saturação 133

11 Média e variância, considerando *overflow* 133

1 INTRODUÇÃO

Aparelhos eletrônicos estão cada vez mais populares no mundo moderno e a necessidade de adicionar mais funcionalidades nesses dispositivos faz necessário encontrar formas de torná-los mais eficientes em termos de consumo de bateria e de uso de *software* e *hardware*. Em termos de processamento de sinais, uma forma de diminuir custos de área em chip e consumo de energia é empregar aritmética de ponto fixo. Dessa forma, palavras de comprimento binário do tamanho mínimo necessário para uma dada aplicação podem ser usadas para representar sinais e variáveis, economizando os recursos disponíveis. Contudo, deve ser sempre mantido o compromisso entre o comprimento mínimo de palavra utilizado e o ruído de quantização gerado, de forma que a relação sinal-ruído não seja prejudicada.

A análise convencional dos efeitos de precisão finita é baseada na linearização do problema, em que a quantização das variáveis internas do sistema é modelada por um erro descrito como um ruído branco [1, 2]. Com isso, alguns efeitos das não-linearidades envolvidas acabam não sendo capturados, prejudicando a análise. Neste trabalho é proposto o uso de cadeias de Markov para analisar efeitos não-lineares em filtros digitais fixos e adaptativos.

Para filtros adaptativos, foi proposto em [3, 4] usar cadeias de Markov para avaliar o efeito da aritmética de ponto-fixo sobre o algoritmo LMS (Least-Mean-Squares) implementado com apenas um coeficiente w de B bits. Os 2^B valores possíveis de w foram definidos como os estados da cadeia. A matriz de transição

de estados foi obtida calculando, para cada estado i, a probabilidade de estar no estado i no instante n e atingir cada um de todos os estados no instante $n+1$. Com isso, puderam ser encontradas informações sobre a distribuição de probabilidade do coeficiente do filtro adaptativo.

Neste trabalho, uma abordagem semelhante à de [3, 4] é aplicada em filtros fixos IIR, definindo as saídas possíveis dos filtros como os estados da cadeia. Deseja-se calcular a matriz de transição de estados e retirar informações sobre a influência das não-linearidades no funcionamento dos filtros. Como, em geral, filtros de ordens elevadas são implementados como cascatas de filtros de primeira e de segunda ordem, o trabalho se concentra nessas estruturas, sendo obtidas suas matrizes de transição de estados. O método é aplicado em estruturas extremamente econômicas (em que o acumulador tem tamanho comparável ao da memória que armazena variáveis e sinais) e em estruturas em que o a precisão do acumulador é suficiente para que a quantização ocorra após todas as operações. A partir da matriz de transição de estados de cada filtro, a probabilidade de saturação da saída é calculada e é obtido o fator de escalamento da entrada de forma iterativa. Também é mostrado que a partir dos autovalores dessa matriz é possível definir implementações de filtros livres de ciclos-limite de entrada nula, o que é apresentado no capítulo 4.

Ainda no contexto de filtros fixos, são analisadas estruturas com filtros implementados em cascata (vide seção 4.7). Nesse caso, deseja-se verificar se o modelo de cadeia de Markov, obtido para um filtro de segunda ordem, pode ser adequadamente substituído por um modelo em que dois filtros de primeira ordem são colocados em cascata e analisados separadamente via cadeia de Markov. Para a comparação, são usados exemplos com filtros passa-tudo e passa-baixa.

Por fim, o estudo do algoritmo LMS unidimensional é retomado no capítulo 6, onde se assume a existência de correlação na entrada do filtro, que passa

a ser considerada na cadeia de Markov. O modelo é comparado à abordagem linearizada equivalente e com simulações de filtros implementados, demonstrando que o modelo com cadeias de Markov é mais preciso.

O trabalho que se segue está organizado da seguinte maneira. No capítulo 2 são tratados os conceitos relacionados ao cálculo de probabilidades, variáveis aleatórias, funções de probabilidade e cadeias de Markov, necessários nas manipulações realizadas para encontrar a matriz de transição de estados. No capítulo 3, filtros digitais e os efeitos de aritmética de ponto-fixo são apresentados, fornecendo o conhecimento básico para a aplicação da metodologia desenvolvida nos capítulo 4 e 5, em que filtros fixos são considerados. Nestes capítulos, o cálculo das funções de densidade de probabilidade usadas nas simulações é apresentado com detalhes e alguns exemplos são usados para mostrar o funcionamento dos modelos desenvolvidos. O capítulo 6 explicita os cálculos realizados para a obtenção do modelo para o algoritmo LMS e simulações são usadas para comparar o modelo obtido com o modelo linear. O capítulo 7 apresenta as conclusões deste trabalho e sugestões para possíveis estudos futuros.

2 PROBABILIDADES E CADEIAS DE MARKOV

O intuito deste trabalho é estudar o comportamento de filtros digitais implementados em precisão finita. Para esse fim, conhecendo a densidade de probabilidade de entrada de um filtro, deseja-se calcular a cadeia de Markov associada a um determinado parâmetro de estudo (que pode ser a saída em um filtro fixo – vide capítulos 4 e 5 – ou o coeficiente estimado por um filtro adaptativo – vide cap. 6). Conhecendo o ponto inicial da cadeia, é possível calcular a densidade de probabilidade do parâmetro para qualquer instante de tempo, o que torna essa abordagem muito atrativa.

Este capítulo introduz os elementos básicos da Teoria de Probabilidade e de cadeias de Markov, que serão as ferramentas básicas em todas as análises realizadas nos capítulos seguintes.

2.1 Propriedades básicas de probabilidades

Se $P(A)$ for a probabilidade de um evento A acontecer, pode-se dizer que

1. $P(A)$ é um número não-negativo e contido no intervalo $0 \leq P(A) \leq 1$.

2. A probabilidade do evento impossível é 0 e do evento certo é igual a 1.

3. Se dois eventos A e B não têm elementos em comum, a probabilidade $P(A \cup B)$, em que $A \cup B$ corresponde a A *união com* B, é dada por

$$P(A \cup B) = P(A) + P(B).$$

A última propriedade, vale lembrar, pode ser estendida para mais eventos, $P(A_1 \cup A_2 \cup \ldots \cup A_n) = P(A_1) + P(A_2) + \ldots + P(A_n)$, em que cada $P(A_k)$ (para $k = 1, 2, \ldots, n$) continua obedecendo às propriedades 1 e 2, com $A_i \cup A_j = \{0\}$, se $i \neq j$.

2.1.1 Independência de eventos

Dados dois eventos A e B, eles são ditos independentes se

$$P(A \cap B) = P(A)P(B), \tag{2.1}$$

em que $A \cap B$ corresponde à intersecção entre A e B.

2.2 Probabilidade condicionada

A probabilidade condicionada de um evento A, dado o evento M, pode ser interpretada como uma maneira de obter informação sobre a saída de um experimento a partir do conhecimento de uma informação parcial. Ela é descrita pela relação

$$P(A|M) = \frac{P(A \cap M)}{P(M)}, \tag{2.2}$$

em que $P(A|M)$ é a probabilidade do evento A, assumido o conhecimento de M, $A \cap M$ é a intersecção entre os eventos A e M e $P(M) > 0$. Essa é uma relação muito importante no escopo deste trabalho e será depois usada na definição de cadeias de Markov, na seção 2.11.

2.2.1 Teorema da probabilidade total

Seja S o conjunto de todos os eventos possíveis em um experimento, particionado em M_k eventos disjuntos (o que significa que $M_i \cap M_j = \emptyset$, para $i \neq j$).

Pode-se calcular a probabilidade de um evento A usando probabilidades condicionadas sobre os M_k eventos, usando

$$P(A) = P(A|M_1)P(M_1) + P(A|M_2)P(M_2) + \ldots + P(A|M_n)P(M_n). \tag{2.3}$$

Uma forma simples de compreender (2.3) é a seguinte: $P(A) = P(A \cap S)$, já que $A \subset S$. Mas, $S = M_1 \cup M_2 \cup \ldots \cup M_n$. Logo, $P(A)$ pode ser descrito com

$$\begin{aligned} P(A) = P(A \cap S) = P(A \cap (M_1 \cup M_2 \cup \ldots \cup M_n)) = \\ = P((A \cap M_1) \cup (A \cap M_2) \cup \ldots \cup (A \cap M_n)). \end{aligned} \tag{2.4}$$

Como M_k são eventos disjuntos, (2.4) pode ser reescrita como

$$P(A) = P(A \cap M_1) + P(A \cap M_2) + \ldots + P(A \cap M_n). \tag{2.5}$$

Usando em cada $P(A \cap M_k)$ a definição de probabilidade condicionada (2.2), obtém-se a equação da probabilidade total apresentada em (2.3). A figura 1 apresenta uma interpretação da teoria de conjuntos em que os eventos correspondem às partições do conjunto S e a intersecção entre A e os eventos M_k é mostrada através da sobreposição de A sobre os demais eventos. Para isso, é usado um diagrama de Venn [5].

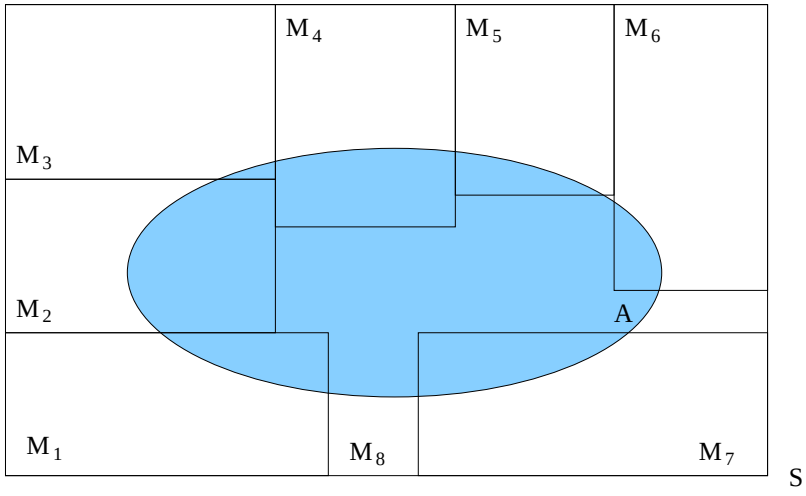

Figura 1: Diagrama de Venn representando a intersecção do evento A com os M_k eventos disjuntos de S (para $k = 1, \ldots, 8$)

2.2.2 Teorema de Bayes

Da definição de probabilidade condicionada (2.2), sabe-se que para calcular $P(M_k|A)$, basta usar

$$P(M_k|A) = \frac{P(M_k \cap A)}{P(A)}, \tag{2.6}$$

com $P(A) > 0$. Se, como na seção 2.2.1, o conjunto dos eventos S for particionado em M_k eventos disjuntos, é possível calcular $P(A)$ como em (2.3) e substituir a equação do teorema da probabilidade total em (2.6). Além disso, pode-se calcular $P(M_k \cap A)$ alternativamente como

$$P(M_k \cap A) = P(A|M_k)P(M_k), \tag{2.7}$$

e usar o resultado em (2.6), obtendo

$$P(M_k|A) = \frac{P(A|M_k)P(M_k)}{P(A|M_1)P(M_1) + \ldots + P(A|M_k)P(M_k) + \ldots + P(A|M_n)P(M_n)}. \tag{2.8}$$

Essa equação corresponde ao Teorema de Bayes, muito aplicado no cálculo de probabilidades condicionadas.

2.3 Variáveis aleatórias e distribuições de probabilidade

Dado um conjunto com as possíveis saídas de um experimento, uma variável aleatória (v.a.) é uma função que associa cada saída desse experimento a um número real [6]. Por exemplo, seja o conjunto de possíveis valores fornecidos por um dado $\beta = \{1,\ 2,\ 3,\ 4,\ 5,\ 6\}$ e $\boldsymbol{X}(i) = 10i$ uma função que associa a cada saída $i \in \beta$ o valor numérico $10i$. $\boldsymbol{X}(i)$ é uma variável aleatória obtida a partir das 6 possíveis saídas do experimento "lançamento de um dado" [5].

Uma variável aleatória é chamada discreta se o conjunto de valores que ela pode assumir for contável. Se esse conjunto for contável e finito, a variável será

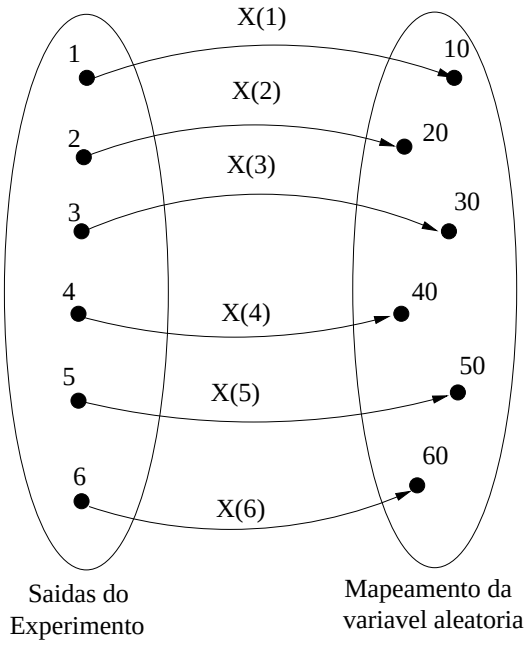

Figura 2: Mapeamento da variável aleatória $\boldsymbol{X}(i) = 10i$, em que i são os possíveis resultados do lançamento de um dado

discreta e finita, como apresentado no exemplo anterior. De fato, a definição de variável aleatória discreta e finita será a considerada nas próximas seções, para a análise de filtros utilizando cadeias de Markov.

A forma mais importante de se caracterizar variáveis aleatórias é através da probabilidade de ocorrência do resultado pertencer a uma semi-reta qualquer. Nesse caso, pode-se definir uma *função de probabilidade acumulada* associada a uma variável aleatória, que será uma característica particular dessa variável e estará relacionada ao seu mapeamento entrada-saída.

2.4 Função de probabilidade acumulada

Dado um número x e a variável aleatória $\boldsymbol{X}$, pode-se definir o evento $A_x = \{\boldsymbol{X} \leq x\}$. Esse evento depende de x e, portanto, sua probabilidade de ocorrência é uma função de x. A função de probabilidade associada a essa relação é chamada de *funçãode probabilidade acumulada* ou *de distribuição de probabilidade* e

é definida por

$$F_x(x) = P(\boldsymbol{X} \leq x), \tag{2.9}$$

para $-\infty < x < \infty$ e com $F_x(x)$ denotando a probabilidade acumulada da variável x. $F_x(x)$ apresenta as seguintes propriedades:

1. $F_x(x) \geq 0$;
2. $F_x(-\infty) = 0$;
3. $F_x(\infty) = 1$;
4. $F_x(x_1) \leq F_x(x_2)$, para $x_1 \leq x_2$;
5. $P(x_1 < x \leq x_2) = F_x(x_2) - F_x(x_1)$.

Para variáveis aleatórias discretas, $F_x(x)$ é caracterizada por patamares constantes que formam degraus discretos no gráfico da função. Para essa situação, $F_x(x)$ pode ser descrita como a soma de degraus,

$$F_x(x) = \sum_{k=1}^{n} P(\boldsymbol{X} = x_k) H(x - x_k), \tag{2.10}$$

em que a função degrau $H(k)$ correspode a

$$H(k) = \begin{cases} 1, \text{ se } k \geq 0 \\ 0, \text{ caso contrário.} \end{cases} \tag{2.11}$$

A seguir, um exemplo é apresentado com o intuito de esclarecer o conceito de probabilidade acumulada.

Exemplo: Considerando novamente a variável aleatória discreta $\boldsymbol{X}(i) = 10i$, pode-se calcular a função $F_x(x)$. Para $F_x(10)$, obtém-se

$$F_x(10) = P(\boldsymbol{X} \leq 10) = P(\boldsymbol{X} = \{10\}) = \frac{1}{6},$$

já que apenas $x = 10$ satisfaz $\boldsymbol{X} \leq 10$, considerando as 6 possibilidades existentes. De forma semelhante,

$$F_x(20) = P(\boldsymbol{X} \leq 20) = P(\boldsymbol{X} = \{10,\ 20\}) = \tfrac{2}{6}$$
$$F_x(30) = P(\boldsymbol{X} \leq 30) = P(\boldsymbol{X} = \{10,\ 20,\ 30\}) = \tfrac{3}{6}$$
$$F_x(40) = P(\boldsymbol{X} \leq 40) = P(\boldsymbol{X} = \{10,\ 20,\ 30,\ 40\}) = \tfrac{4}{6}$$
$$F_x(50) = P(\boldsymbol{X} \leq 50) = P(\boldsymbol{X} = \{10,\ 20,\ 30,\ 40,\ 50\}) = \tfrac{5}{6}$$
$$F_x(60) = P(\boldsymbol{X} \leq 60) = P(\boldsymbol{X} = \{10,\ 20,\ 30,\ 40,\ 50,\ 60\}) = 1.$$

Das propriedades de $F_x(x)$, sabe-se que $F_x(x < 10) = 0$ e $F_x(x > 60) = 1$. Com isso, o gráfico de $F_x(x)$ fica como apresentado na figura 3. Essa figura

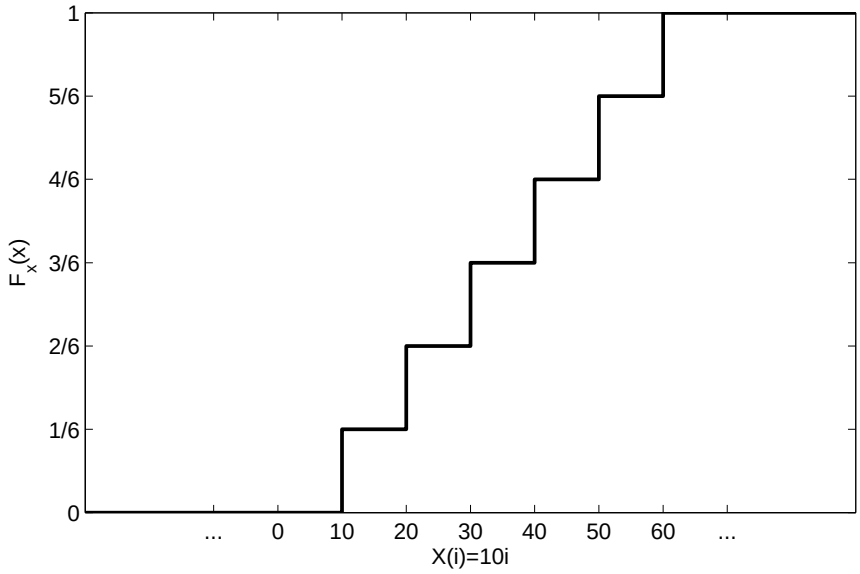

Figura 3: Probabilidade acumulada da variável aleatória $\boldsymbol{X}$

equivale à distribuição de probabilidade

$$F_x(x) = \frac{1}{6}\sum_{i=1}^{6} H(x - 10i).$$

A partir da probabilidade acumulada, uma outra função também pode ser definida: a *função densidade de probabilidade*.

2.5 Função de densidade de probabilidade

A *função de densidade de probabilidade* (f.d.p.) é definida como a derivada de $F_x(x)$ em relação a x [5], isto é,

$$f_x(x) = \frac{dF_x(x)}{dx}, \tag{2.12}$$

sendo que esta abordagem é válida para variáveis contínuas e discretas.

Assim como $F_x(x)$, a função de densidade de probabilidade apresenta algumas características relevantes:

1. $f_x(x) \geq 0$, o que vem do fato de $F_x(x)$ ser uma função monotônica crescente;
2. $\int_{-\infty}^{\infty} f_x(x)dx = 1$;
3. $F_x(x) = \int_{-\infty}^{x} f_x(x)dx$;
4. $P(x_1 < \boldsymbol{X} \leq x_2) = \int_{x_1}^{x_2} f_x(x)dx$.

Se $\boldsymbol{X}$ for uma variável aleatória discreta, com $F_x(x)$ como apresentado em (2.10), $f_x(x)$ corresponderá a

$$f_x(x) = \sum_{k=1}^{n} P(\boldsymbol{X} = x_k)\delta(x - x_k), \tag{2.13}$$

em que $\delta(x)$ é o *delta de Dirac*.

Para essa situação, o gráfico da função corresponde a impulsos de área $P(\boldsymbol{X} = x_k)$ nos pontos em que $\boldsymbol{X} = x_k$.

Exemplo: Considerando novamente o experimento "lançamento de um dado honesto", como definido no exemplo anterior, deseja-se encontrar a função de densidade de probabilidade de $X(i) = 10i$. A partir da equação (2.4) de $F_x(x)$, obtém-se

$$f_x(x) = \frac{1}{6}\sum_{i=1}^{6} \delta(x - 10i), \tag{2.14}$$

de onde se consegue a figura 4.

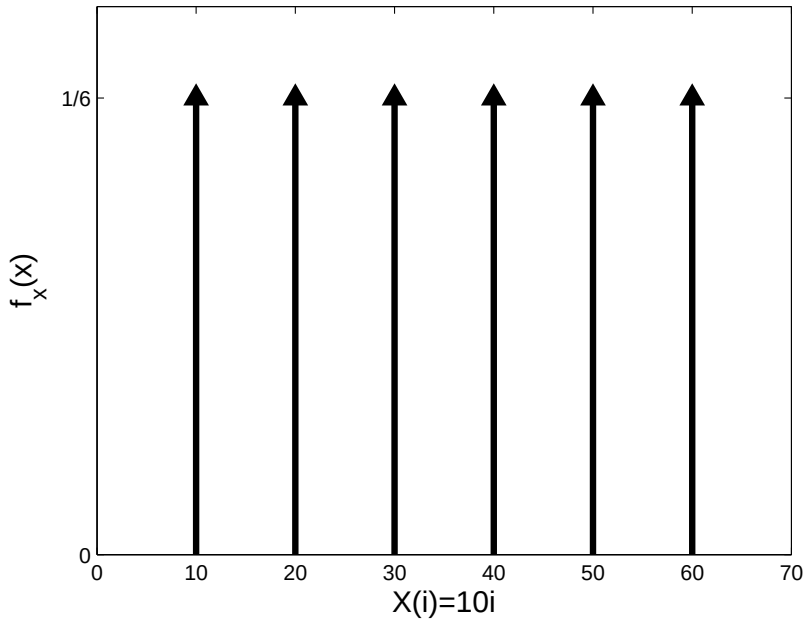

Figura 4: Densidade de probabilidade da variável aleatória $\boldsymbol{X}$

2.6 Função densidade de probabilidade da soma de variáveis aleatórias independentes

Se duas variáveis aleatórias independentes $\boldsymbol{X}$ e $\boldsymbol{Y}$ com f.d.p. $f_x(x)$ e $f_y(y)$ forem somadas, a densidade de probabilidade $f_z(z)$ de $\boldsymbol{Z} = \boldsymbol{X} + \boldsymbol{Y}$ pode ser calculada pela integral de convolução

$$f_z(z) = \int_{-\infty}^{\infty} f_x(\alpha) f_y(z - \alpha) d\alpha. \tag{2.15}$$

Essa é uma importante relação que será depois usada para calcular a f.d.p. da saída de um filtro digital, em função de sua entrada.

2.7 Funções de probabilidade consideradas no escopo deste texto

Neste tópico, são apresentadas algumas funções de probabilidade usadas ao longo do texto . Ao final da seção, é apresentada a forma geral de obtenção da f.d.p. da entrada dos filtros usados nos exemplos.

2.7.1 Variável aleatória uniforme

Uma variável aleatória $\boldsymbol{X}$ contínua é dita uniforme se sua função densidade de probabilidade é tal que

$$f_x(x) = \begin{cases} \dfrac{1}{x_2 - x_1}, \text{ se } x_1 \leq x \leq x_2 \\ 0, \text{ caso contrário.} \end{cases} \tag{2.16}$$

Para essa situação, a $F_x(x)$ correspondente é uma rampa dada por

$$F_x(x) = \begin{cases} 0, \text{ se } x < x_1 \\ \dfrac{x - x_1}{x_2 - x_1}, \text{ se } x_1 \leq x \leq x_2 \\ 1, \text{ se } x > x_2. \end{cases} \tag{2.17}$$

Também é possível definir variáveis aleatórias discretas uniformes. Se uma variável aleatória $\boldsymbol{Y}$ for discreta e uniforme, então suas funções de probabilidades serão calculadas por

$$f_y(y) = \frac{1}{N} \sum_{i=1}^{N} \delta(y - y_i) \tag{2.18}$$

e

$$F_y(y) = \frac{1}{N} \sum_{i=1}^{N} H(y - y_i), \tag{2.19}$$

e $F_y(y)$ será composta por degraus.

2.7.2 Variável aleatória normal ou gaussiana

Uma variável aleatória $\boldsymbol{X}$ é dita normal ou gaussiana se sua função de densidade de probabilidade for calculada por

$$f_x(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)/2\sigma^2}, \tag{2.20}$$

em que μ e σ^2 são números reais e correspondem à média e à variância de $\boldsymbol{X}$ (vide Seção 2.10).

A distribuição de probabilidades, nesse caso, é dada por

$$F_x(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-(x-\mu)^2/2\sigma^2}. \tag{2.21}$$

De fato, a integral de $F_x(x)$ não apresenta solução fechada e somente pode ser calculada através de métodos numéricos. Nesse caso, usam-se os valores tabelados para a função $G(x)$ [5],

$$G(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\epsilon^2/2} d\epsilon, \tag{2.22}$$

com a propriedade

$$G(-x) = 1 - G(x), \tag{2.23}$$

para encontrar os valores de $F_x(x)$. A relação entre $F_x(x)$ e $G(x)$ é tal que

$$F_x(x) = G\left(\frac{x-\mu}{\sigma}\right). \tag{2.24}$$

Portanto, para calcular $P(x_1 \leq \boldsymbol{X} \leq x_2)$, basta fazer

$$P(x_1 \leq \boldsymbol{X} \leq x_2) = F_x(x_2) - F_x(x_1) = G\left(\frac{x_2-\mu}{\sigma}\right) - G\left(\frac{x_1-\mu}{\sigma}\right) \tag{2.25}$$

e consultar uma tabela.

2.7.3 Cálculo da probabilidade de entrada do filtro digital

Ao longo deste trabalho, assume-se que a entrada digital do filtro $x_d(n)$ é uma variável aleatória discreta, obtida após a amostragem de $x_c(t)$, que corresponde a uma variável aleatória contínua. Portanto, é necessário obter a densidade de probabilidade de $x_d(n)$ a partir da densidade de $x_c(t)$, definindo alguma forma de correspondência entre as duas variáveis aleatórias. Como exemplo, suponha que $x_c(t)$ possui distribuição uniforme entre -1 e 1 e que após a amostragem, somente são possíveis os valores $x_d(n) = -1$, -0.5, 0 e 0.5. Nesse caso, para cada valor possível de $x_d(n)$, deve ser definido um pequeno intervalo de valores de $x_c(t)$ correspondente à conversão A/D (analógico-digital). Se for definido

$$x_d(n) = \begin{cases} -1 \text{ se } -1 \leq x_c(t) < -0.75 \\ -0.5 \text{ se } -0.75 \leq x_c(t) < -0.25 \\ 0 \text{ se } -0.25 \leq x_c(t) < 0.25 \\ 0.5 \text{ se } 0.25 \leq x_c(t) \leq 1, \end{cases}$$

a probabilidade associada a cada intervalo fornecerá a probabilidade de cada $x_d(n)$ correspondente, isto é,

$$\begin{cases} P(-1 \leq x_c(t) < -0.75) = P(x_d(n) = -1) = 0.125 \\ P(-0.75 \leq x_c(t) < -0.25) = P(x_d(n) = -0.5) = 0.25 \\ P(-0.25 \leq x_c(t) < 0.25) = P(x_d(n) = 0) = 0.25 \\ P(0.25 \leq x_c(t) \leq 1) = P(x_d(n) = 0.5) = 0.375, \end{cases}$$

definindo a densidade de probabilidade de $x_d(n)$, apresentada na figura 5.

Essa forma de obter a probabilidade de entrada pode ser usada para definir a densidade de probabilidade discreta de $x_d(n)$ a partir da densidade contínua de $x_c(t)$. Nas simulações realizadas durante este trabalho, as probabilidades associadas às funções apresentadas nas seções 2.7.1 e 2.7.2 foram obtidas dessa forma.

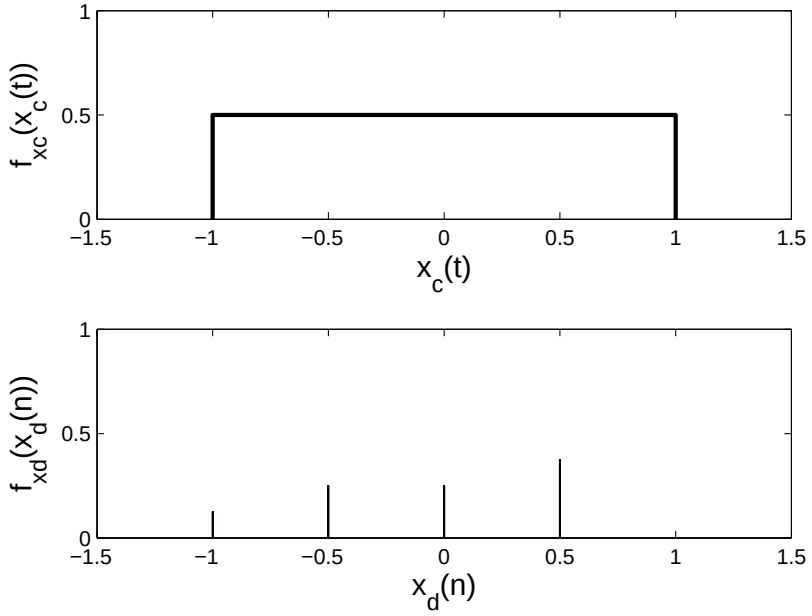

Figura 5: Densidade de probabilidade da entrada antes (acima) e depois (abaixo) da amostragem

2.8 Funções de probabilidade com duas variáveis aleatórias

Em alguns experimentos, mais de uma variável aleatória pode estar ligada ao resultado que se deseja observar. Nesse caso, as funções de distribuição e de densidade de probabilidade podem ser definidas em termos dessas múltiplas variáveis. Se for considerada a situação em que duas variáveis aleatórias $\boldsymbol{X}$ e $\boldsymbol{Y}$ estão associadas a um experimento, a distribuição de probabilidade conjunta é definida como

$$F_{xy}(x, y) = P(\boldsymbol{X} \leq x,\ \boldsymbol{Y} \leq y), \tag{2.26}$$

com a função de densidade de probabilidade conjunta $f_{xy}(x, y)$ dada por

$$f_{xy}(x, y) = \frac{\partial^2 F_{xy}(x, y)}{\partial x \partial y}, \tag{2.27}$$

ou ainda

$$F_{xy}(x, y) = \int_{-\infty}^{x} \int_{-\infty}^{y} f_{xy}(\alpha, \beta)\ d\alpha d\beta. \tag{2.28}$$

Vale lembrar que $F_{xy}(x, y)$ e $f_{xy}(x, y)$, por caracterizarem funções de probabilidade, devem sempre respeitar as propriedades

1. $f_{xy}(x, y) \geq 0$
2. $0 \leq F_{xy}(x, y) \leq 1$
3. $F_{xy}(\infty, \infty) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{xy}(\alpha, \beta)\, d\alpha d\beta = 1$
4. $F_{xy}(-\infty, -\infty) = 0$,

de forma que as probabilidades podem ser calculadas como

$$\begin{aligned} P(\boldsymbol{X} < x,\ y_1 < \boldsymbol{Y} \leq y_2) &= F_{xy}(x, y_2) - F_{xy}(x, y_1), \\ P(x_1 < \boldsymbol{X} \leq x_2,\ \boldsymbol{Y} < y) &= F_{xy}(x_2, y) - F_{xy}(x_1, y) \ \text{ e} \\ P(x_1 < \boldsymbol{X} \leq x_2,\ y_1 < \boldsymbol{Y} \leq y_2) &= F_{xy}(x_2, y_2) - F_{xy}(x_1, y_2) \\ &\quad - F_{xy}(x_2, y_1) + F_{xy}(x_1, y_1). \end{aligned}$$

A partir das funções de distribuição e de densidade de probabilidade conjunta, é possível definir as distribuições marginais das variáveis aleatórias, como

1. Distribuição marginal de $\boldsymbol{X}$: $F_x(x) = F_{xy}(x, \infty)$
2. Distribuição marginal de $\boldsymbol{Y}$: $F_y(x) = F_{xy}(\infty, y)$
3. Densidade marginal de $\boldsymbol{X}$: $f_x(x) = \int_{-\infty}^{\infty} f_{xy}(x, y)\, dy$
4. Densidade marginal de $\boldsymbol{Y}$: $f_y(y) = \int_{-\infty}^{\infty} f_{xy}(x, y)\, dx$,

que permitem observar as probabilidades associadas a cada variável separadamente.

Os conceitos de distribuição e de densidade de probabilidade conjunta podem ser aplicados em funções de mais variáveis. Dessa forma, as propriedades anteriores podem ser estendidas para funções multi-variáveis.

2.9 Funções de probabilidade condicionada

Semelhante à probabilidade condicionada definida em (2.2), pode-se definir a distribuição de probabilidade de $\boldsymbol{X}$, dado um evento M, isto é,

$$F_x(x|M) = P(\boldsymbol{X} \leq x|M) = \frac{F(\boldsymbol{X} \leq x,\ M)}{P(M)}, \tag{2.29}$$

assumindo $P(M) > 0$ e definindo $(\boldsymbol{X} \leq x,\ M)$ como a intersecção entre os eventos $\boldsymbol{X} \leq x$ e M. Nesse caso, a função de densidade de probabilidade é descrita como

$$f_x(x|M) = \frac{dF_x(x|M)}{dx}. \tag{2.30}$$

A probabilidade $P(x_1 < \boldsymbol{X} \leq x_2|M)$ pode ser calculada por meio de

$$P(x_1 < \boldsymbol{X} \leq x_2|M) = F_x(x_2|M) - F_x(x_1|M) = \int_{x_1}^{x_2} f_x(x|M)\ dx, \tag{2.31}$$

sendo que $F_x(x|M)$ e $f_x(x|M)$ possuem as seguintes características

1. $F_x(-\infty|M) = 0$
2. $F_x(\infty|M) = 1$
3. $0 \leq F_x(\infty|M) \leq 1$.
4. $f_x(x|M) \geq 0$
5. $\int_{-\infty}^{\infty} f_x(x|M)\ dx = 1$.

A definição de distribuição e de densidade de probabilidade para uma variável aleatória será muito importante ao longo deste texto, devido à sua importância para o entendimento de Cadeias de Markov. As propriedades aqui apresentadas são válidas tanto para variáveis contínuas quanto para variáveis discretas.

2.10 Definição de esperança matemática e variância

2.10.1 Esperança matemática

A esperança ou média de uma variável aleatória $\boldsymbol{X}$ é definida como a integral

$$\mu_x = \int_{-\infty}^{\infty} x f_x(x)\, dx, \tag{2.32}$$

em que μ_x corresponde à média ou esperança de $\boldsymbol{X}$. Também é comum encontrar as notações $\mathrm{E}\{\boldsymbol{X}\}$ e $\bar{\boldsymbol{X}}$ para a denominação da esperança matemática.

Para uma variável aleatória discreta, a esperança pode ser simplificada com

$$\mu_x = \sum_{i=1}^{N} x_i P(\boldsymbol{X} = x_i). \tag{2.33}$$

2.10.2 Variância

A variância de $\boldsymbol{X}$ é definida como o valor

$$\sigma_x^2 = \int_{-\infty}^{\infty} (x - \mu_x)^2 f_x(x)\, dx, \tag{2.34}$$

que pode ser calculado para variáveis discretas por

$$\sigma_x^2 = \sum_{i=1}^{N} (\mu_x - x_i)^2 P(\boldsymbol{X} = x_i). \tag{2.35}$$

A notação var$[\boldsymbol{X}]$ também costuma ser empregada para definir a variância de $\boldsymbol{X}$ (vide [6], por exemplo).

2.11 Cadeias de Markov de tempo discreto

Uma cadeia de Markov de tempo discreto é uma sequência contável de variáveis aleatórias $\{\boldsymbol{X}_k\}_{k=0}^{\infty}$ com valores associados a um conjunto discreto e finito $I = \{1,\ 2, \ldots,\ N\}$, denominado *espaço de estados* [7]. A principal característica de

uma cadeia de Markov é o fato de não reter memória de onde esteve no passado. Isso significa que apenas o estado corrente pode influenciar para onde o processo irá no instante seguinte, o que é conhecido como propriedade de Markov [8],

$$P(\boldsymbol{X}_n = i_n | \boldsymbol{X}_{n-1} = i_{n-1}, \cdots, \boldsymbol{X}_0 = i_0) = P(\boldsymbol{X}_n = i_n | \boldsymbol{X}_{n-1} = i_{n-1}), \quad (2.36)$$

em que $i_n \in I$ é o estado da cadeia, no instante n. Quando (2.36) é independente do instante n, a cadeia ainda é dita *homogênea* [7], o que será assumido para todas as cadeias apresentadas neste texto a partir deste ponto.

Em geral, a equação (2.36) é substituída pela notação

$$p_{ij} = P(\boldsymbol{X}_n = i | \boldsymbol{X}_{n-1} = j), \quad (2.37)$$

em que cada $p_{ij} \geq 0$ representa a *probabilidade de transição de estado* do estado j para o estado i. Com esses p_{ij}, e lembrando que o conjunto de estados é limitado a N elementos, define-se a *matriz de transição de estados* da cadeia de Markov $\mathbb{P}$ [6, 7, 8] , em que cada elemento p_{ij} ocupa a posição correspondente à linha i e à coluna j, em uma matriz de dimensões $N \times N$.

Pela própria maneira como a matriz de transição de estados é definida, os elementos de cada uma de suas colunas compõem uma função de densidade de probabilidade condicionada ao evento $\boldsymbol{X}_{n-1} = j$. Portanto,

$$\sum_{i=1}^{N} p_{ij} = \sum_{i=1}^{N} P(\boldsymbol{X}_n = i | \boldsymbol{X}_{n-1} = j) = 1,$$

para cada coluna.

Exemplo: Imagine uma cadeia de Markov com três estados possíveis, ou

seja, $I = \{1,\ 2,\ 3\}$, com a seguinte matriz $\mathbb{P}$

$$\mathbb{P} = \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \left[\begin{array}{ccc} 0.2 & 0.5 & 0.7 \\ 0.5 & 0.3 & 0.1 \\ 0.3 & 0.2 & 0.2 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ 1 \\ 2 \\ 3 \end{array},$$

em que os estados são colocados acima e à direita da matriz para facilitar a identificação dos p_{ij}. O elemento $p_{21} = 0.5$, por exemplo, corresponde à probabilidade de começar no estado 1 e atingir o estado 2 no instante seguinte, que é uma probabilidade maior do que a de atingir o estado 3 a partir do estado 1 ($p_{31} = 0.3$). Ainda deve ser notado que a soma $p_{11} + p_{21} + p_{31} = 1$, o que também é válido para as outras colunas de $\mathbb{P}$. A figura 6 mostra o *diagrama de transição de estados* para $\mathbb{P}$, muito usado para facilitar a visualização das transições.

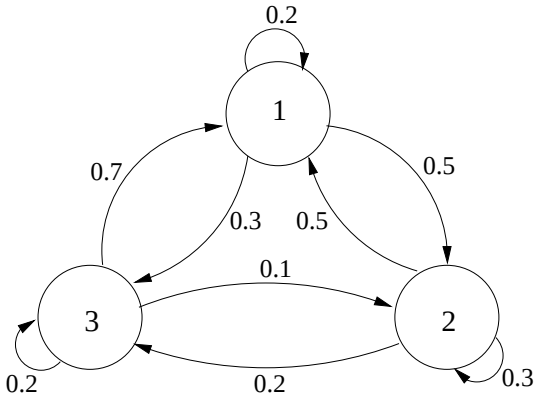

Figura 6: Diagrama de transição de estados da matriz $\mathbb{P}$ usada no exemplo

2.11.1 Cálculo da probabilidade depois de n passos

Em muitas situações é interessante calcular a probabilidade de transição em uma cadeia de Markov depois de n passos. A maneira de encontrar essas probabilidades é usar a equação de Chapman-Kolmogorov [6] para encontrar $p_{ij}^{(n)}$

recursivamente, isto é,

$$P(\boldsymbol{X}_n = i|\boldsymbol{X}_0 = j) = p_{ij}^{(n)} = \sum_{k=1}^{N} p_{kj}^{(n-1)} p_{ik}. \tag{2.38}$$

Para verificar (2.38), basta expandir $P(\boldsymbol{X}_n = i|\boldsymbol{X}_0 = j)$ por meio do teorema da probabilidade total (2.3),

$$\begin{aligned} P(\boldsymbol{X}_n = i|\boldsymbol{X}_0 = j) &= \textstyle\sum_{k=1}^{N} P(\boldsymbol{X}_{n-1} = k|\boldsymbol{X}_0 = j) P(\boldsymbol{X}_n = i|\boldsymbol{X}_{n-1} = k,\ \boldsymbol{X}_0 = j) \\ &= \textstyle\sum_{k=1}^{N} p_{kj}^{(n-1)} p_{ik}, \end{aligned}$$

em que a propriedade de Markov (2.36) é usada para simplificar

$$P(\boldsymbol{X}_n = i|\boldsymbol{X}_{n-1} = k,\ \boldsymbol{X}_0 = j) = P(\boldsymbol{X}_n = i|\boldsymbol{X}_{n-1} = k).$$

De fato, a equação (2.38) fornece uma forma iterativa de obter as probabilidades de transição após n passos, que pode ser aplicada em $\mathbb{P}$ para obter iterativamente $\mathbb{P}^n$, ou seja,

$$\begin{aligned} &\mathbb{P}^{(0)} = \boldsymbol{I}_{N\times N} \\ &\mathbb{P}^{(1)} = \mathbb{P} \\ &\mathbb{P}^{(2)} = \mathbb{P}^{(1)}\mathbb{P}^{(1)} = \mathbb{P}\mathbb{P} = \mathbb{P}^2 \\ &\mathbb{P}^{(3)} = \mathbb{P}^{(1)}\mathbb{P}^{(2)} = \mathbb{P}^{(2)}\mathbb{P}^{(1)} = \mathbb{P}^2\mathbb{P} = \mathbb{P}^3 \\ &\vdots \\ &\mathbb{P}^{(n)} = \mathbb{P}^{(1)}\mathbb{P}^{(n-1)} = \mathbb{P}^{(n-1)}\mathbb{P}^{(1)} = \mathbb{P}^{n-1}\mathbb{P} = \mathbb{P}^n. \end{aligned} \tag{2.39}$$

Portanto, para encontrar o elemento p_{ij}^n, basta procurar pelo elemento p_{ij} da matriz $\mathbb{P}^n$, calculada por (2.39).

2.11.1.1 Classificação dos estados e definição de classes

Um estado i é dito *acessível* ao estado j se a probabilidade p_{ij}^n de atingir i partindo do estado j, para algum instante n, for maior do que zero. Se $A(i)$ for o conjunto de todos os estados acessíveis ao estado i, i é chamado *recorrente* se

para todo estado j acessível a partir de i, i também for acessível a partir j. Ou seja, para todo j pertencente a $A(i)$, i deve pertencer a $A(j)$. Portanto, quando se começa em um estado recorrente i, só é possível visitar os estados j tais que $i \in A(j)$ [6]. De forma complementar ao conceito de estado recorrente, existe o estado *transiente*. Define-se como estado transiente qualquer estado que seja não-recorrente, o que significa que não será mais visitado após um número finito de passos.

O diagrama de estados da figura 7 apresenta um exemplo de cadeia com estados transientes e recorrentes. Na figura, os estados 3 e 4 são transientes, já

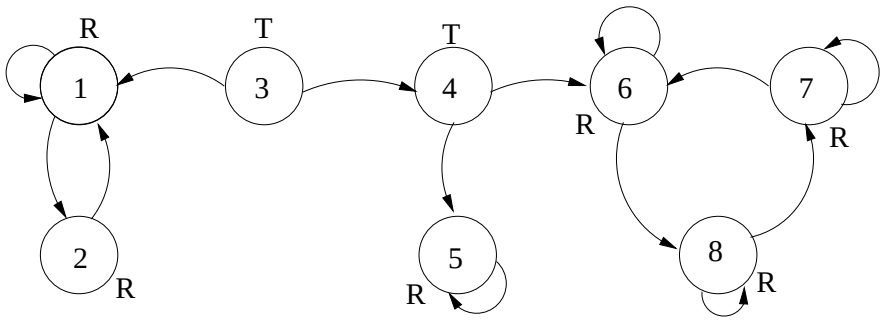

Figura 7: Diagrama de transição de estados mostrando estados recorrentes (indicados pela letra R) e transientes (indicados pela letra T).

que 3 não é acessível a nenhum estado, enquanto 4 apenas é visitado por 3, mas não é acessado por nenhum estado recorrente. Os demais estados são recorrentes e formam as chamadas *classes de recorrência*.

Uma classe de recorrência é um conjunto de estado em que os elementos são acessíveis entre si, mas sem acesso aos estados fora desse conjunto. Na figura 7, os conjuntos de estados $\{1,\ 2\}$, $\{5\}$ e $\{6,\ 7,\ 8\}$ formam classes de recorrência. Se, por exemplo, a cadeia atinge o estado 6, as próximas transições ficam limitadas aos estados 6, 7 e 8 e nenhum outro estados fora da classe pode ser atingido.

Usando os conceitos de classe de recorrência e de estados transientes, pode-se sempre decompor uma cadeia de Markov em termos desses elementos. Dessa forma, para que exista um estado transiente, deverá existir pelo menos uma classe

de recorrência, embora existam cadeias em que todos os estados pertençam à uma única classe de recorrência, sem estados transientes (vide figuras 8 e 9). Portanto, se a cadeia começar em um estado transiente, ela evoluirá até encontrar uma classe recorrente. Caso ela seja iniciada em uma classe recorrente, ela permanecerá sempre no conjunto de estados acessíveis a essa classe.

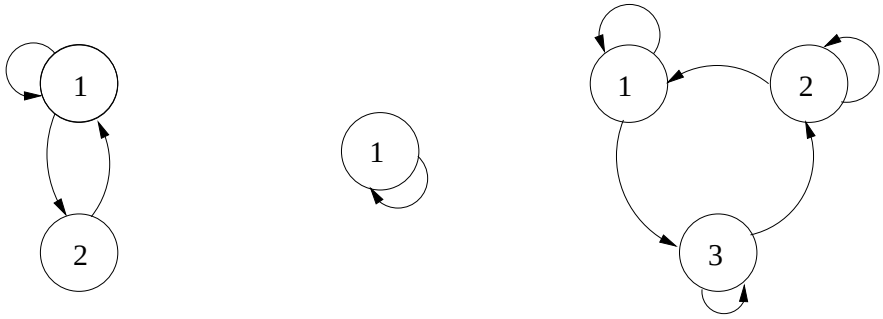

Figura 8: Cadeias de Markov composta somente por classes recorrentes

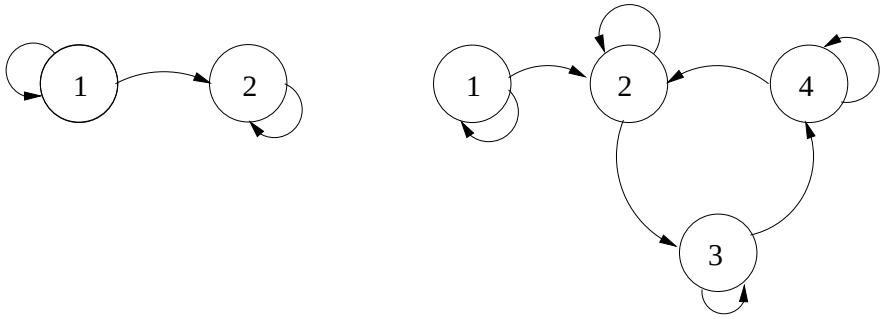

Figura 9: Cadeias de Markov com classes recorrentes e estados transientes

As classes recorrentes de uma cadeia alternam seus estados dentro de um conjunto definido de possibilidades. Define-se um ciclo da classe recorrente como sendo o número de passos entre duas visitas a um mesmo estado i, para uma dada sequência possível de transição de estados. A periodicidade de uma classe corresponde ao máximo divisor comum entre todos os ciclos possíveis dentro de uma classe de recorrência [7]. Considera-se uma classe aperiódica quando seu período é igual a 1. Na figura 10, por exemplo, escolhendo qualquer estado da classe final $\{1,\ 2,\ 3\}$, é necessário um mínimo de 3 passos para atingir o mesmo estado novamente, enquanto que para a classe $\{5,\ 6\}$, um mínimo de dois

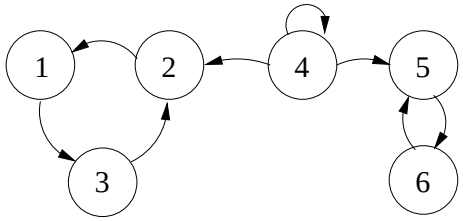

Figura 10: Classes recorrentes de periodicidade 3 (classe $\{1,\ 2,\ 3\}$) e 2 (classe $\{5,\ 6\}$)

passos é necessário, o que define a periodicidade dessas classes como sendo 3 e 2, respectivamente.

2.11.1.2 Comportamento em regime estacionário

A partir de uma distribuição de probabilidade inicial dos estados $\boldsymbol{\pi}(0)$ (que também pode ser chamada de *condição inicial* da cadeia), de dimensão $N \times 1$, e uma matriz de transição de estados $\mathbb{P}$, de dimensões $N \times N$, pode-se obter a distribuição de probabilidade dos estados no instante seguinte, calculando

$$\boldsymbol{\pi}(1) = \mathbb{P}\boldsymbol{\pi}(0). \tag{2.40}$$

Da mesma forma, pode-se calcular a distribuição do próximo instante, isto é,

$$\boldsymbol{\pi}(2) = \mathbb{P}\boldsymbol{\pi}(1) = \mathbb{P}\mathbb{P}\boldsymbol{\pi}(0) = \mathbb{P}^2\boldsymbol{\pi}(0).$$

Aplicando a ideia de (2.40) sucessivas vezes, a distribuição $\boldsymbol{\pi}(n)$ pode ser calculada como

$$\boldsymbol{\pi}(n) = \mathbb{P}\boldsymbol{\pi}(n-1) = \mathbb{P}^n\boldsymbol{\pi}(0), \tag{2.41}$$

permitindo obter a distribuição de probabilidade depois de n passos, apenas com o conhecimento de $\mathbb{P}$ e da condição inicial $\boldsymbol{\pi}(0)$, para n tão grande quanto o desejado.

Pelo Teorema de Perron-Frobenius [9], sabe-se que $\mathbb{P}$ possui pelo menos um autovalor igual a 1 e que esse autovalor é o seu maior autovalor. Usando um vetor

$\boldsymbol{v} = [1\ 1\ \ldots 1]$ de dimensões $1 \times N$, nota-se facilmente que 1 é um autovalor de $\mathbb{P}$, já que

$$\boldsymbol{v}\mathbb{P} = \left[\begin{array}{ccc} \sum_{i=1}^{N} p_{i1} & \cdots & \sum_{i=1}^{N} p_{iN} \end{array} \right] = 1\boldsymbol{v}, \tag{2.42}$$

onde $\boldsymbol{v}$ é um autovetor à esquerda de $\mathbb{P}$. Como os autovalores à esquerda e à direita são iguais, também deve existir um autovetor à direita de $\mathbb{P}$ correspondente ao autovalor 1 [9]. Isso significa que $\mathbb{P}$ sempre tem ao menos um autovalor igual a 1.

Se for possível encontrar um vetor coluna $\boldsymbol{\pi}$ em que cada elemento $\pi_i \geq 0$ e $\sum_{i=1}^{N} \pi_i = 1$, tal que

$$\boldsymbol{\pi} = \mathbb{P}\boldsymbol{\pi}, \tag{2.43}$$

a aplicação de (2.41) fornecerá

$$\boldsymbol{\pi} = \mathbb{P}^n\boldsymbol{\pi}, \tag{2.44}$$

e $\boldsymbol{\pi}$, além de ser um autovetor de $\mathbb{P}$, será denominada uma *distribuição estacionária ou invariante* da matriz $\mathbb{P}$ [7], já que será independente de n.

Convergência para o equilíbrio

Se uma cadeia com matriz de transição $\mathbb{P}$ possuir apenas uma classe de recorrência e for aperiódica, com $\boldsymbol{\pi}^T = [\pi_1\ \ldots\ \pi_N]$ uma distribuição de probabilidade invariante, então [6, 8]

1. $p_{ij}^{(n)} \to \pi_i$, quando $n \to \infty$, $\forall i, j$
2. $\pi_i = \sum_{j=1}^{N} \pi_j p_{ij}^{(n)}$
3. $\sum_{j=1}^{N} \pi_j = 1$
4. $\pi_i = 0$, para todo estado transiente i
5. $\pi_i > 0$, para todo estado recorrente i.

Quando se observa a matriz de transição de estados em regime estacionário

$\mathbb{P}^{\infty}$, isto é,

$$\mathbb{P}^{\infty} = \lim_{n\to\infty} \mathbb{P}^{n}, \tag{2.45}$$

essas propriedades definem uma matriz de probabilidade em que todas as colunas são iguais, de forma que a probabilidade de atingir um estado i no próximo passo independe do estado inicial.

Exemplo 1: Seja a matriz $\mathbb{P}$

$$\mathbb{P} = \begin{bmatrix} 0.2 & 0.7 \\ 0.8 & 0.3 \end{bmatrix},$$

correspondente à cadeia da figura 11.

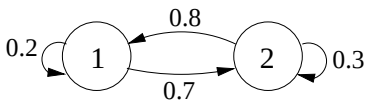

Figura 11: Cadeia de Markov formada por uma classe única de recorrência aperiódica

Do diagrama de estados de $\mathbb{P}$, nota-se que existe apenas uma classe de recorrência $\{1,\ 2\}$ e que ela é aperiódica. Logo, pelo exposto anteriormente, $\mathbb{P}^{\infty}$ deve ter as colunas iguais e a distribuição do estado estacionário deve ser independente da condição inicial $\boldsymbol{\pi}(0)$. Pode-se calcular a distribuição em regime por meio da equação matricial (2.44) para valores diferentes de condição inicial:

1. Condição inicial $\boldsymbol{\pi}(0) = [1\ 0]^T$:

$$\boldsymbol{\pi}(1) = \mathbb{P}\boldsymbol{\pi}(0) = [0.2\ 0.8]^T$$
$$\boldsymbol{\pi}(2) = \mathbb{P}\boldsymbol{\pi}(1) = [0.6\ 0.4]^T$$
$$\boldsymbol{\pi}(3) = \mathbb{P}\boldsymbol{\pi}(2) = [0.4\ 0.6]^T$$
$$\vdots$$
$$\boldsymbol{\pi}(15) = \mathbb{P}\boldsymbol{\pi}(14) = [0.4667\ 0.5333]^T$$
$$\vdots$$
$$\lim_{n\to\infty} \boldsymbol{\pi}(n) = [0.4667\ 0.5333]^T.$$

2. Condição inicial $\boldsymbol{\pi}(0) = [0\ 1]^T$:

$$\begin{aligned}
&\boldsymbol{\pi}(1) = \mathbb{P}\boldsymbol{\pi}(0) = [0.7\ 0.3]^T \\
&\boldsymbol{\pi}(2) = \mathbb{P}\boldsymbol{\pi}(1) = [0.35\ 0.65]^T \\
&\boldsymbol{\pi}(3) = \mathbb{P}\boldsymbol{\pi}(2) = [0.525\ 0.475]^T \\
&\vdots \\
&\boldsymbol{\pi}(15) = \mathbb{P}\boldsymbol{\pi}(14) = [0.4667\ 0.5333]^T \\
&\vdots \\
&\lim_{n\to\infty} \boldsymbol{\pi}(n) = [0.4667\ 0.5333]^T.
\end{aligned}$$

3. Condição inicial $\boldsymbol{\pi}(0) = [0.7\ 0.3]^T$:

$$\begin{aligned}
&\boldsymbol{\pi}(1) = \mathbb{P}\boldsymbol{\pi}(0) = [0.35\ 0.65]^T \\
&\boldsymbol{\pi}(2) = \mathbb{P}\boldsymbol{\pi}(1) = [0.525\ 0.475]^T \\
&\boldsymbol{\pi}(3) = \mathbb{P}\boldsymbol{\pi}(2) = [0.4375\ 0.5625]^T \\
&\vdots \\
&\boldsymbol{\pi}(15) = \mathbb{P}\boldsymbol{\pi}(14) = [0.4667\ 0.5333]^T \\
&\vdots \\
&\lim_{n\to\infty} \boldsymbol{\pi}(n) = [0.4667\ 0.5333]^T
\end{aligned}$$

Dessa maneira, qualquer condição inicial pode ser escolhida sem que a distribuição estacionária se modifique. Alternativamente, esse procedimento pode ser evitado calculando-se o autovetor associado ao autovalor 1 de $\mathbb{P}$, normalizado para a soma dos elementos ser igual a 1. Isso pode ser feito, por exemplo, através da resolução do sistema de equações

$$\begin{cases}
\pi_0 + \pi_1 = 1 \\
\pi_0 = 0.2\pi_0 + 0.7\pi_1 \\
\pi_1 = 0.8\pi_0 + 0.3\pi_1
\end{cases},$$

como apresentado em (2.43). Escolhendo a primeira equação e uma das outras restantes (isso porque uma das equações é combinação linear das outras duas), resolve-se o sistema, obtendo $\boldsymbol{\pi} = [0.4667\ 0.5333]^T$, tal como calculado antes. De

fato, se for considerada uma potência de ordem 15 da matriz $\mathbb{P}$, já é possível observar que as colunas da matriz tendem para o valor de $\boldsymbol{\pi}$,

$$\mathbb{P}^{15} = \begin{bmatrix} 0.4667 & 0.4667 \\ 0.5333 & 0.5333 \end{bmatrix},$$

que corresponde ao valor de $\mathbb{P}^{\infty}$.

3 FILTROS DIGITAIS FIXOS E EFEITOS DE PRECISÃO FINITA

3.1 Filtros digitais

A aplicação de filtros digitais abrange diversas áreas, desde sistemas de comunicações até processamento de imagens, sinais biológicos e síntese de voz. Em qualquer situação, sempre se busca o melhor aproveitamento dos recursos de *software* e *hardware* disponíveis. Existem, basicamente, dois tipos de filtros digitais: filtros digitais de resposta ao impulso finita (FIR – em inglês *finite impulse response*) e filtros digitais de resposta ao impulso infinita (IIR – em inglês *infinite impulse response*). Filtros não-recursivos, cuja saída em qualquer instante não depende da saída em outros instantes, são filtros FIR. Filtros IIR apresentam realimentação da saída, o que os torna muito vantajosos por permitir a obtenção de respostas mais seletivas com um número menor de coeficientes, quando comparados com filtros FIR. Em (3.1) e (3.2) são apresentadas funções de rede típicas de filtros FIR e IIR no domínio da transformada Z e em (3.3) e (3.4) são mostradas as equações de diferenças correspondentes.

$$H_{FIR}(z) = \sum_{k=0}^{N} b_k z^{-k}, \tag{3.1}$$

$$H_{IIR}(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}}, \tag{3.2}$$

$$y_{FIR}(n) = \sum_{k=0}^{N} b_k x_{FIR}(n-k) \tag{3.3}$$

e

$$y_{IIR}(n) = \sum_{k=0}^{N} b_k x_{IIR}(n-k) - \sum_{k=1}^{M} a_k y_{IIR}(n-k). \tag{3.4}$$

Nessas equações, os termos a_k e b_k correspondem aos coeficientes dos filtros, enquanto $x_{FIR,IIR}(n)$ e $y_{FIR,IIR}(n)$ correspondem à entrada e à saída dos filtros, respectivamente. M e N representam os índices máximos dos coeficientes a_k e b_k, com $a_0 = 1$.

Filtros digitais costumam ser classificados com relação à *ordem*. A ordem de um filtro digital está ligada à memória do sistema e corresponde ao número de contribuições previamente armazenadas na memória do processador e utilizadas para calcular a próxima saída do filtro. Por exemplo, o filtro

$$H_{IIR}(z) = \frac{0.2 + 0.5z^{-1}}{1 + 0.8z^{-1} + 0.16z^{-2}}$$

usa no máximo duas amostras anteriores ($y(n-1)$ e $y(n-2)$) da saída, o que o classifica como um filtro de segunda ordem.

Filtros FIR e IIR podem ser realizados de diversas formas. As mais básicas, segundo [1, 10], são descritas a seguir, considerando estruturas IIR, que são as usadas neste trabalho.

3.1.1 Formas diretas

Na forma direta, filtros IIR são implementados diretamente segundo a equação (3.2). As figuras 12 e 13 apresentam filtros IIR implementados na forma direta I e II, respectivamente, usando por conveniência $N = M$.

3.1.2 Realização em cascata

Em geral, quando um filtro de ordem mais elevada é necessário, usam-se cascatas de filtros, de forma que módulos de filtros de primeira e segunda ordem

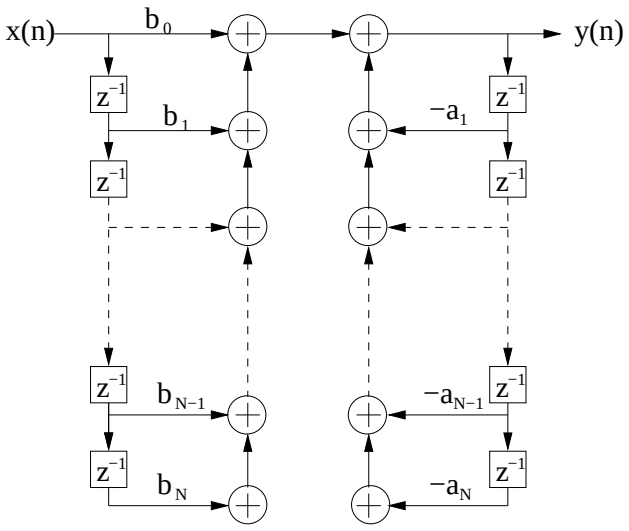

Figura 12: Filtro IIR implementado na forma direta I

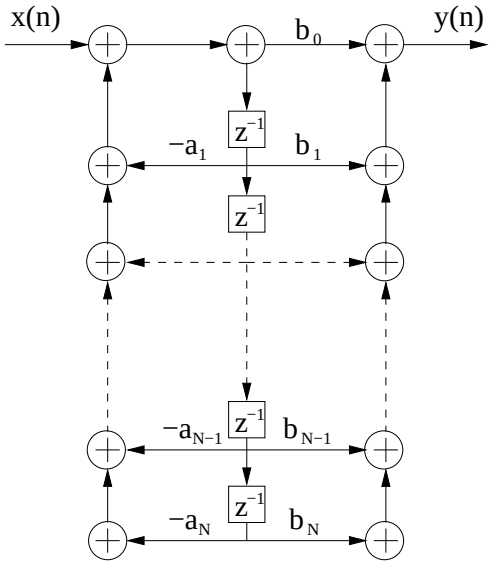

Figura 13: Filtro IIR implementado na forma direta II

são concatenados para obter a resposta desejada. Isso equivale a reescrever a equação (3.2) como um produtório, isto é,

$$H_{IIR}(z) = \prod_{k=1}^{N1} \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}, \tag{3.5}$$

em que $N1 = (N+1)/2$. A figura 14 mostra a implementação em cascata de estruturas de segunda ordem na forma direta II.

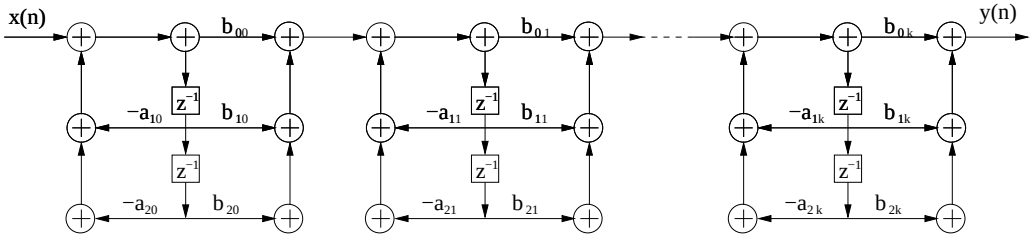

Figura 14: Filtro IIR implementado como uma cascata de filtros de segunda ordem na forma direta II

3.1.3 Realização paralela

Na implementação paralela, usa-se a expansão da equação (3.2) em termos de suas frações parciais,

$$H_{IIR} = \sum_{k=0}^{Np} c_k z^{-k} + \sum_{k=1}^{N1} \frac{e_{0k} + e_{1k}z^{-1}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}, \tag{3.6}$$

em que $Np = M - N$, $N1 = (N+1)/2$ e e_{0k}, e_{1k}, a_{1k} e a_{2k} são os coeficientes que compõem os filtros que irão definir a estrutura paralela. Os termos c_k ponderam versões anteriores da entrada e apenas são definidos quando $Np \geq 0$.

Na figura 15, é apresentado um exemplo de filtro implementado através da soma de blocos de filtragem em paralelo.

Além dessas formas, existem outras realizações possíveis que não são apresentadas aqui, como formas transpostas, em treliça, em espaço de estados e outras [2].

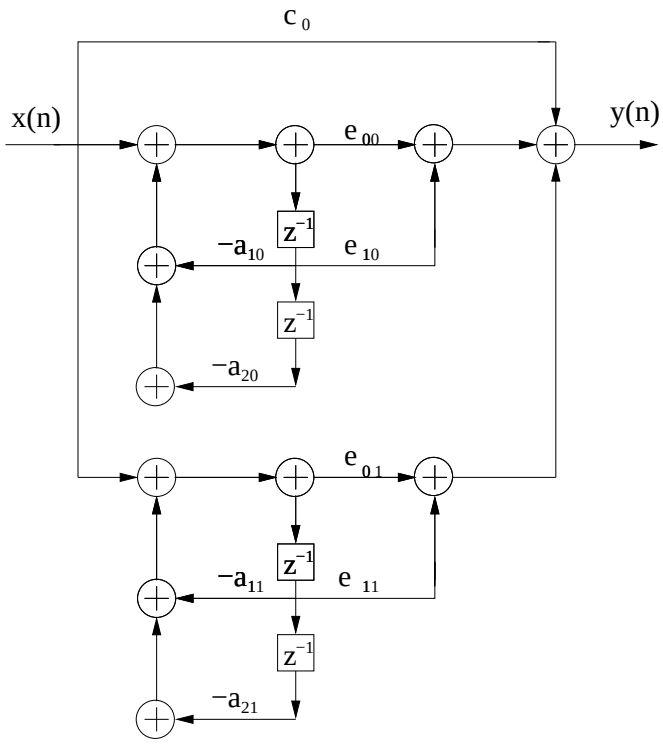

Figura 15: Filtro IIR implementado na forma paralela, usando blocos de segunda ordem na forma direta II

3.2 Efeitos de precisão finita

Na prática, um filtro fixo digital é implementado por *software* em um computador digital, em um DSP (processador digital de sinais – *digital signal processor*) ou em um *hardware* específico para uma aplicação desejada [1]. Os números oriundos do processamento são armazenados em registradores de tamanho finito, sob a forma de *palavras binárias* (ou seja, palavras compostas por zeros e uns). Dessa forma, coeficientes e sinais precisam ser acomodados nos registradores disponíveis, ou seja, precisam ser *quantizados* para valores que possam ser armazenados nos registradores. A quantização das grandezas provoca erros no processamento, que podem ser de três tipos [1, 2]:

1. Erros de quantização da entrada, devido à quantização em um conjunto de níveis discretos, oriundos da conversão analógico-digital dos sinais.

2. Erros de quantização de coeficientes, que precisam ser ajustados para o

comprimento dos registradores e podem alterar a resposta em frequência do filtro.

3. Erros de quantização em dados internos, que aparecem na saída dos multiplicadores.

Esses erros estão todos ligados à aritmética de precisão finita usada na implementação e podem ter um grande impacto sobre a saída do filtro. Pode-se implementar um filtro em dois tipos de aritmética: aritmética de ponto fixo e de ponto flutuante. A primeira costuma ser encontrada em *hardwares* de aplicação específica, por ser menos custosa em termos de área de *chip* e por ser mais simples de projetar. A aritmética de ponto flutuante pode ser encontrada em computadores de uso geral e em algumas famílias de DSPs e tem como característica o baixo ruído de quantização [10] e a maior facilidade de desenvolvimento de *software*.

3.2.1 Sistema binário

Um número positivo G qualquer pode ser representado em notação binária, como

$$G = \sum_{k=-m}^{n} g_k \, 2^{-k}, \tag{3.7}$$

em que os g_k podem assumir os valores 0 ou 1. Se for definido que G pertence ao intervalo $[0,\ 1]$, (3.7) pode ser reescrita como

$$G = \sum_{k=0}^{n} g_k \, 2^{-k} = g_0 + g_1 \, 2^{-1} + \ldots + g_n \, 2^{-n}. \tag{3.8}$$

A representação descrita em (3.8) é uma extensão para números binários da notação decimal usada em cálculos manuais, que não é a mais adequada para usar em computadores. De modo mais geral, os elementos g_k podem ser relacionados a um número G através de uma função como

$$\beta(G) = g_0.g_1 g_2 \ldots g_{n-1} g_n, \tag{3.9}$$

em que

$$G = \beta^{-1}(g_0.g_1g_2 \ldots g_{n-1}g_n), \tag{3.10}$$

segundo a notação usada em [1]. Essa forma corresponde à maneira como G pode ser armazenada em um registrador. No caso da representação descrita por (3.8), g_0 representa a parte inteira de G enquanto $(g_1g_2 \ldots g_{n-1}g_n)$ corresponde à parte fracionária. Para representar números negativos, há diversas alternativas, algumas das quais serão vistas a seguir.

3.2.2 Representação em ponto fixo

Em ponto fixo, usam-se três representações: sinal-módulo, complemento-a-um e complemento-a-dois.

3.2.2.1 Representação em sinal-módulo

Nessa representação, o bit mais significativo representa o sinal do número, seguido de um valor binário representado por seu módulo. Nesse caso, o valor 0 representa o sinal (+), enquanto 1 equivale ao sinal (−). Com isso, a representação de G em sinal-módulo $[G]_M$ é dada por

$$[G]_M = \begin{cases} \beta^{-1}(0.g_1 \ldots g_n) = (g_1\ 2^{-1} + \ldots + g_n\ 2^{-n}), & \text{se } \ g_0 = 0 \\ -\beta^{-1}(0.g_1 \ldots g_n) = -(g_1\ 2^{-1} + \ldots + g_n\ 2^{-n}), & \text{se } \ g_0 = 1. \end{cases} \tag{3.11}$$

3.2.2.2 Representação em complemento-a-um

Em complemento-a-um, um número G é representado como

$$[G]_{C1} = \begin{cases} \beta(G) & \text{se } \ G \geq 0 \\ \beta(2 - 2^{-n} - |G|), & \text{se } \ G < 0, \end{cases} \tag{3.12}$$

em que n é o máximo comprimento de palavra binária e $|\cdot|$ é a função módulo. Tal como a representação em sinal-módulo, os números positivos apresentam o

bit mais significativo igual a 0 se forem positivos e 1 se forem negativos. Para se obter o negativo de um número, basta trocar todos os 1 por 0 e todos os 0 por 1 (obtendo o que é chamado de *complemento* do número).

3.2.2.3 Representação em complemento-a-dois

Em complemento-a-dois, o número G é representado como

$$[G]_{C2} = \begin{cases} \beta(G) & \text{se } G \geq 0 \\ \beta(2 - |G|), & \text{se } G < 0, \end{cases} \tag{3.13}$$

de forma que é mantida a ideia de que o bit mais significativo igual a 0 corresponde a um número positivo e o bit mais significativo igual a 1 representa um número negativo. Nesta notação, para obter o negativo de um número, basta somar 1 ao bit menos significativo do complemento de $\beta(G)$. Um importante fato dessa representação e da representação em complemento-a-um [2] é que se o resultado da soma de diversos termos estiver dentro da faixa representável, então a soma será sempre correta, mesmo que ocorra *overflow* durante as somas parciais (o *overflow* corresponde à situação em que a adição de dois números binários altera erradamente o bit de sinal do resultado, o que será melhor explicado na Seção 3.2.4.1).

3.2.3 Representação em ponto flutuante

Uma outra forma de representar um número é através da aritmética de ponto flutuante. Nesse caso, um número é representado por

$$G = G_m 2^e, \tag{3.14}$$

em que G_m é a *mantissa* de G e e é o *expoente*, com $1/2 \leq |G_m| < 1$. Para armazenar a mantissa e o expoente de G durante a implementação, o registrador é dividido em duas partes. A principal vantagem da representação em ponto flutu-

ante é a ampla faixa dinâmica. Em contrapartida, a mantissa requer quantização tanto após multiplicações quanto após somas. Essa representação não será usada ao longo do texto.

3.2.4 Quantização

Apesar de existirem as duas formas de representação citadas anteriormente, em ponto fixo e ponto flutuante, a partir de agora somente estruturas implementadas em ponto fixo serão consideradas. É possível estender os resultados apresentados aqui para ponto flutuante, embora o desenvolvimento seja um pouco mais complicado.

Quando se define o comprimento de um registrador, fica estabelecido um conjunto finito de números que pode ser representado e armazenado. Se o tamanho do registrador for igual a B, a menor variação que pode ser representada é dada por $\Delta = 2^{-B+1}$, que corresponde ao bit menos significativo. Caso haja um número cuja representação binária exceda B bits, ele precisará ser *truncado* ou *arredondado* para o valor mais próximo dessa representação. Independentemente da forma escolhida, haverá um *erro de quantização*, definido como [2]

$$e = G - Q[G], \tag{3.15}$$

em que $Q[\cdot]$ corresponde à operação de quantização. Tradicionalmente, para análises de sistemas quantizados, assume-se que [10]

1. Cada fonte de quantização do filtro é considerada um processo branco e estacionário no sentido amplo;

2. Cada fonte de ruído de quantização tem distribuição uniforme no intervalo de quantização;

3. Cada fonte de ruído de quantização é descorrelacionada com relação à en-

trada do quantizador correspondente, com relação às outras fontes e com relação à entrada do filtro.

Se for considerada uma representação em complemento-a-dois limitada ao intervalo $[-1,\ 1]$, o erro $e_a(n)$ de arredondamento (vide figura 16 para $B = 3$ bits e $\Delta = 1/4$) está limitado a $-\Delta/2 \leq e_a(n) < \Delta/2$ e sua média corresponde a $\mathrm{E}\{e_a(n)\} = 0$. Nesse caso, a variância é dada por

$$\sigma^2_{e_a} = \frac{\Delta^2}{12}. \tag{3.16}$$

Caso seja considerado truncamento, o resultado é sempre menor que o valor

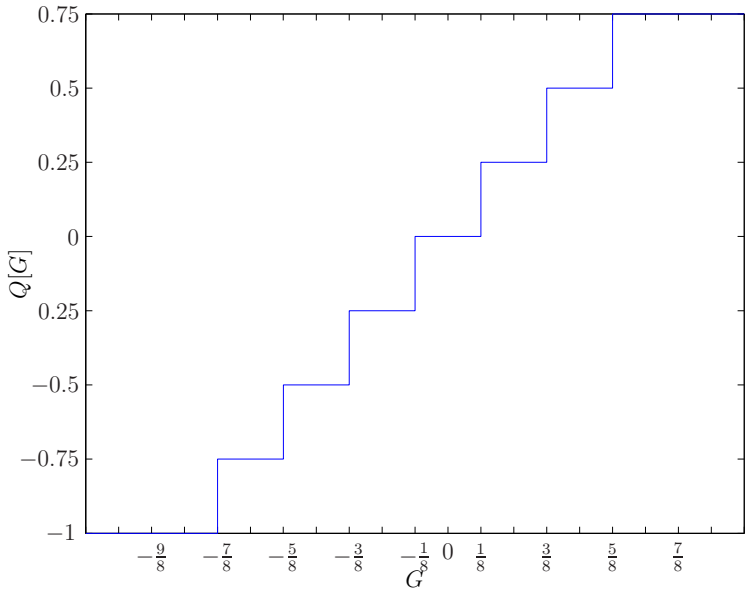

Figura 16: Arredondamento em uma implementação de 3 bits

original e $e_T(n)$ está no intervalo $-\Delta \leq e_T(n) < 0$ (vide figura 17 para $B = 3$ bits e $\Delta = 1/4$). Dessa forma, a média e a variância são calculadas por

$$\mathrm{E}\{e_T(n)\} = -\frac{\Delta}{2} \tag{3.17}$$

e

$$\sigma^2_{e_T} = \frac{\Delta^2}{12}, \tag{3.18}$$

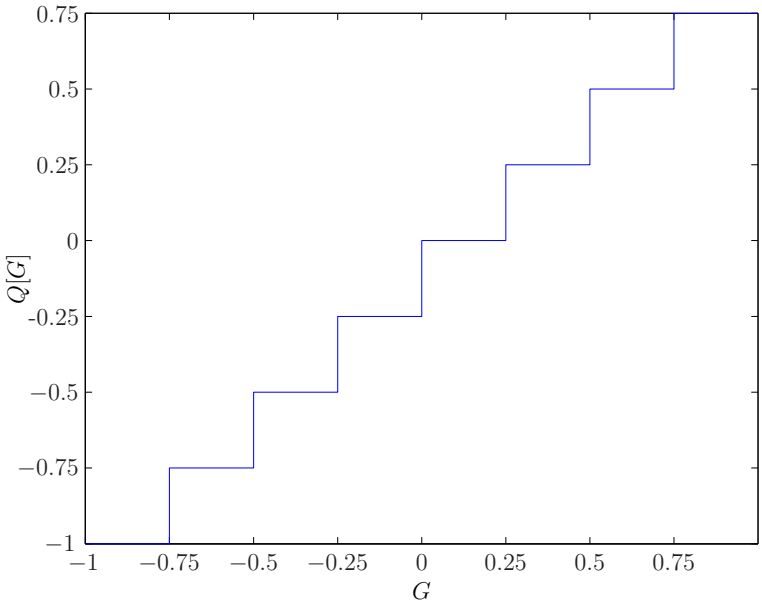

Figura 17: Truncamento em uma implementação de 3 bits

respectivamente. Se for, ainda, usado *truncamento em módulo*, obtém-se

$$\mathrm{E}\{e_{TM}(n)\} = 0 \tag{3.19}$$

e

$$\sigma^2_{e_{TM}} = \frac{\Delta^2}{3}. \tag{3.20}$$

Dessa forma, o arredondamento é a forma de quantização que apresenta melhores características, dado que o erro ou ruído de quantização possui média nula e menor variância.

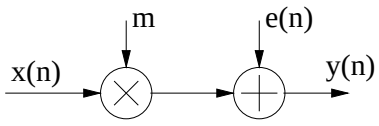

Figura 18: Modelo do erro de quantização após multiplicador

Quando se realiza uma multiplicação em ponto fixo, o número de bits necessários corresponde à soma do número de bits dos fatores subtraída de uma unidade e, portanto, é necessário quantizar o resultado. Por outro lado, o resultado de uma soma em notação de ponto fixo, caso o resultado não exceda a

representação utilizada, é sempre exato, dispensando quantização. Por esse motivo, filtros digitais são implementados com quantizadores após as multiplicações, de forma que o resultado da multiplicação seja ajustado para o número de bits disponível.

Utilizando arredondamento, truncamento ou truncamento em módulo, pode-se modelar o efeito da quantização através da soma de um erro $e(n)$ de média $\mathrm{E}\{e(n)\}$ e variância σ_e^2 após cada multiplicador [1, 2, 10]. Essa é a forma usualmente aplicada para a modelagem de efeitos não-lineares em filtros, por apresentar resultados satisfatórios quando os filtros são implementados com muitos bits. Contudo, quando é usado um número reduzido de bits, o erro envolvido na quantização das grandezas passa a ser mais significativo frente aos valores representáveis, tornando o modelo linearizado ineficiente. Como opção ao modelo linear, neste trabalho são usadas cadeias de Markov. Essa abordagem fornece um resultado exato, já que são considerados os efeitos de todas as não-linearidades envolvidas. Nas próximas seções, os modelos são comparados, com o intuito de mostrar que a aplicação de cadeias de Markov pode ser vantajosa, principalmente quando se consideram implementações com poucos bits.

3.2.4.1 *Overflow* e saturação

Durante o processo de filtragem implementado em precisão finita, pode-se obter um sinal cujo valor excede o máximo ou o mínimo da representação binária (por exemplo, para uma representação em complemento-a-dois de 3 bits, a soma de dois números positivos 011 e 001 excede o valor máximo da representação positiva, alterando o bit de sinal). Nesse caso, é necessário usar algum método para determinar uma forma representável do sinal, o que pode ser feito por meio de duas possíveis abordagens: *overflow* ou *saturação*.

O *overflow* é uma situação de exceção em que o bit de sinal é alterado inde-

vidamente durante as operações, devido à obtenção de um número que excede a representação usada. Nesse caso, pode-se permitir que o valor obtido após uma operação em que ocorre *overflow* seja mantido (o que corresponde a tratar a exceção por *overflow*) ou pode-se saturar os valores que excedem a representação em um valor mínimo ou máximo (o que corresponde a tratar a exceção por *saturação*.

Exemplo: Sejam os números 011 e 001 dois valores em notação de complemento-a-dois de 3 bits, representando números entre -1 e 1. A tabela 1 mostra a correspondência entre a notação binária e os sinais.

Tabela 1: Representação binária para valores entre -1 e 1 em complemento-a-dois e considerando 3 bits

Valores	-1	-0.75	-0.5	-0.25	0	0.25	0.5	0.75
Representação binária	100	101	110	111	000	001	010	011

Nesse caso, a soma de $0.75 + 0.25$ que deveria resultar 1, é dada em notação binária por $011 + 001 = 100$, que equivale a -1. Da mesma forma, a soma $0.75 + 0.75 = 1.5$ corresponde a $0.11 + 0.11 = 110$, fornecendo o resultado -0.5. Portanto, quando o resultado da soma excede os limites da representação, a aritmética de complemento-a-dois altera o bit de sinal do resultado, em um processo de exceção conhecido como exceção por *overflow*. Essa forma de tratar as exceções (considerando arredondamento para cima) é apresentada na figura 19, considerando sinais de 3 bits e $\Delta = 1/4$.

Uma outra forma de lidar com os resultados que excedem a representação binária é por meio de saturação. Para essa situação, quando o resultado de uma soma excede o valor mínimo da representação, ele é limitado ao valor do limite inferior da representação. Da mesma forma, sinais que excedem o limite máximo representável são saturados nesse valor. Portanto, se fosse aplicada a exceção por saturação nas duas somas anteriores, o resultado seria limitado a

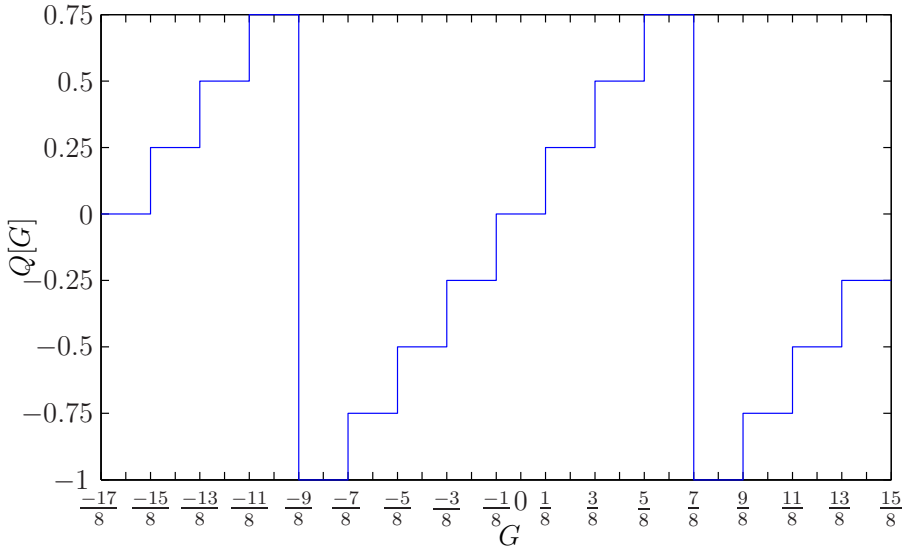

Figura 19: Exceção de *overflow* para um sinal de 3 bits

011, que corresponde a 0.75. A figura 20 apresenta a saturação para a notação em complemento-a-dois com 3 bits, usando arredondamento.

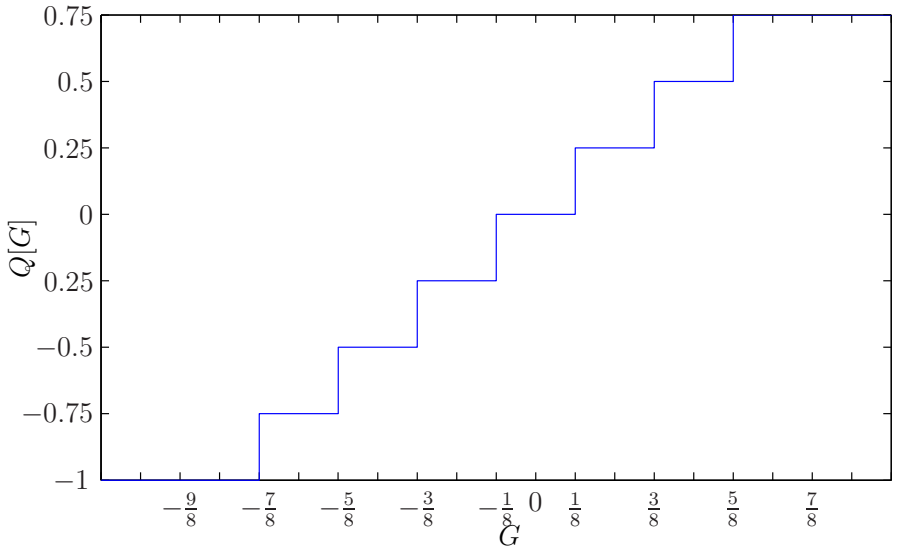

Figura 20: Exceção de saturação

3.2.5 Escalamento da entrada

Quando a entrada de um filtro digital é elevada, deve-se fazer o escalamento para reduzir a chance de *overflow* nos nós internos da estrutura do filtro. Todavia, se o filtro operar com sinais de amplitude muito baixa, a relação sinal-ruído será

baixa [2]. Portanto, o escalamento da entrada deve ser realizado levando em conta essas duas situações.

Se a representação em complemento-a-um ou em complemento-a-dois estiver sendo usada, somente é necessário fazer o escalamento dos valores na entrada dos multiplicadores para evitar *overflow*. Isso se deve ao fato de que nessas representações, se o resultado da soma de dois ou mais números estiver dentro da faixa representável, ela estará sempre correta, independentemente da ordem da soma e da existência de *overflow* em operações intermediárias. Com isso, se a entrada do filtro $x(n)$ for limitada por um certo M, tal que $|x(n)| \leq M$, para reduzir a chance de *overflow* a níveis aceitáveis após cada multiplicação, deve-se multiplicar $x(n)$ (vide figura 21) por um fator de escalamento m_{esc}, de forma que $|v_i(n)|$ de cada multiplicador i não exceda a faixa representável.

Segundo [2], existem duas maneiras mais comumente usadas para determinar m_{esc}:

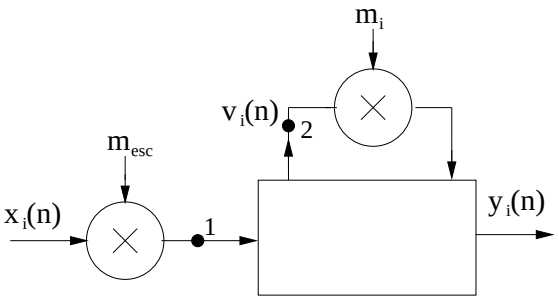

Figura 21: Escalamento da entrada de um filtro digital

1. **Método A:** calcula-se a função de transferência $F_i(z)$ entre os pontos 1 e 2 (vide figura 21) para encontrar $v_i(n)$, que é dado pela convolução

$$v_i(n) = \sum_{k=0}^{\infty} m_{esc} f_i(k) x(n-k), \tag{3.21}$$

em que $f_i(k)$ é a transformada Z inversa de $F_i(z)$. Nesse caso, tomando os

módulos em ambos os lados da equação,

$$|v_i(n| \leq \sum_{k=0}^{\infty} |m_{esc} f_i(k)| \cdot |x(n-k)| \tag{3.22}$$

e lembrando que $|x(n)| \leq M$, obtém-se

$$|v_i(n)| \leq M \sum_{k=0}^{\infty} |m_{esc} f_i(k)|. \tag{3.23}$$

Uma condição suficiente para que $|v_i(n)| \leq M$ é que

$$\sum_{k=0}^{\infty} |m_{esc} f_i(k)| \leq 1 \ \Rightarrow m_{esc} \ \leq \frac{1}{\sum_{k=0}^{\infty} |f_i(k)|}. \tag{3.24}$$

A condição (3.24) garante que não ocorrerá *overflow* no filtro, mas não considera que o escalamento pode levar a sinais internos de amplitudes muito baixas. Essa condição equivale à desigualdade de Hölder [9] para sequências, com $p = 1$, ou seja,

$$|v_i(n)| = \|x(n)\|_q \|m_{esc} f_i(n)\|_p, \ \text{ com } \ \frac{1}{p} + \frac{1}{q} = 1, \tag{3.25}$$

onde as normas ℓ_p, para $p = \infty$, são calculadas por

$$\|x(n)\|_\infty = \sup|x(n)| = M \tag{3.26}$$

e

$$\|m_{esc} f_i(n)\|_\infty = \sum_{k=0}^{\infty} |m_{esc} f_i(k)| = |m_{esc}| \sum_{k=0}^{\infty} |f_i(k)|. \tag{3.27}$$

2. **Método B:** Usa-se normas L_p [9] para fazer o escalamento da entrada no domínio transformado. Pode-se escrever

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{3.28}$$

e

$$V_i(z) = m_{esc} F_i(z) X(z). \tag{3.29}$$

No domínio do tempo, pode-se calcular $v_i(n)$ como

$$v_i(n) = \frac{1}{2\pi j} \oint_C V_i(z) z^{n-1} dz, \tag{3.30}$$

em que C é a região de convergência comum a $F_i(z)$ e a $X(z)$. Mas, de (3.29), obtém-se

$$v_i(n) = \frac{1}{2\pi j} \oint_C m_{esc} F_i(z) X(z) z^{n-1} dz, \tag{3.31}$$

cujo correspondente no domínio da frequência será

$$v_i(n) = \frac{1}{2\pi} \int_0^{2\pi} m_{esc} F_i(e^{j\omega}) X_i(e^{j\omega}) e^{j\omega n} d\omega. \tag{3.32}$$

Se for usada a norma L_p,

$$\| F_i(e^{j\omega}) \|_p = \left(\frac{1}{2\pi} \int_0^{2\pi} |F_i(e^{j\omega})|^p d\omega \right)^{\frac{1}{p}} \tag{3.33}$$

para $\int_0^{2\pi} |F_i(e^{j\omega})|^p d\omega < \infty$ e $p \geq 1$. Para as normas L_1, L_2 e L_∞ isso é equivalente a

$$\| F(e^{j\omega}) \|_1 = \frac{1}{2\pi} \int_0^{2\pi} |F(e^{j\omega})| d\omega,$$

$$\| F(e^{j\omega}) \|_2 = \left(\sum_{k=1}^{n} |f_i(k)|^2 \right)^{\frac{1}{2}}$$

e

$$\| F(e^{j\omega}) \|_\infty = \max_{0 \leq \omega \leq 2\pi} \{ |F(e^{j\omega})| \}.$$

Com isso, pode-se encontrar situações particulares da desigualdade de Hölder

$$|v_i(n)| \leq \| m_{esc} F_i(e^{j\omega}) \|_1 \cdot \| X(e^{j\omega}) \|_\infty, \tag{3.34}$$

$$^1|v_i(n)| \leq \| m_{esc} F_i(e^{j\omega}) \|_2 \cdot \| X(e^{j\omega}) \|_2 . \tag{3.35}$$

e

$$|v_i(n)| \leq \| m_{esc} F_i(e^{j\omega}) \|_\infty \cdot \| X(e^{j\omega}) \|_1 . \tag{3.36}$$

[1]Essa desigualdade corresponde à desigualdade de Schwarz, que é um caso particular da desigualdade de Hölder [9]

Dessa forma, se $|X(e^{j\omega})|$ for limitado superiormente por M, para que $v_i(n)$ também seja limitado por M é necessário que $\| m_{esc}F_i(e^{j\omega}) \|_p \leq 1$. Com isso,

$$m_{esc} \leq \frac{1}{\| F_i(e^{j\omega}) \|_p}. \tag{3.37}$$

Na prática, usa-se (3.34) quando $X(e^{j\omega})$ é limitada, enquanto (3.35) é usada para sinais cuja energia da entrada é finita. A equação (3.36) é aplicada quando $X(e^{j\omega})$ possui uma componente predominante de frequência. Nesse caso, as normas $\| X(e^{j\omega}) \|_2$ e $\| X(e^{j\omega}) \|_\infty$ não são definidas e somente $\| X(e^{j\omega}) \|_1$ e $\| F(e^{j\omega}) \|_\infty$ $(p = \infty)$ podem ser usadas [1].

No caso de existir mais de um multiplicador no filtro, o fator de escala é escolhido através de

$$m_{esc} = \frac{1}{\max\{\| F_1 \|_p, \| F_2 \|_p, \ldots, \| F_n \|_p\}}, \tag{3.38}$$

usando como base o pior caso para definir um fator que satisfaça a necessidade de escalamento de todos os multiplicadores.

3.2.6 Quantizações de coeficientes

Embora não seja o objetivo deste trabalho, a quantização de coeficientes de filtros digitais pode influenciar a estabilidade da resposta e provocar efeitos indesejados na saída. Por esse motivo, alguns de seus efeitos são apresentados a seguir.

No projeto de um filtros digitais, os coeficientes são calculados considerando alta precisão. Quando esses coeficientes são implementados na prática, é necessário ajustá-los para o comprimento dos registradores, o que implica a quantização dos coeficientes e possíveis alterações na resposta em frequência, dado que são introduzidas perturbações nos pólos e zeros do filtro. Dependendo da estrutura usada na implementação, a sensibilidade da resposta do filtro aos erros

de quantização dos coeficientes pode fazer a resposta muito diferente do desejado, o que justifica a busca por estruturas menos sensíveis aos erros dos coeficientes. Basicamente, para se identificar e reduzir o efeito da quantização dos coeficientes faz-se a análise da sensibilidade da resposta em relação à variação dos coeficientes, definindo estatisticamente coeficientes com o comprimento de palavra desejado que minimizem a distância entre a resposta obtida e a desejada.

A análise da sensibilidade da resposta em termos da variação dos coeficientes é baseada na observação de

$$S_{c_i} = \frac{\partial |H(e^{j\omega})|}{\partial c_i}, \tag{3.39}$$

em que S_{c_i} corresponde à variação da amplitude da resposta $|H(e^{j\omega})|$ com respeito à variação de cada coeficiente c_i. Nesse caso, a variação total da amplitude, $\Delta|H(e^{j\omega})|$, será dada por

$$\Delta|H(e^{j\omega})| = \sum_{i=1}^{m} \Delta c_i S_{c_i}^M. \tag{3.40}$$

A escolha de um valor máximo desejado para a variação ($\Delta|H(e^{j\omega})|_{max}$), tal que seja possível calcular a probabilidade $P(\Delta|H(e^{j\omega})| < \Delta|H(e^{j\omega})|_{max})$, permite definir estatisticamente um comprimento de palavra, de forma que seja obedecida uma confiabilidade desejada para os coeficientes do filtro [2]. Dessa forma, através das sensibilidades S_{c_i}, pode-se escolher o comprimento dos coeficientes do filtro digital. Análises baseadas na sensibilidade da resposta aos coeficientes podem ser observadas em trabalhos como [11, 12, 13, 14], por exemplo.

Existem implementações mais sensíveis a erros de quantização do que outras. Pode-se observar isso nas formas diretas, que são raramente implementadas em filtros maiores do que de segunda ordem, por serem mais sensíveis aos erros de quantização dos coeficientes do que implementações paralelas e em cascata [10].

Na literatura, existem ainda outras formas de determinar o comprimento

de palavra dos coeficientes, como o uso de algoritmos genéticos [15, 16] e uso de informações sobre a complexidade do *hardware* associada a informações dos sinais envolvidos para definir a sensibilidade usada no cálculo das palavras [17]. Ainda assim, a maneira mais comum de se definir o comprimento dos coeficientes é usando funções de sensibilidade semelhantes à (3.39).

3.2.7 Ciclos-limite

Ciclos-limite são oscilações parasitas que aparecem na saída de filtros recursivos quando a entrada é nula ou constante. Nessas circunstâncias, os erros de quantização tendem a ficar altamente correlacionados, fazendo com que a saída do filtro oscile. De acordo com sua origem, ciclos-limite são classificados como granulares (ou por quantização) e por *overflow*.

3.2.7.1 Ciclos-limite granulares

Em precisão infinita, se a entrada de um filtro IIR estável se torna zero, a saída deve tender assintoticamente para zero. Contudo, quando o mesmo filtro é implementado em aritmética de precisão finita, os sucessivos arredondamentos ou truncamentos dos produtos podem levar a padrões repetitivos, fazendo com que a saída oscile. Nesse caso, diz-se que o filtro apresenta ciclos-limite granulares. Essas oscilações são relacionadas aos bits menos significativos e são extremamente indesejadas, sendo necessário eliminá-las ou, pelo menos, manter sua amplitude limitada.

3.2.7.2 Ciclos-limite por *overflow*

Os ciclos-limite por *overflow* podem ocorrer quando os sinais excedem a faixa permitida pelos registradores. Nesse caso, as exceções de *overflow* (semelhantes às apresentadas na figura 19) podem levar a oscilações de amplitude elevada na

saída do filtro digital, já que os os bits mais significativos são atingidos [1].

3.2.7.3 Evitando ciclos-limite

Um filtro recursivo é dito livre de ciclos-limite se as oscilações decrescem com o tempo e a saída converge para o valor da saída do filtro ideal. Ciclos-limite são um problema de grande interesse em engenharia e existe uma grande quantidade de trabalhos com o intuito de identificar estruturas livres dessas oscilações.

Existem diversos trabalhos que estudam ciclos-limite em filtros digitais representados no espaço de estados (como [18, 19]), isto é, na forma

$$\begin{cases} \boldsymbol{x}(n+1) = Q\left[\boldsymbol{A}\boldsymbol{x}(n) + \boldsymbol{b}u(n)\right] \\ y(n) = \boldsymbol{c}^T\boldsymbol{x}(n) + du(n) \end{cases}, \tag{3.41}$$

onde $\boldsymbol{x}$ é um vetor de estados, $u(n)$ é a entrada do filtro e $y(n)$ é a saída. A

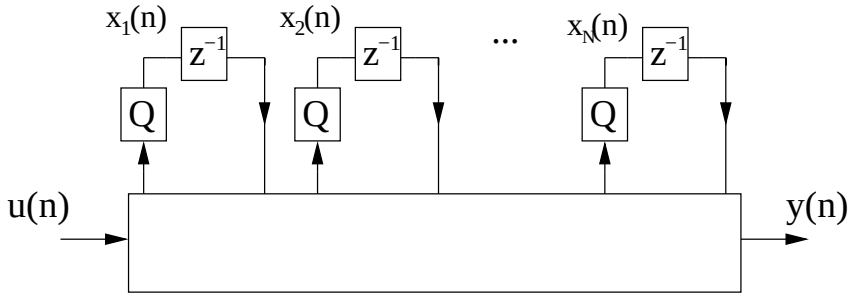

Figura 22: Filtro digital com quantizadores

matriz $\boldsymbol{A}$ é a matriz de estados, $\boldsymbol{b}$ e $\boldsymbol{c}$ são vetores que correspondem à entrada e à saída e d representa a conexão direta entre entrada e saída do filtro. A função $Q[\cdot]$ corresponde à forma quantizada do valor entre colchetes.

A base para a eliminação de ciclos-limite de entrada zero está na análise da parte recursiva da equação (3.41) ($Q\left[\boldsymbol{A}\boldsymbol{x}(n)\right]$). Em [1], é mostrado que basta que a matriz $\boldsymbol{A}$ corresponda a um filtro estável e que exista uma matriz positiva definida diagonal $\boldsymbol{D}$ tal que

$$\hat{\boldsymbol{x}}^T(\boldsymbol{D} - \boldsymbol{A}^T\boldsymbol{D}\boldsymbol{A})\hat{\boldsymbol{x}} \leq 0, \tag{3.42}$$

para qualquer vetor $\hat{\boldsymbol{x}}$, de forma que usando apenas o *truncamento de módulo* é possível eliminar os ciclos-limite granulares de entrada zero. Para filtros de segunda ordem, em [2] é mostrado que basta que os elementos de $\boldsymbol{A}$ cumpram

$$a_{12}a_{21} \geq 0 \tag{3.43}$$

ou

$$\begin{cases} a_{12}a_{21} < 0 \\ |a_{11} - a_{22}| + \det(\boldsymbol{A}) \leq 1 \end{cases} \tag{3.44}$$

para que a equação (3.42) seja satisfeita. Nesse caso, $\det(\boldsymbol{A})$ corresponde ao determinante de $\boldsymbol{A}$.

De fato, o argumento em (3.42) também pode ser aproveitado para evitar ciclos-limite de entrada constante, através da modificação da estrutura da figura 22 para a da figura 23 (segundo apresentado em [1]), assumindo que o filtro da

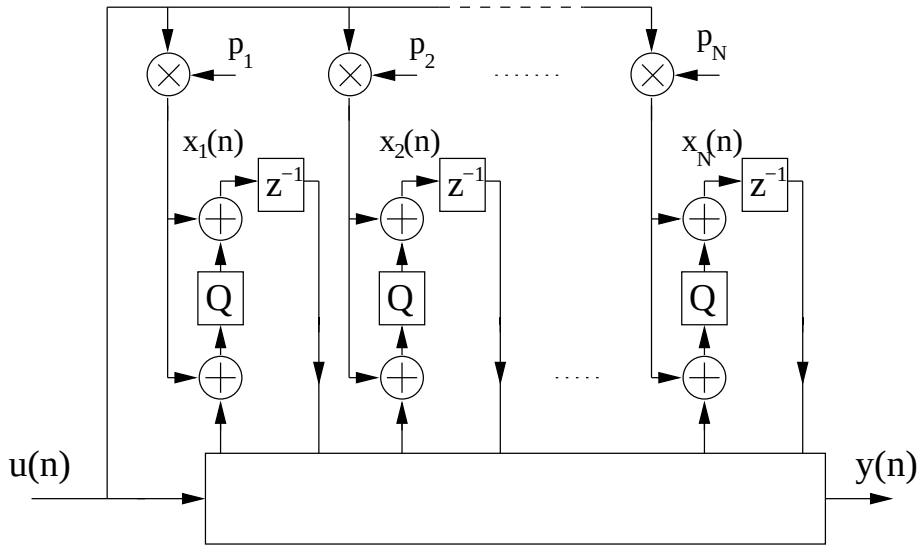

Figura 23: Filtro digital com quantizadores

figura 22 não possui ciclos-limite de entrada nula. Nesse caso, com $\boldsymbol{p}$ igual a

$$\boldsymbol{p} = [p_1 \;\; p_2 \ldots p_N]^T = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b}, \tag{3.45}$$

mostra-se que com a mudança de variável $\boldsymbol{x}'(n) = \boldsymbol{x}(n) - \boldsymbol{p}u(n)$ na equação do

filtro

$$\boldsymbol{x}(n) = Q\left[\boldsymbol{A}\boldsymbol{x}(n) - \boldsymbol{p}u(n) + \boldsymbol{b}u(n)\right] + \boldsymbol{p}u(n) \tag{3.46}$$

obtém-se

$$\boldsymbol{x}'(n) = Q\left[\boldsymbol{A}\boldsymbol{x}'(n)\right], \tag{3.47}$$

por onde se chega em uma equação semelhante à (3.42).

Baseados na descrição no espaço de estados de (3.42) existem muito estudos sobre ciclos-limite, como [18, 19], em que são derivadas condições suficientes e necessárias para evitar ciclos-limite de entrada nula. Em [20, 21, 22], também é usada a descrição no espaço de estados, associada a algoritmos de busca exaustiva. Nesse caso, definem-se "regiões" no espaço de estados onde os coeficientes não provocam o surgimento de ciclos-limite. A busca exaustiva é realizada durante o teste das diversas possibilidades para o estado inicial do filtro, com o intuito de definir regiões onde o filtro é globalmente assintoticamente estável. Essas abordagens conseguem fornecer bons resultados para a escolha dos coeficientes, mas costumam ser custosas e são geralmente evitadas.

Uma outra forma de se reduzir o efeito de ciclos-limite é a escolha de palavras de comprimento suficiente para minimizar a amplitude das oscilações, levando em conta as restrições da implementação. Em [11], é proposto um método analítico para definir palavras de comprimentos variados para coeficientes e sinais de filtros digitais implementados em FPGAs. Com isso, reduz-se o comprimento de registradores e evita-se ciclos-limite. Para isso, os coeficientes em precisão finita são modelados como se fossem os coeficientes ideais adicionados de uma perturbação. A análise da influência das perturbações dos coeficientes ideais nos pólos e zeros do filtro é usada para definir o número de bits dos coeficientes. O número de bits usado em cada coeficiente é então escolhido tendo por base as tolerâncias pré-determinadas no projeto para cada coeficiente. Já para os sinais, o comprimento da parte inteira é definido com base em um critério de limitação

de amplitude, com o intuito de evitar *overflow*. Para a determinação da parte fracionária, consideram-se os bits necessários para minimizar erros oriundos de truncamento.

Em [23], por outro lado, considera-se uma interpretação do filtro digital como uma máquina de estados finitos. As saídas possíveis do filtro são definidas como estados da máquina e a transição dos estados passados para os estados atuais é descrita por uma matriz de conectividade, composta por zeros e uns. Essa matriz apresenta todas as transições possíveis entre os estados da máquina (deve-se lembrar que em filtros recursivos as saídas passadas são usadas na obtenção da saída atual, o que pode ser encarado como uma transição de estado). A partir dessa matriz, são removidos todos os estados que não formam ciclos, restando apenas os ciclos-limite. Os ciclos-limite são removidos por meio da "quebra" dos ciclos, realocando a posição dos estados na matriz. A escolha da nova posição na matriz é definida por meio de uma função custo que minimiza o erro cometido ao se modificar a conexão entre os estados. Dessa forma, os ciclos-limite podem ser completamente eliminados.

Ao longo desse texto, uma outra abordagem será usada para prever e tentar eliminar ciclos-limite: cadeias de Markov. Como será mostrado posteriormente, essa maneira de encontrar os ciclos-limite é muito semelhante à proposta de [23], dado que quando a entrada é nula, o filtro se torna determinístico e as matrizes de conectividade e de Markov se tornam iguais. A diferença está na forma com que se trata a matriz, como será abordado no capítulo 4.

4 FILTROS DIGITAIS IMPLEMENTADOS COM ACUMULADOR DE PRECISÃO SIMPLES

Neste capítulo, são apresentados os cálculos para a obtenção da matriz de transição de estados em filtros implementados com um acumulador de precisão reduzida, em que é necessária a quantização das grandezas após cada multiplicação. Assume-se que a entrada dos filtros possui uma função densidade de probabilidade conhecida e descorrelacionada e calcula-se a matriz de transição de estados para filtros de primeira e de segunda ordem. Os valores da média e da variância obtidos com o modelo de cadeias de Markov é comparado aos valores calculados com a abordagem linearizada, através de exemplos. Mostra-se que é possível usar a matriz de transição de estados para escalar a entrada de filtros digitais e para definir implementações livres de ciclos-limite de entrada nula. Além disso, são estudados filtros implementados em cascata e modelados individualmente via cadeia de Markov.

4.1 Implementação com acumulador de precisão simples

Em aplicações extremamente econômicas em termos de *hardware*, um filtro digital pode ser implementado com um acumulador de comprimento de palavra semelhante ao da memória dedicada aos coeficientes e sinais. Nessa situação, após cada operação intermediária, é necessário ajustar o comprimento do resultado

para a precisão disponível, incorrendo em um erro provocado pelo grande número de não-linearidades.

Neste capítulo, são considerados filtros implementados de forma econômica, e a abordagem via cadeias de Markov é aplicada para estudar o efeito das não-linearidades. É importante frisar que o método pode ser aplicado para qualquer estrutura particular de implementação de filtros digitais, em particular também para implementações com acumuladores em precisão dupla (vide cap. 5). Para ilustrar o alcance do método, as probabilidades de transição de estados da cadeia de Markov são calculadas em duas situações. Na primeira, a fim de modelar o efeito causado pela não-linearidade de saturação, dois estados adicionais são definidos, com o objetivo de descrever a saída do filtro para valores maiores ou menores que o *range* da representação (respectivamente valores menores que -1 e maiores que [1] $1-\Delta$). Na segunda situação, a não-linearidade de *overflow* é usada nos valores fora da faixa de representação para o cálculo de $\mathbb{P}$.

A forma como as probabilidades são calculadas é apresentada a seguir, assumindo um filtro de segunda ordem (vide figura 24), descrito pela equação

$$\begin{aligned} y(n) = \; & R\left\{R\left\{Q\left[b_0x(n)\right] + R\left\{Q\left[b_1x(n-1)\right] + Q\left[b_2x(n-2)\right]\right\}\right\} + \\ & R\left\{Q\left[-a_1y(n-1)\right] + Q\left[-a_2y(n-2)\right]\right\}\right\}, \end{aligned} \tag{4.1}$$

onde $Q[\cdot]$ corresponde à não-linearidade de saturação (ou de *overflow*) após cada multiplicação, e $R\{\cdot\}$ é o ajuste realizado após uma soma. Os cálculos realizados para um filtro digital de primeria ordem (vide figura 25), isto é,

$$y(n) = \; R\left\{R\left\{Q\left[b_0x(n)\right] + Q\left[b_1x(n-1)\right]\right\} + Q\left[-a_1y(n-1)\right]\right\} \tag{4.2}$$

são uma parte do cálculo realizado para um filtro de segunda ordem. Por esse motivo, os resultados para filtros de primeira ordem são apresentados rapidamente

[1] De fato, em uma representação binária no conjunto $\{-1, 1\}$, os valores representáveis ficam limitados entre -1 e $(1-\Delta)$, obtidos em passos de $\Delta = 2^{-B+1}$

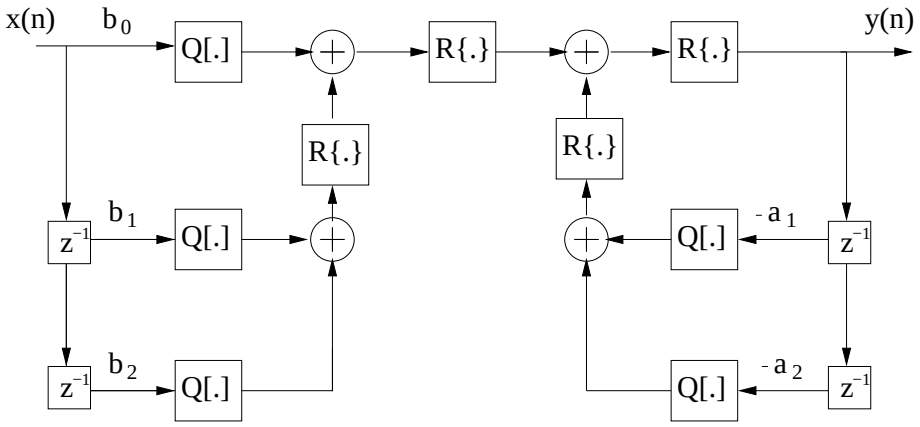

Figura 24: Filtro IIR de segunda ordem implementado na forma direta I, mostrando não-linearidades de precisão finita

ao longo do tratamento de filtros de segunda ordem.

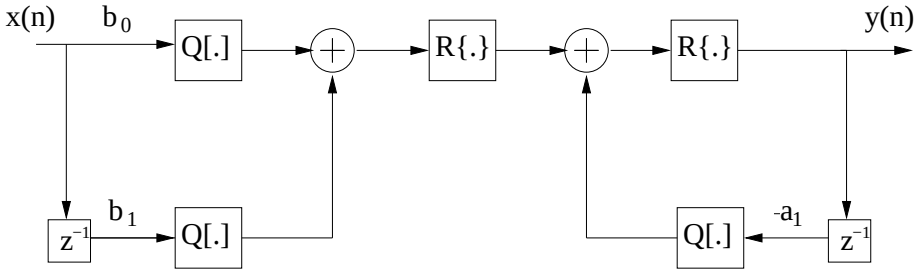

Figura 25: Filtro IIR de primeira ordem implementado na forma direta I, mostrando não-linearidades de precisão finita

A abordagem usada neste capítulo pode ser aplicada em outras formas de implementação, como o caso de arquiteturas com acumuladores intermediários com precisão dupla, que ocorrem na maioria dos DSPs (conforme é apresentado no cap. 5). Além disso, também é possível estender a aplicação de cadeias de Markov para filtros realizados de outras formas (por exemplo, as apresentadas na seção 3.1), sendo necessário modificar a maneira como a matriz de transição de estados é calculada, para que as não-linearidades próprias de cada estrutura sejam corretamente consideradas (vide seção 4.7, onde são estudados filtros implementados em cascata).

4.2 Cálculo da probabilidade de saturação usando cadeias de Markov

É possível encontrar a probabilidade da saída de um filtro digital de primeira ou de segunda ordem ser representado pelos valores definidos no domínio da representação, em função dos valores de entradas e de saídas anteriores do filtro. (No escopo deste texto, considera-se que os filtros digitais são implementados conforme o descrito nas equações (4.2) e (4.1).) A partir dessa consideração, é possível construir uma cadeia de Markov em que cada possível saída de um filtro corresponde a um estado da cadeia, que é dependente da saída (ou estado) anterior do mesmo filtro. Dessa forma, obtém-se a *matriz de transição de estados* $\mathbb{P}$, que pode ser usada para analisar seu funcionamento. Para um filtro de primeira ordem, as probabilidades descritas em $\mathbb{P}$ representam $P(y(n)|y(n-1))$, enquanto que para um filtro de segunda ordem, a probabilidade é definida por $P(y(n)|y(n-1), y(n-2))$, com $y(n),\ y(n-1),\ y(n-2) \in \{-1 + k\Delta,\ 0 \leq k < 2^B\}$.

Quando se aplica não-linearidade de saturação em um filtro digital, limita-se sua saída aos valores existentes em uma faixa de valores pré-estabelecida (por exemplo, limita-se a saída para valores entre -1 e $(1 - \Delta)$). Se for calculada a probabilidade de $y(n)$ assumir o valor -1 para um dado valor anterior da saída, por exemplo, o valor obtido corresponderá ao efeito somado de $y(n)$ ser igual a -1 e de $y(n)$ ser saturado para -1, sendo impossível distinguí-los. Nesse caso, a informação sobre a relevância da saturação para que a saída seja igual -1 é perdida. O mesmo ocorre para o limite superior da representação.

Para tornar visível o efeito da saturação, novos estados -1_S e $(1-\Delta)_S$ podem ser criados para representar, respectivamente, as saídas -1 e $(1 - \Delta)$ obtidas por saturação. Os estados adicionais fazem com que a matriz de transição de estados sofra um aumento do número de elementos. Originalmente, um filtro de primeira

ordem possui uma $\mathbb{P}$ quadrada de dimensões $N \times N$, em que N corresponde ao número de elementos da representação ($N = 2^B$). Se forem acrescentados 2 estados de saturação, a nova matriz será $(N+2) \times (N+2)$. Por outro lado, a $\mathbb{P}$ de um filtro de segunda ordem sem os estados de saturação apresenta dimensões $N^2 \times N^2$, enquanto a nova matriz é $(N+2)^2 \times (N+2)^2$.

Exemplo: Suponha um filtro IIR de primeira ordem e implementado com palavras de 2 bits (ou seja, o conjunto de números representáveis corresponde a $\{-1,\ -0.5,\ 0,\ 0.5\}$), descrito pela equação

$$H(z) = \frac{0.5}{1 + 0.5z^{-1}}.$$

Para esse filtro, a equação de diferenças equivalente (considerando a não-linearidade de saturação) é

$$y(n) = R\{Q[0.5x(n)] + Q[-0.5y(n-1)]\}.$$

Deseja-se calcular $\mathbb{P}$ quando a entrada possui f.d.p. uniforme e de média nula (ou seja, a probabilidade de $x(n) = -0.5$, 0 ou 0.5 é igual a $1/3$ e a probabilidade de $x(n) = -1$ é igual a 0). Considera-se que os números são arredondados para cima, ou seja, $Q[0.25] = 0.5$ e $Q[-0.25] = 0$.

Primeiro, considera-se a existência de apenas 4 estados para $y(n)$, os estados -1, -0.5, 0 e 0.5. Calculando todos os valores possíveis de $y(n)$, obtém-se:

1) Se $y(n-1) = -1$:

$$y(n) = R\{Q[0.5(-1)] + Q[-0.5(-1)]\} = 0$$
$$y(n) = R\{Q[0.5(-0.5)] + Q[-0.5(-1)] = 0.5$$
$$y(n) = R\{Q[0.5(0)] + Q[-0.5(-1)]\} = 0.5$$
$$y(n) = R\{Q[0.5(0.5)] + Q[-0.5(-1)]\} = R\{1\} = 0.5$$

2) Se $y(n-1) = -0.5$:

$$y(n) = R\{Q[0.5(-1)] + Q[-0.5(-0.5)]\} = 0$$
$$y(n) = R\{Q[0.5(-0.5)] + Q[-0.5(-0.5)] = 0.5$$
$$y(n) = R\{Q[0.5(0)] + Q[-0.5(-0.5)]\} = 0.5$$
$$y(n) = R\{Q[0.5(0.5)] + Q[-0.5(-0.5)]\} = R\{1\} = 0.5$$

3) Se $y(n-1) = 0$:

$$y(n) = R\{Q[0.5(-1)] + Q[-0.5(0)]\} = -0.5$$
$$y(n) = R\{Q[0.5(-0.5)] + Q[-0.5(0)] = 0$$
$$y(n) = R\{Q[0.5(0)] + Q[-0.5(0)]\} = 0$$
$$y(n) = R\{Q[0.5(0.5)] + Q[-0.5(0)]\} = 0.5$$

4) Se $y(n-1) = 0.5$:

$$y(n) = R\{Q[0.5(-1)] + Q[-0.5(0.5)]\} = -0.5$$
$$y(n) = R\{Q[0.5(-0.5)] + Q[-0.5(0.5)] = 0$$
$$y(n) = R\{Q[0.5(0)] + Q[-0.5(0.5)]\} = 0$$
$$y(n) = R\{Q[0.5(0.5)] + Q[-0.5(0.5)]\} = 0.5.$$

Se forem usados os valores calculados para $y(n-1) = -1$, determinam-se as probabilidades da primeira coluna de $\mathbb{P}$,

$$\begin{cases} p_{11} &= P(y(n) = -1 | y(n-1) = -1) = 0 \\ p_{21} &= P(y(n) = -0.5 | y(n-1) = -1) = 0 \\ p_{31} &= P(y(n) = 0 | y(n-1) = -1) = 0 \\ p_{41} &= P(y(n) = 0.5 | y(n-1) = -1) \\ &= P(x(n) = -0.5) + P(x(n) = 0) + P(x(n) = 0.5) = 1. \end{cases}$$

Da mesma maneira, para as outras colunas observam-se os outros valores obtidos

para $y(n)$ e as probabilidades com relação à entrada,

$$\begin{cases} p_{12} &= P(y(n) = -1 | y(n-1) = -0.5) = 0 \\ p_{22} &= P(y(n) = -0.5 | y(n-1) = -0.5) = 0 \\ p_{32} &= P(y(n) = 0 | y(n-1) = -0.5) = 0 \\ p_{42} &= P(y(n) = 0.5 | y(n-1) = -0.5) \\ &= P(x(n) = -0.5) + P(x(n) = 0) + P(x(n) = 0.5) = 1, \end{cases}$$

$$\begin{cases} p_{13} &= P(y(n) = -1 | y(n-1) = 0) = 0 \\ p_{23} &= P(y(n) = -0.5 | y(n-1) = 0) = P(x(n) = -1) = 0 \\ p_{33} &= P(y(n) = 0 | y(n-1) = 0) = P(x(n) = -0.5) + P(x(n) = 0) = 0.667 \\ p_{43} &= P(y(n) = 0.5 | y(n-1) = 0) = P(x(n) = 0.5) = 0.333 \end{cases}$$

e

$$\begin{cases} p_{14} &= P(y(n) = -1 | y(n-1) = 0.5) = 0 \\ p_{24} &= P(y(n) = -0.5 | y(n-1) = 0.5) = P(x(n) = -1) = 0 \\ p_{34} &= P(y(n) = 0 | y(n-1) = 0.5) = P(x(n) = -0.5) + P(x(n) = 0) = 0.667 \\ p_{44} &= P(y(n) = 0.5 | y(n-1) = 0.5) = P(x(n) = 0.5) = 0.333. \end{cases}$$

Com isso, a matriz de transição de estados pode ser escrita como

$$\mathbb{P} = \begin{array}{c} \begin{matrix} -1.0 & -0.5 & 0 & 0.5 \end{matrix} \\ \left[\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.667 & 0.667 \\ 1.000 & 1.000 & 0.333 & 0.333 \end{matrix}\right] \end{array} \begin{matrix} \textbf{Estados} \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \end{matrix} ,$$

em que são fornecidos acima e à direita os estados da cadeia para facilitar a interpretação da matriz.

A matriz encontrada, conforme observado no cálculo das probabilidades, não diferencia os valores 0.5 obtidos naturalmente durante os cálculos daqueles que surgem por saturação. Se, por sua vez, as probabilidades dos valores saturados

forem separadas das outras, por meio dos estados -1_S e 0.5_S, a nova matriz de transição passa a ser descrita por

$$\mathbb{P} = \begin{array}{c} \begin{array}{cccccc} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{array} \quad \textbf{Estados} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.667 & 0.667 & 0.667 \\ 0.667 & 0.667 & 0.667 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0 & 0 & 0 \end{array}\right] \begin{array}{c} -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array} \end{array} .$$

Para a nova $\mathbb{P}$ encontrada, nota-se que as colunas correspondentes aos estados -1_S e -1 possuem a mesma distribuição de probabilidades, já que o resultado de $P(y(n)|y(n-1) = -1_S)$ deve ser igual a $P(y(n)|y(n-1) = -1)$, dado que a saída do filtro para $y(n-1) = -1$ ou $y(n-1) = -1_S$ é a mesma. Para as colunas dos estados 0.5 e 0.5_S o argumento é o mesmo, $P(y(n)|y(n-1) = 0.5) = P(y(n)|y(n-1) = 0.5_S)$.

Após esse exemplo inicial, na próxima seção são calculadas as probabilidades condicionadas de filtros de segunda ordem, considerando a não-linearidade de saturação. Para isso, o cálculo é feito da maneira mais geral possível, de forma que o resultado possa ser estendido para diversas densidades de probabilidade de entrada. O cálculo para filtros de primeira ordem é apresentado no final da seção.

4.2.1 Cálculo das probabilidades em um filtro de segunda ordem, considerando a não-linearidade de saturação

Nessa seção, inicialmente são calculadas as probabilidades associadas à entrada e suas versões passadas, sendo em seguida encontrada a probabilidade condicionada da saída com relação às saídas anteriores. Assume-se que os filtros são implementados com notação de complemento-a-dois e que, durante a quantização,

os valores são arredondados para cima.

4.2.1.1 Densidade de probabilidade da entrada

A entrada discreta do filtro $x(n)$ possui uma função densidade de probabilidade que pode ser descrita genericamente por

$$f_x(x(n)) = \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \delta(x(n) - k\Delta), \tag{4.3}$$

em que os coeficientes γ_k são as probabilidades para cada valor possível de $x(n) \in [-1, \quad (1-\Delta)]$, podendo representar os elementos de qualquer distribuição de probabilidade desejada.

Quando $-1 \leq b_0 x(n) \leq (1-\Delta)$, a probabilidade $P(Q\left[b_0 x(n)\right] = i\Delta)$, para i inteiro e $-2^{B-1} \leq i \leq 2^{B-1} - 1$, é calculada por

$$\begin{aligned} P(Q\left[b_0 x(n)\right] = i\Delta) &= P(i\Delta - 0.5\Delta \leq b_0 x(n) < i\Delta + 0.5\Delta) \\ &= P\left(\frac{i\Delta - 0.5\Delta}{b_0} \leq x(n) < \frac{i\Delta + 0.5\Delta}{b_0}\right) \end{aligned} \tag{4.4}$$

Fazendo $s_1 = \frac{i\Delta - 0.5\Delta}{b_0}$ e $s_2 = \frac{i\Delta + 0.5\Delta}{b_0}$, obtém-se

$$\begin{aligned} P(Q\left[b_0 x(n)\right] = i\Delta) &= \int_{s_1}^{s_2} \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \delta(x - k\Delta) dx \\ &= \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \int_{s_1}^{s_2} \delta(x - k\Delta) dx. \end{aligned} \tag{4.5}$$

Lembrando que $x(n)$ pode ser escrito como $x(n) = j\Delta$, para $-2^{B-1} \leq j \leq 2^{B-1} - 1$, pode-se calcular

$$\begin{aligned} P(Q\left[b_0 x(n)\right] = i\Delta) &= \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \int_{j=\frac{s_1}{\Delta}}^{\frac{s_2}{\Delta}} \delta(j\Delta - k\Delta) dj \\ &= \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \sum_{j=\lceil \frac{s_1}{\Delta} \rceil}^{\lfloor \frac{s_2}{\Delta} \rfloor} \delta_D[(j-k)\Delta], \end{aligned} \tag{4.6}$$

onde $\lceil a \rceil$ fornece o menor inteiro maior ou igual a a e $\lfloor a \rfloor$ fornece o maior inteiro

menor ou igual a a. $\delta_D[\cdot]$ é o delta de Kronecker, definido como

$$\delta_D[\alpha] = \begin{cases} 1, & \text{se } \alpha = 0 \\ 0, & \text{se } \alpha \neq 0. \end{cases}$$

Mas $b_0x(n)$ pode exceder os limites da representação, ou seja, ser menor que $(-1-\Delta/2)$ ou maior que $(1-\Delta/2)$. Se a não-linearidade de saturação for usada nos valores de $b_0x(n)$ que excedem o intervalo, quando $b_0x(n)$ for menor que -1 ou maior que $(1-\Delta)$, $Q[b_0x(n)]$ será igual a -1 e $(1-\Delta)$, respectivamente. Isso fará com que os valores saturados correspondam aos estados de -1 e $(1-\Delta)$ (vide figura 26, que exemplifica o efeito da saturação para uma implementação em 2 bits). Uma abordagem semelhante a essa é usada em [3, 4], onde cadeias

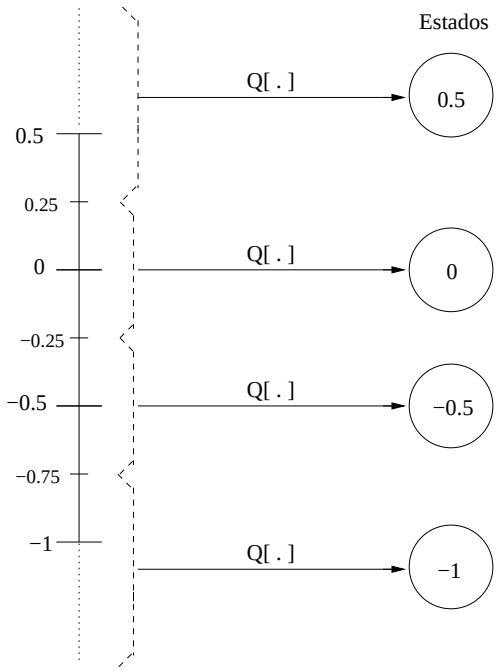

Figura 26: Não-linearidade de saturação para implementação de 2 bits

de Markov são aplicadas para estudar o algoritmo LMS. Ao invés de se fazer os valores saturados corresponderem aos mesmos estados de -1 e $(1-\Delta)$, pode-se definir dois novos estados para representar a ocorrência de saturação nos valores que excedem o limite (vide figura 27, que mostra um exemplo com 2 bits), de tal

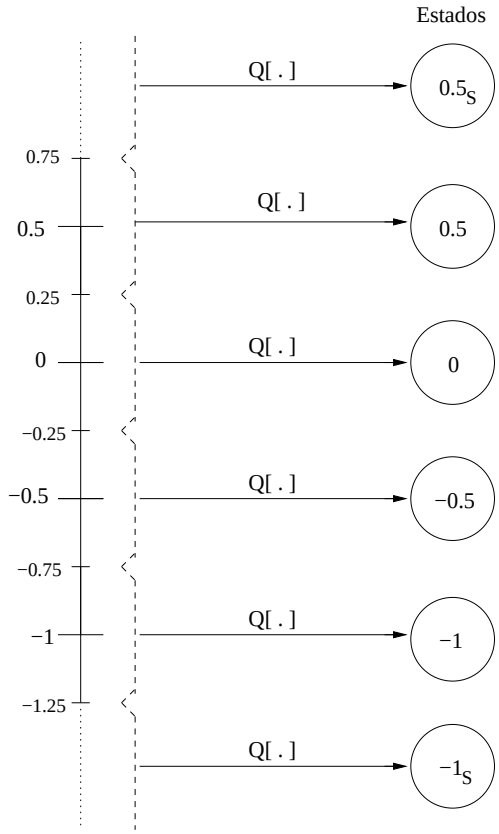

Figura 27: Saturação usando estados adicionais, para implementação com 2 bits. O estado 0.5_S corresponde a todos os valores maiores que 0.5, que são saturados nesse valor. Da mesma maneira, -1_S representa os valores menores que -1, saturados nesse valor.

forma que se pode calcular as probabilidades de saturação através de

$$\begin{aligned} P(b_0x(n) \geq (1-0.5\Delta)) &= P\left(\tfrac{1-0.5\Delta}{b_0} \leq x(n) < \infty\right) \\ &= \textstyle\sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \int_{\frac{1-0.5\Delta}{b_0}}^{\infty} \delta(x-k\Delta)dx. \end{aligned} \tag{4.7}$$

e

$$\begin{aligned} P(b_0x(n) < -1-0.5\Delta) &= P\left(-\infty < x(n) < \tfrac{-1-0.5\Delta}{b_0}\right) \\ &= \textstyle\sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \int_{-\infty}^{\frac{-1-0.5\Delta}{b_0}} \delta(x-k\Delta)dx \end{aligned} . \tag{4.8}$$

Mas, x pertence a um conjunto finito de valores e pode ser escrito como $x = j\Delta$, para j inteiro pertencente ao intervalo $[-2^{B-1},\ 2^{B-1}-1]$. Nesse caso, a integral (4.7) pode ser substituída pela somatória

$$P(b_0x(n) \geq (1-0.5\Delta)) = \textstyle\sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \sum_{j=\lceil \frac{1-0.5\Delta}{b_0\Delta} \rceil}^{\infty} \delta_D[(j-k)\Delta]. \tag{4.9}$$

De forma semelhante, é possível substituir (4.8) por

$$P(b_0x(n) < -1-0.5\Delta) = \textstyle\sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \sum_{j=-\infty}^{\lfloor \frac{1-0.5\Delta}{b_0\Delta} \rfloor} \delta_D[(j-k)\Delta], \tag{4.10}$$

em que $P(b_0x(n) \geq (1-0.5\Delta))$ e $P(b_0x(n) < -1-0.5\Delta)$ correspondem à probabilidade de saturação em -1 e $(1-\Delta)$, respectivamente. Para deixar visível a diferença entre os estados -1 e $(1-\Delta)$, pode-se denotar

$$x_0 = \begin{cases} (-1-\Delta), & \text{se } b_0x(n) < (-1-0.5\Delta) \\ Q[b_0x(n)], & \text{se } (-1-0.5\Delta) \leq b_0x(n) < (1-0.5\Delta) \\ 1, & \text{se } b_0x(n) \geq (1-0.5\Delta) \end{cases} , \tag{4.11}$$

onde $x_0 = (-1-\Delta)$ denota o estado em que ocorre saturação em -1 (ou seja, o estado -1_S) e $x_0 = 1$ denota o estado em que ocorre a saturação em $(1-\Delta)$ (ou seja, o estado $(1-\Delta)_S$). Note que os valores $(-1-\Delta)$ e $+1$ são usados apenas para simplificar a notação nas somatórias que seguem (como (4.13)); os valores de saída usados são sempre -1 e $(1-\Delta)$. Com isso, para usar uma notação

reduzida, pode-se definir

$$X_0^i = \begin{cases} P(x_0 = -1 - \Delta), \text{ se } i = -2^{B-1} - 1 \\ P(x_0 = i\Delta), \text{ se } -2^{B-1} \leq i \leq 2^{B-1} - 1 \\ P(x_0 = 1), \text{ se } i = 2^{B-1} \end{cases}, \tag{4.12}$$

que pode ser usada para escrever a f.d.p. de x_0 como

$$f_{x_0}(x_0) = \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^j \delta(x_0 - j\Delta). \tag{4.13}$$

De forma semelhante, definindo

$$x_1 = \begin{cases} (-1 - \Delta), \text{ se } b_1 x(n-1) < (-1 - 0.5\Delta) \\ Q[b_1 x(n-1)], \text{ se } (-1 - 0.5\Delta) \leq b_1 x(n-1) < (1 - 0.5\Delta) \\ 1, \text{ se } b_1 x(n-1) \geq (1 - 0.5\Delta) \end{cases} \tag{4.14}$$

e

$$X_1^i = \begin{cases} P(x_1 = -1 - \Delta), \text{ se } i = -2^{B-1} - 1 \\ P(x_1 = i\Delta), \text{ se } -2^{B-1} \leq i \leq 2^{B-1} - 1 \\ P(x_1 = 1), \text{ se } i = 2^{B-1} \end{cases}, \tag{4.15}$$

para o termo $b_1 x(n-1)$, e

$$x_2 = \begin{cases} (-1 - \Delta), \text{ se } b_2 x(n-2) < (-1 - 0.5\Delta) \\ Q[b_2 x(n-2)], \text{ se } (-1 - 0.5\Delta) \leq b_2 x(n-2) < (1 - 0.5\Delta) \\ 1, \text{ se } b_2 x(n-2) \geq (1 - 0.5\Delta) \end{cases}, \tag{4.16}$$

e

$$X_2^i = \begin{cases} P(x_2 = -1 - \Delta), \text{ se } i = -2^{B-1} - 1 \\ P(x_2 = i\Delta), \text{ se } -2^{B-1} \leq i \leq 2^{B-1} - 1 \\ P(x_2 = 1), \text{ se } i = 2^{B-1} \end{cases}, \tag{4.17}$$

para o termo $b_2 x(n-2)$. As densidades de probabilidade de x_1 e x_2 podem ser descritas como

$$f_{x_1}(x_1) = \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_1^j \delta(x_1 - j\Delta) \tag{4.18}$$

e

$$f_{x_2}(x_2) = \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_2^j \delta(x_2 - j\Delta). \tag{4.19}$$

Dessa forma, a probabilidade de obter -1 foi dividida entre a probabilidade de se obter -1 sem que haja saturação (quando x_0, x_1 ou $x_2 = -1$) e a probabilidade de obter -1 devido à saturação (quando x_0, x_1 ou $x_2 = (-1 - \Delta)$). Para $(1 - \Delta)$ pode-se pensar de forma semelhante.

4.2.1.2 Probabilidade de $R\{Q[b_1x(n-1)] + Q[b_2x(n-2)]\}$

Definindo $\bar{x}_{12} = Q[b_1x(n-1)] + Q[b_2x(n-2)]$ e assumindo que $x(n)$ e $x(n-1)$ são variáveis independentes, a densidade de probabilidade da soma das variáveis aleatórias $Q[b_1x(n-1)]$ e $Q[b_2x(n-2)]$ pode ser encontrada através da convolução de $f_{x_1}(x_1)$ e $f_{x_2}(x_2)$, isto é,

$$f_{\bar{x}_{12}}(\bar{x}_{12}) = \int_{-\infty}^{\infty} f_{x_1}(\alpha) f_{x_2}(\bar{x}_{12} - \alpha) d\alpha. \tag{4.20}$$

Com isso, obtém-se

$$\begin{aligned} & f_{\bar{x}_{12}}(\bar{x}_{12}) = \\ & = \int_{-\infty}^{\infty} \left(\sum_{i=-2^{B-1}-1}^{2^{(B-1)}} X_1^i \delta(\alpha - i\Delta) \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_2^j \delta(\bar{x}_{12} - \alpha - j\Delta) \right) d\alpha \\ & = \sum_{i=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^i X_2^j \int_{-\infty}^{\infty} \delta(\alpha - i\Delta) \delta(\bar{x}_{12} - \alpha - j\Delta) d\alpha \\ & = \sum_{i=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^i X_2^j \delta(\bar{x}_{12} - i\Delta - j\Delta), \end{aligned} \tag{4.21}$$

em que foi cometido um abuso de notação para escrever $\int_{-\infty}^{\infty} \delta(\alpha - i\Delta)\delta(\bar{x}_{12} - \alpha - j\Delta)d\alpha = \delta(\bar{x}_{12} - i\Delta - j\Delta)$.

Para obter a densidade de probabilidade de $R\{\bar{x}_{12}\}$, é necessário encontrar as probabilidades das saturações. Semelhante ao realizado na seção 4.2.1.1, é necessário integrar (4.21) para obter as probabilidades dos estados e definir uma nova função de probabilidade. As novas probabilidades podem ser calculadas

através de

$$P(R\{\bar{x}_{12}\} = i\Delta) = P(i\Delta - 0.5\Delta \leq \bar{x}_{12} < i\Delta + 0.5\Delta), \tag{4.22}$$

para $-2^{B-1} \leq i \leq 2^{B-1} - 1$.

Com isso,

$$\begin{aligned}
&P(R\{\bar{x}_{12}\} = i\Delta) = \\
&= \textstyle\int_{i\Delta-0.5\Delta}^{i\Delta+0.5\Delta} f_{\bar{x}_{12}}(\bar{x}_{12}) d\bar{x}_{12} \\
&= \textstyle\sum_{k=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^k X_2^j \int_{i\Delta-0.5\Delta}^{i\Delta+0.5\Delta} \delta(\bar{x}_{12} - k\Delta - j\Delta) d\bar{x}_{12} \\
&= \textstyle\sum_{k=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^k X_2^j \delta_D[i\Delta - k\Delta - j\Delta], \\
&\text{para} \quad -2^{B-1} \leq i \leq 2^{B-1} - 1.
\end{aligned} \tag{4.23}$$

Para calcular as probabilidades de saturação, ou seja, $\bar{x}_{12} \geq (1 - 0.5\Delta)$ e $\bar{x}_{12} < (-1 - 0.5\Delta)$, deve-se fazer

$$\begin{aligned}
&P(\bar{x}_{12} \geq (1 - 0.5\Delta)) = \\
&= P((1 - 0.5\Delta) \leq \bar{x}_{12} < \infty) \\
&= \textstyle\int_{(1-0.5\Delta)}^{\infty} f_{\bar{x}_{12}}(\bar{x}_{12}) dx_{12} \\
&= \textstyle\sum_{k=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^k X_2^j \int_{(1-0.5\Delta)}^{\infty} \delta(\bar{x}_{12} - k\Delta - j\Delta) d\bar{x}_{12} \\
&= \textstyle\sum_{k=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^k X_2^j \sum_{\bar{x}_{12}=\lceil(1-0.5\Delta)\rceil}^{\infty} \delta_D[\bar{x}_{12} - k\Delta - j\Delta] \\
&= \textstyle\sum_{k=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^k X_2^j \sum_{i=2^{(B-1)}}^{\infty} \delta_D[i\Delta - k\Delta - j\Delta]
\end{aligned} \tag{4.24}$$

e

$$\begin{aligned}
&P(\bar{x}_{12} < (-1 - 0.5\Delta)) = \\
&= P(-\infty < \bar{x}_{12} < (-1 - 0.5\Delta)) \\
&= \textstyle\int_{\bar{x}_{12}=-\infty}^{(-1-0.5\Delta)} f_{\bar{x}_{12}}(\bar{x}_{12}) d\bar{x}_{12} \\
&= \textstyle\sum_{k=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^k X_2^j \int_{\bar{x}_{12}=-\infty}^{(-1-0.5\Delta)} \delta(\bar{x}_{12} - k\Delta - j\Delta) d\bar{x}_{12} \\
&= \textstyle\sum_{k=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^k X_2^j \sum_{\bar{x}_{12}=-\infty}^{\lfloor(-1-0.5\Delta)\rfloor} \delta_D[\bar{x}_{12} - k\Delta - j\Delta] \\
&= \textstyle\sum_{k=-2^{B-1}-1}^{2^{(B-1)}} \sum_{j=-2^{B-1}-1}^{2^{(B-1)}} X_1^k X_2^j \sum_{i=-\infty}^{-2^{(B-1)}-1} \delta_D[i\Delta - k\Delta - j\Delta]
\end{aligned} \tag{4.25}$$

Se for definido x_{12} tal que

$$x_{12} = \begin{cases} (-1-\Delta), \text{ se } \bar{x}_{12} < -1 \\ \bar{x}_{12}, \text{ se } -1 \leq \bar{x}_{12} \leq (1-0.5\Delta) \\ 1, \text{ se } \bar{x}_{12} > (1-0.5\Delta) \end{cases}, \tag{4.26}$$

e for usado X_{12}^l,

$$X_{12}^l = \begin{cases} P(x_{12} = -1-\Delta), \text{ se } l = -2^{B-1}-1 \\ P(x_{12} = l\Delta), \text{ se } -2^{B-1} \leq l \leq 2^{B-1}-1 \\ P(x_{12} = 1), \text{ se } l = 2^{B-1} \end{cases}, \tag{4.27}$$

para simplificar a notação, pode-se escrever a densidade de probabilidade

$$f_{x_{12}}(x_{12}) = \sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{12}^l \delta(x_{12} - l\Delta). \tag{4.28}$$

4.2.1.3 Probabilidade de $R\{Q[b_0x(n)] + R\{Q[b_1x(n-1)] + Q[b_2x(n-2)]\}\}$

Semelhante ao usado na seção 4.2.1.2, calcula-se a densidade de probabilidade de $Q[b_0x(n)] + R\{Q[b_1x(n-1)] + Q[b_2x(n-2)]\} = \bar{x}_{012}$, assumindo que $x(n)$, $x(n-1)$ e $x(n-2)$ são variáveis independentes, através de

$$f_{\bar{x}_{012}}(\bar{x}_{012}) = \int_{-\infty}^{\infty} f_{x_0}(\alpha) f_{\bar{x}_{012}}(\bar{x}_{012} - \alpha) d\alpha. \tag{4.29}$$

Isso fornece

$$f_{\bar{x}_{012}}(\bar{x}_{012}) = \sum_{i=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^i X_{12}^j \delta(\bar{x}_{012} - i\Delta - j\Delta). \tag{4.30}$$

A função densidade de probabilidade de $x_{012} = R[\bar{x}_{012}]$, por sua vez, é calculada por

$$f_{x_{012}}(x_{012}) = \sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{012}^l \delta(x_{012} - l\Delta), \tag{4.31}$$

para x_{012} dado por

$$x_{012} = \begin{cases} (-1-\Delta), \ \text{ se } \ \bar{x}_{012} < -1 \\ \bar{x}_{012}, \ \text{ se } \ -1 \leq \bar{x}_{012} \leq (1-0.5\Delta) \\ 1, \ \text{ se } \ \bar{x}_{012} > (1-0.5\Delta) \end{cases}, \tag{4.32}$$

e onde os X_{012}^l são calculados como

$$X_{012}^l = \begin{cases} P(x_{012} = -1-\Delta), \ \text{ se } \ l = -2^{B-1}-1 \\ P(x_{012} = l\Delta), \ \text{ se } \ -2^{B-1} \leq l \leq 2^{B-1}-1 \\ P(x_{012} = 1), \ \text{ se } \ l = 2^{B-1} \end{cases}, \tag{4.33}$$

com

$$\begin{aligned} & P(x_{012} = l\Delta) = \\ & = P(\bar{x}_{012} = l\Delta) \\ & = \textstyle\int_{l\Delta-0.5\Delta}^{l\Delta+0.5\Delta} f_{\bar{x}_{012}}(\bar{x}_{012}) d\bar{x}_{012} \\ & = \textstyle\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_{12}^j \int_{l\Delta-0.5\Delta}^{l\Delta+0.5\Delta} \delta(\bar{x}_{012} - k\Delta - j\Delta) d\bar{x}_{012} \\ & = \textstyle\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_{012}^j \delta_D[(l-k-j)\Delta], \end{aligned} \tag{4.34}$$

para $-2^{B-1} \leq l \leq 2^{B-1}-1$, e

$$\begin{aligned} & P(x_{012} = 1) = \\ & = P((1-0.5\Delta) \leq \bar{x}_{012} < \infty) \\ & = \textstyle\int_{(1-0.5\Delta)}^{\infty} f_{\bar{x}_{012}}(\bar{x}_{012}) d\bar{x}_{012} \\ & = \textstyle\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_{012}^j \int_{(1-0.5\Delta)}^{\infty} \delta(\bar{x}_{012} - k\Delta - j\Delta) d\bar{x}_{012} \\ & = \textstyle\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_{012}^j \sum_{\bar{x}_{012}=\lceil(1-0.5\Delta)\rceil}^{\infty} \delta_D[\bar{x}_{012} - k\Delta - j\Delta] \\ & = \textstyle\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_{012}^j \sum_{i=2^B-2}^{\infty} \delta_D[(i-k-j)\Delta] \end{aligned} \tag{4.35}$$

e

$$
\begin{aligned}
&P(x_{012} = -1 - \Delta) = \\
&= P(-\infty < \bar{x}_{012} < (-1 - 0.5\Delta)) \\
&= \textstyle\int_{\bar{x}_{012}=-\infty}^{(-1-0.5\Delta)} f_{\bar{x}_{012}}(\bar{x}_{012}) d\bar{x}_{012} \\
&= \textstyle\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_{012}^j \int_{\bar{x}_{012}=-\infty}^{(-1-0.5\Delta)} \delta(\bar{x}_{012} - k\Delta - j\Delta) d\bar{x}_{012} \\
&= \textstyle\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_{012}^j \sum_{\bar{x}_{012}=-\infty}^{\lfloor(-1-0.5\Delta)\rfloor} \delta_D[\bar{x}_{012} - k\Delta - j\Delta] \\
&= \textstyle\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_{012}^j \sum_{i=-\infty}^{-2^{(B-1)}-1} \delta_D[(i-k-j)\Delta].
\end{aligned}
\tag{4.36}
$$

Com isso, a probabilidade de $y(n)$ pode ser calculada em função da entrada $x(n)$ e de suas versões atrasadas $x(n-1)$ e $x(n-2)$.

4.2.1.4 Densidade de probabilidade condicionada de $y(n)$

A probabilidade de a saída assumir algum valor, considerando as saídas passadas, é dada por

$$
P\left(y(n) = k\Delta | y(n-1) = i\Delta, y(n-2) = j\Delta\right), \tag{4.37}
$$

com i, j, $k \in [-2^{B-1} - 1,\ 2^{B-1}]$.

Renomeando $R\{Q[-a_1 y(n-1)] + Q[-a_2 y(n-2)]\} = Y^{ij}$, a probabilidade condicionada de $y(n)$ com relação a $y(n-1)$ e $y(n-2)$ pode ser escrita como

$$
\begin{aligned}
&P\left(y(n) = k\Delta | y(n-1) = i\Delta, y(n-2) = j\Delta\right) = \\
&\qquad = P\left(R\left[x_{012} + Y^{ij}\right] = k\Delta | y(n-1) = i\Delta, y(n-2) = j\Delta\right) \\
&= P\left(k\Delta - 0.5\Delta - Y^{ij} \leq x_{012} < k\Delta + 0.5\Delta - Y^{ij} | y(n-1) = i\Delta, y(n-2) = j\Delta\right).
\end{aligned}
\tag{4.38}
$$

Usando a equação (4.31) e integrando,

$$\begin{aligned} &P\left(y(n)=k\Delta|y(n-1)=i\Delta, y(n-2)=j\Delta\right)= \\ &=\textstyle\int_{k\Delta-0.5\Delta-Y^{ij}}^{k\Delta+0.5\Delta-Y^{ij}} \sum_{l=-2^{B-1}-1}^{2^B-2} X_{012}^l \delta(\alpha-l\Delta)d\alpha \\ &=\textstyle\sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{012}^l \sum_{m=\lceil k\Delta-0.5\Delta-Y^{ij}\rceil}^{\lfloor k\Delta+0.5\Delta-Y^{ij}\rfloor} \delta_D[m-l\Delta] \\ &=\textstyle\sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{012}^l \delta_D[k\Delta-Y^{ij}-l\Delta], \end{aligned} \tag{4.39}$$

se $-2^{B-1} \leq k \leq 2^{B-1}-1$. Quando ocorre a saturação positiva, $(x_{012}+[Y]_{ij}) \geq (1-0.5\Delta)$, tem-se

$$\begin{aligned} &P\left(x_{012}+Y^{ij} \geq (1-0.5\Delta)|y(n-1)=i\Delta, y(n-2)=j\Delta\right)= \\ &=P\left((1-0.5\Delta)-Y^{ij} \leq x_{012} < \infty|y(n-1)=i\Delta, y(n-2)=j\Delta\right) \\ &=\textstyle\int_{(1-0.5\Delta)-Y^{ij}}^{\infty} \sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{012}^l \delta(\alpha-l\Delta)d\alpha \\ &=\textstyle\sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{012}^l \sum_{m=\lceil (1-0.5\Delta)-[Y]_{ij}\rceil}^{\infty} \delta_D[m-l\Delta]. \end{aligned} \tag{4.40}$$

Por outro lado, quando há a saturação negativa, $(x_{012}+Y^{ij}) < (-1-0.5\Delta)$, e a probabilidade é calculada por

$$\begin{aligned} &P\left(x_{012}+Y^{ij} < (-1-0.5\Delta)|y(n-1)=i\Delta, y(n-2)=j\Delta\right)= \\ &=P\left(-\infty < x_{012} < (-1-0.5\Delta)-Y^{ij}|y(n-1)=i\Delta, y(n-2)=j\Delta\right) \\ &=\textstyle\int_{-\infty}^{(-1-0.5\Delta)-Y^{ij}} \sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{012}^l \delta(\alpha-l\Delta)d\alpha \\ &=\textstyle\sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{012}^l \sum_{m=-\infty}^{\lfloor (-1-0.5\Delta)-Y^{ij}\rfloor} \delta_D[m-l\Delta]. \end{aligned} \tag{4.41}$$

Definindo $y(n)$ como

$$y(n)=\begin{cases} (-1-\Delta), \text{ se } (x_{012}+Y^{ij}) < (-1-0.5\Delta) \\ Q[y(n)], \text{ se } (-1-0.5\Delta) \leq (x_{012}+Y^{ij}) < (1-0.5\Delta) \\ 1, \text{ se } (x_{012}+Y^{ij}) \geq (1-0.5\Delta) \end{cases} \tag{4.42}$$

e

$$Z_k^{ij}=\begin{cases} P\left(x_{012}+Y^{ij} < (-1-0.5\Delta)|i\Delta, j\Delta\right) \text{ se } k \leq -2^{(B-1)-1} \\ P\left(y(n)=k\Delta|i\Delta, j\Delta\right), \text{ se } -2^{(B-1)} \leq k \leq 2^{(B-1)}-1 \\ P\left(x_{012}+Y^{ij} \geq (1-0.5\Delta)|i\Delta, j\Delta\right), \text{ se } k \geq 2^{(B-1)} \end{cases}, \tag{4.43}$$

a função densidade de probabilidade condicionada de $y(n)$ é dada por

$$f_y(y(n)|y(n-1), y(n-2)) = \sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} Z_k^{ij}\delta(y(n) - k\Delta). \tag{4.44}$$

Portanto, seguindo essa lista de passos, pode-se encontrar as probabilidades condicionadas de $y(n)$ a partir da probabilidade da entrada dos filtros digitais e das saídas anteriores $y(n-1)$ e $y(n-2)$.

4.2.2 Probabilidades em um filtro de primeira ordem

Em um filtro de primeira ordem, o cálculo da função densidade de probabilidade de $x_{01} = R\{Q[b_0x(n)] + Q[b_1x(n-1)]\}$ é semelhante ao cálculo de x_{12} da seção 4.2.1.2, fornecendo

$$f_{x_{01}}(x_{01}) = \sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{01}^l\delta(x_{01} - l\Delta), \tag{4.45}$$

com

$$X_{01}^l = \begin{cases} \sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_1^j \delta_D[(l-k-j)\Delta], \\ \text{para} \quad -2^{B-1} \leq l \leq 2^{B-1} - 1 \\ \sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_1^j \sum_{i=-\infty}^{-2^{(B-1)}-1} \delta_D[(i-k-j)\Delta], \\ \text{para} \;\; l = -2^{B-1} - 1 \\ \sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} \sum_{j=-2^{(B-1)}-1}^{2^{(B-1)}} X_0^k X_1^j \sum_{i=2^{(B-1)}}^{\infty} \delta_D[(i-k-j)\Delta], \\ \text{para} \;\; l = 2^{B-1}. \end{cases}$$

e

$$x_{01} = \begin{cases} (-1-\Delta), \;\; \text{se} \;\; \bar{x}_{01} < (-1-0.5\Delta) \\ Q[\bar{x}_{01}], \;\; \text{se} \;\; (-1-0.5\Delta) \leq \bar{x}_{01} < (1-0.5\Delta) \\ 1, \;\; \text{se} \;\; \bar{x}_{01} \geq (1-0.5\Delta). \end{cases}$$

Com isso e fazendo $Y^i = Q[a_1y(n-1)]$, a probabilidade de $y(n)$ é dada por

$$P\left(y(n) = k\Delta | y(n-1) = i\Delta\right)$$

e a $f_y(y(n)|y(n-1))$ será

$$f_y(y(n)=k\Delta|y(n-1)=i\Delta)=\sum_{k=-2^{(B-1)}-1}^{2^{(B-1)}} Z_k^i\delta(y(n)-k\Delta), \tag{4.46}$$

com

$$Z_k^i=\begin{cases}\sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{01}^l\delta_D[k\Delta-Y^i-l\Delta],\\ \text{se } \ -2^{B-1}\leq k\leq 2^{B-1}-1\\ \sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{01}^l\sum_{m=\lceil(1-0.5\Delta)-Y^i\rceil}^{\infty}\delta_D[m-l\Delta],\\ \text{se } \ k=2^{(B-1)}\\ \sum_{l=-2^{(B-1)}-1}^{2^{(B-1)}} X_{01}^l\sum_{m=-\infty}^{\lfloor(-1-0.5\Delta)-[Y]_i\rfloor}\delta_D[m-l\Delta],\\ \text{se } \ k=-2^{B-1}-1.\end{cases}$$

e

$$y(n)=\begin{cases}(-1-\Delta), \ \text{ se } \ (x_{01}+Y^i)<(-1-0.5\Delta)\\ Q[y(n)], \ \text{ se } \ (-1-0.5\Delta)\leq(x_{01}+Y^i)<(1-0.5\Delta)\\ 1, \ \text{ se } \ (x_{01}+Y^i)\geq(1-0.5\Delta).\end{cases}$$

4.3 Cálculo da probabilidade de *overflow* usando cadeias de Markov

A aplicação da não-linearidade de *overflow* é uma outra forma de lidar com os valores que excedem os limites da representação e é naturalmente encontrada em notações em complemento-a-dois (vide exemplo da pág. 52). Nesse caso, filtros digitais de primeira e segunda ordem mantêm suas matrizes de transição de estados com dimensões $N\times N$ e $N^2\times N^2$, respectivamente, dado que não são adicionados estados de saturação. A seguir é apresentado um exemplo.

Exemplo: Suponha um filtro IIR de primeira ordem em que os sinais são representados com 2 bits (ou seja, o conjunto de números com que os sinais podem ser representados é $\{-1,\ -0.5,\ 0,\ 0.5\}$) e os coeficientes são representados com

3 bits, descrito pela equação

$$H(z) = \frac{1}{1 + 0.75z^{-1}}.$$

Para esse filtro, a equação de diferenças equivalente corresponde a

$$y(n) = R\{Q[x(n)] + Q[-0.75y(n-1)]\}.$$

Deseja-se calcular a matriz de transição de estados $\mathbb{P}$ quando a entrada possui f.d.p. uniforme e de média nula (ou seja, a probabilidade de $x(n) = -0.5$, 0 ou 0.5 é igual a 1/3 e a probabilidade de $x(n) = -1$ é igual a 0). Assume-se que os números são arredondados para cima, ou seja, $Q[0.25] = 0.5$ e $Q[-0.25] = 0$.

Calculando todos os valores possíveis para $y(n)$, obtém-se

1) Se $y(n-1) = -1$:

$$y(n) = R\{Q[-1] + Q[-0.75(-1)]\} = R\{-2\} = 0$$
$$y(n) = R\{Q[-0.5] + Q[-0.75(-1)] = R\{-1.5\} = 0.5$$
$$y(n) = R\{Q[0] + Q[-0.75(-1)]\} = -1$$
$$y(n) = R\{Q[0.5] + Q[-0.75(-1)]\} = -0.5$$

2) Se $y(n-1) = -0.5$:

$$y(n) = R\{Q[-1] + Q[-0.75(-0.5)]\} = -0.5$$
$$y(n) = R\{Q[-0.5] + Q[-0.75(-0.5)] = 0$$
$$y(n) = R\{Q[0] + Q[-0.75(-0.5)]\} = 0.5$$
$$y(n) = R\{Q[0.5] + Q[-0.75(-0.5)]\} = R\{1\} = -1$$

3) Se $y(n-1) = 0$:

$$y(n) = R\{Q[-1] + Q[-0.75(0)]\} = -1$$
$$y(n) = R\{Q[-0.5] + Q[-0.75(0)] = -0.5$$
$$y(n) = R\{Q[0] + Q[-0.75(0)]\} = 0$$
$$y(n) = R\{Q[0.5] + Q[-0.75(0)]\} = 0.5$$

4) Se $y(n-1) = 0.5$:

$$y(n) = R\{Q[-1] + Q[-0.75(0.5)]\} = R\{-1.5\} = 0.5$$
$$y(n) = R\{Q[-0.5] + Q[-0.75(0.5)] = -1$$
$$y(n) = R\{Q[0] + Q[-0.75(0.5)]\} = -0.5$$
$$y(n) = R\{Q[0.5] + Q[-0.75(0.5)]\} = 0$$

Se forem usados os valores calculados para $y(n-1) = -1$, determinam-se as probabilidades da primeira coluna de $\mathbb{P}$,

$$\begin{cases} p_{11} &= P(y(n) = -1 | y(n-1) = -1) = P(x(n) = 0) = 0.333 \\ p_{21} &= P(y(n) = -0.5 | y(n-1) = -1) = P(x(n) = 0.5) = 0.333 \\ p_{31} &= P(y(n) = 0 | y(n-1) = -1) = P(x(n) = -1) = 0 \\ p_{41} &= P(y(n) = 0.5 | y(n-1) = -1) = P(x(n) = -0.5) = 0.333 \end{cases}$$

Da mesma maneira, para as outras colunas observam-se os outros valores obtidos para $y(n)$ e as probabilidades com relação à entrada,

$$\begin{cases} p_{12} &= P(y(n) = -1 | y(n-1) = -0.5) = P(x(n) = 0.5) = 0.333 \\ p_{22} &= P(y(n) = -0.5 | y(n-1) = -0.5) = P(x(n) = -1) = 0 \\ p_{32} &= P(y(n) = 0 | y(n-1) = -0.5) = P(x(n) = -0.5) = 0.333 \\ p_{42} &= P(y(n) = 0.5 | y(n-1) = -0.5) = P(x(n) = 0) = 0.333 \end{cases}$$

$$\begin{cases} p_{13} &= P(y(n) = -1 | y(n-1) = -0.5) = P(x(n) = -1) = 0 \\ p_{23} &= P(y(n) = -0.5 | y(n-1) = -0.5) = P(x(n) = -0.5) = 0.333 \\ p_{33} &= P(y(n) = 0 | y(n-1) = -0.5) = P(x(n) = 0) = 0.333 \\ p_{43} &= P(y(n) = 0.5 | y(n-1) = -0.5) = P(x(n) = 0.5) = 0.333 \end{cases}$$

e

$$\begin{cases} p_{14} &= P(y(n) = -1 | y(n-1) = -0.5) = P(x(n) = -0.5) = 0.333 \\ p_{24} &= P(y(n) = -0.5 | y(n-1) = -0.5) = P(x(n) = 0) = 0.333 \\ p_{34} &= P(y(n) = 0 | y(n-1) = -0.5) = P(x(n) = 0.5) = 0.333 \\ p_{44} &= P(y(n) = 0.5 | y(n-1) = -0.5) = P(x(n) = -1) = 0. \end{cases}$$

Com isso, a matriz $\mathbb{P}$ é definida por

$$\mathbb{P} = \begin{array}{c} \begin{matrix} -1.0 & -0.5 & 0 & 0.5 \end{matrix} \quad \textbf{Estados} \\ \begin{bmatrix} 0.333 & 0.333 & 0 & 0.333 \\ 0.333 & 0 & 0.333 & 0.333 \\ 0 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0 \end{bmatrix} \begin{matrix} -1.0 \\ -0.5 \\ 0 \\ 0.5 \end{matrix} \end{array},$$

o que encerra este exemplo.

Na seção seguinte, calculam-se as probabilidades condicionadas da saída para filtros sujeitos à não-linearidade de *overflow*. Tal como para a saturação, a probabilidade da entrada será escrita de forma geral, para que as equações encontradas possam ser estendidas para diversas funções de densidade de probabilidade.

4.3.1 Probabilidades em um filtro de segunda ordem

4.3.1.1 Cálculo da densidade de probabilidade da entrada

Quando a não-linearidade usada na implementação é o *overflow*, ela pode ser descrita matematicamente como a função

$$R[\alpha] = \frac{1}{2^{(B-1)}} \left[(\lceil 2^{(B-1)}\alpha - 0.5 \rceil + 2^{(B-1)}) \bmod 2^B \right] - 1, \tag{4.47}$$

onde B é o número de bits usado e $b \bmod a$ corresponde ao resto da divisão de b por a, com o mesmo sinal do divisor a. Dessa função e da figura 19 (vide pág. 53), nota-se que a quantização por *overflow* para valores dentro do conjunto $[-1,\ \ (1-\Delta)]$ se repete, ou seja, a figura 19 entre os valores -1 e $(1-\Delta)$ se repete ao longo do eixo tanto para números positivos quanto para números negativos de forma cíclica e com um período igual a 2.

Exemplo: Para uma quantização por *overflow* de 3 bits, encontrar as quantizações para -3, -1, 1 e 3.

Nesse caso, pode-se olhar a figura 19 ou então calcular pela equação (4.47):

$$\begin{aligned}
R[-3] &= \frac{1}{4} \left[(\lceil 4 \times (-3) - 0.5 \rceil + 4) \bmod 8 \right] - 1 = -1 \\
R[-1] &= \frac{1}{4} \left[(\lceil 4 \times (-1) - 0.5 \rceil + 4) \bmod 8 \right] - 1 = -1 \\
R[1] &= \frac{1}{4} \left[(\lceil 4 \times (1) - 0.5 \rceil + 4) \bmod 8 \right] - 1 = -1 \\
R[3] &= \frac{1}{4} \left[(\lceil 4 \times (1) - 0.5 \rceil + 4) \bmod 8 \right] - 1 = -1,
\end{aligned} \tag{4.48}$$

de forma que a quantização de qualquer número α escolhido como $\alpha = -1 + 2i$, $i \in \mathbb{Z}$ será igual a -1. Ou seja: tomando um valor $\theta \in [-1,\ \ (1-\Delta)]$ qualquer da faixa, todos os números da forma $\theta + 2i$, $i \in \mathbb{Z}$, serão representados por *overflow* como θ.

Supondo que $x(n)$ possui uma f.d.p. descrita pela equação (4.3) (pág. 72),

para calcular a probabilidade de $Q[b_0x(n)] = x_0$, basta encontrar

$$P(x_0 = i\Delta) = P(Q[b_0x(n)] = i\Delta) = \sum_{j=-\infty}^{\infty} P\left(b_0x(n) = i\Delta + 2j\right), \tag{4.49}$$

para $-2^{(B-1)} \leq i \leq 2^{(B-1)} - 1$. Com isso, pode-se calcular

$$\begin{aligned}
&P(x_0 = i\Delta) = \\
&= \textstyle\sum_{j=-\infty}^{\infty} P(i\Delta + 2j - 0.5\Delta \leq b_0x(n) < i\Delta + 2j + 0.5\Delta) \\
&= \textstyle\sum_{j=-\infty}^{\infty} P(\frac{i\Delta+2j-0.5\Delta}{b_0} \leq x(n) < \frac{i\Delta+2j+0.5\Delta}{b0}) \\
&= \textstyle\sum_{j=-\infty}^{\infty} \int_{\frac{i\Delta+2j-0.5\Delta}{b0}}^{\frac{i\Delta+2j+0.5\Delta}{b0}} \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k\delta(\alpha - k\Delta)d\alpha \\
&= \textstyle\sum_{j=-\infty}^{\infty} \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \sum_{\alpha=\lceil\frac{i\Delta+2j-0.5\Delta\rceil}{b0}}^{\lfloor\frac{i\Delta+2j+0.5\Delta}{b0}\rfloor} \delta_D[\alpha - k\Delta].
\end{aligned} \tag{4.50}$$

Definindo $X_0^i = P(x_0 = i\Delta)$, para $-2^{B-1} \leq i \leq 2^{B-1} - 1$, a função densidade de probabilidade de x_0 é calculada por

$$f_{x_0}(x_0) = \sum_{i=-2^{B-1}}^{2^{B-1}-1} X_0^i\delta(x_0 - i\Delta). \tag{4.51}$$

Da mesma forma, é possível calcular as funções de probabilidade dos termos $x_1 = Q[b_1x(n-1)]$ e $x_2 = Q[b_2x(n-2)]$, isto é,

$$f_{x_1}(x_1) = \sum_{i=-2^{B-1}}^{2^{B-1}-1} X_1^i\delta(x_1 - i\Delta) \tag{4.52}$$

e

$$f_{x_2}(x_2) = \sum_{i=-2^{B-1}}^{2^{B-1}-1} X_2^i\delta(x_2 - i\Delta), \tag{4.53}$$

respectivamente, em que $X_1^i = P(x_1 = i\Delta)$ e $X_2^i = P(x_2 = i\Delta)$ são calculados de forma semelhante à equação (4.50).

4.3.1.2 Probabilidade de $R\{Q[b_1x(n-1)] + Q[b_2x(n-2)]\}$

Assumindo que $x(n-1)$ e $x(n-2)$ são variáveis independentes e definindo $\bar{x}_{12} = Q[b_1x(n-1)] + Q[b_2x(n-2)]$, a partir da convolução das f.d.p. de x_1 e x_2,

obtém-se

$$\begin{aligned} & f_{\bar{x}_{12}}(\bar{x}_{12}) = \\ & = \int_{-\infty}^{\infty} f_{x_1}(\alpha) f_{x_2}(x_{12}-\alpha) d\alpha \\ & = \int_{-\infty}^{\infty} \sum_{i=-2^{B-1}}^{2^{B-1}-1} X_1^i \delta(\alpha - i\Delta) \sum_{j=-2^{B-1}}^{2^{B-1}-1} X_2^j \delta(x_{12}-\alpha-j\Delta) d\alpha \\ & = \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} X_1^i X_2^j \delta_D[x_{12}-i\Delta-j\Delta]. \end{aligned} \tag{4.54}$$

Se $x_{12} = R\{\bar{x}_{12}\}$, para encontrar a probabilidade de $P(x_{12} = l\Delta)$ (para $-2^{B-1} \leq l \leq 2^{B-1}-1$), é necessário somar todos as possibilidades, considerando o *overflow*:

$$\begin{aligned} & P(x_{12} = l\Delta) = \\ & = \sum_{k=-\infty}^{\infty} P(l\Delta + 2k - 0.5\Delta \leq x_{12} \leq l\Delta + 2k + 0.5\Delta) \\ & = \sum_{k=-\infty}^{\infty} \int_{l\Delta+2k-0.5\Delta}^{l\Delta+2k+0.5\Delta} \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} X_1^i X_2^j \delta(\alpha - i\Delta - j\Delta) d\alpha \\ & = \sum_{k=-\infty}^{\infty} \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} X_1^i X_2^j \int_{l\Delta+2k-0.5\Delta}^{l\Delta+2k-0.5\Delta} \delta(\alpha - i\Delta - j\Delta) d\alpha \\ & = \sum_{k=-\infty}^{\infty} \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} X_1^i X_2^j \delta_D[l\Delta + 2k - i\Delta - j\Delta], \end{aligned} \tag{4.55}$$

que pode ser definido como $X_{12}^l = P(x_{12} = l\Delta)$, para $-2^{B-1} \leq l \leq 2^{B-1}-1$, para reduzir a notação. Na prática, os valores de *overflow* que aparecem após a soma $Q[b_1 x(n-1)] + Q[b_2 x(n-2)]$ limitam-se ao intervalo $[-2, (2-2\Delta)]$. Portanto, a somatória que leva em conta os valores excedentes devido ao *overflow* pode usar apenas o intervalo de -2 a 2 no cálculo de X_{12}^l, o que leva a

$$X_{12}^l = \sum_{k=-2}^{2} \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} X_1^i X_2^j \delta_D[l\Delta + 2k - i\Delta - j\Delta], \tag{4.56}$$

para $-2^{B-1} \leq l \leq 2^{B-1} - 1$.

A função de densidade de probabilidade fica definida como

$$f_{x_{12}}(x_{12}) = \sum_{l=-2^{B-1}}^{2^{B-1}-1} X_{12}^l \delta(x_{12} - l\Delta) \tag{4.57}$$

4.3.1.3 Probabilidade de $R\{Q[b_0x(n)] + R\{Q[b_1x(n-1)] + Q[b_2x(n-2)]\}\}$

Aplicando a mesma técnica da seção anterior, por meio da convolução de f_{x_0} com $f_{x_{12}}$ obtém-se a $f_{x_{012}}$,

$$f_{x_{012}}(x_{012}) = \sum_{l=-2^{B-1}}^{2^{B-1}-1} X_{012}^{l}\delta(x_{012} - l\Delta), \tag{4.58}$$

em que $x_{012} = R\{Q[b_0x(n)] + R\{Q[b_1x(n-1)] + Q[b_2x(n-2)]\}\}$ e X_{012}^{l} é dado por

$$X_{012}^{l} = \sum_{k=-2}^{2} \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} X_0^i X_{12}^j \delta_D[l\Delta + 2k - i\Delta - j\Delta]. \tag{4.59}$$

4.3.1.4 Probabilidade condicionada de $y(n)$

A probabilidade condicionada de $y(n)$ é obtida por

$$P(y(n) = l\Delta | y(n-1) = i\Delta, y(n-2) = j\Delta). \tag{4.60}$$

Se for definido $Y^{ij} = R\{Q[-a_1y(n-1)] + Q[-a_2y(n-2)]\}$, a probabilidade condicionada da saída será

$$\begin{aligned}
&P(y(n) = l\Delta | y(n-1) = i\Delta, y(n-2) = j\Delta) = \\
&= \textstyle\sum_{k=-\infty}^{\infty} P(l\Delta - 0.5\Delta + 2k \leq y(n) < l\Delta + 0.5\Delta + 2k | i\Delta, j\Delta) \\
&= P(l\Delta - 0.5\Delta + 2k - Y^{ij} \leq x_{012} < l\Delta + 0.5\Delta + 2k - Y^{ij} | i\Delta, j\Delta) \\
&= \textstyle\sum_{k=-\infty}^{\infty} \int_{l\Delta-0.5\Delta+2k-Y^{ij}}^{l\Delta+0.5\Delta+2k-Y^{ij}} \sum_{m=-2^{B-1}}^{2^{B-1}-1} X_{012}^{m}\delta(x_{012} - m\Delta)dx_{012} \\
&= \textstyle\sum_{k=-\infty}^{\infty} \sum_{m=-2^{B-1}}^{2^{B-1}-1} X_{012}^{m}\delta_D[l\Delta + 2k - Y^{ij} - m\Delta],
\end{aligned} \tag{4.61}$$

para $-2^{B-1} - 1 \leq l \leq 2^{B-1}$. Como $-2 \leq (Y^{ij} + x_{012}) \leq (2 - 2\Delta)$, pode-se limitar k de -2 até 2. Usando isso e fazendo $Z_l^{ij} = P(y(n) = l\Delta | y(n-1) = i\Delta, y(n-2) = j\Delta)$,

$$Z_l^{ij} = \sum_{k=-2}^{2} \sum_{m=-2^{B-1}}^{2^{B-1}-1} X_{012}^{m}\delta_D[l\Delta + 2k - Y^{ij} - m\Delta]. \tag{4.62}$$

Dessa forma, pode-se escrever a função de probabilidade $f_y(y(n)|y(n-1), y(n-2))$ como

$$f_y(y(n)|y(n-1) = i\Delta, y(n-2) = j\Delta) = \sum_{l=-2^{B-1}}^{2^{B-1}-1} Z_l^{ij}\delta(y(n) - l\Delta), \tag{4.63}$$

para $-2^{B-1} \leq l \leq 2^{B-1} - 1$.

4.3.2 Probabilidades em um filtro de primeira ordem

Para um filtro de primeira ordem com a não-linearidade de *overflow*, pode-se usar a mesma abordagem da seção 4.3.1.2 para calcular a densidade de probabilidade de $R\{Q[b_0x(n)] + Q[b_1x(n-1)]\}$. Definindo $x_{01} = R\{Q[b_0x(n)] + Q[b_1x(n-1)]\}$, obtém-se

$$f_{x_{01}}(x_{01}) = \sum_{l=-2^{B-1}}^{2^{B-1}-1} X_{01}^l\delta(x_{01} - l\Delta) \tag{4.64}$$

com

$$X_{01}^l = \sum_{k=-\infty}^{\infty} \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} X_0^i X_1^j \delta_D[l\Delta + 2k - i\Delta - j\Delta], \tag{4.65}$$

para $-2^{B-1} \leq l \leq 2^{B-1}-1$. Com isso, fazendo $Y^i = Q[a_1y(n-1)]$, a probabilidade condicionada de $y(n)$, dado $y(n-1)$, pode ser calculada como

$$f_y(y(n)|y(n-1)) = \sum_{l=-2^{B-1}}^{2^{B-1}-1} Z_l^i\delta(y(n) - l\Delta), \tag{4.66}$$

com

$$Z_l^i = \sum_{k=-2}^{2} \sum_{m=-2^{B-1}}^{2^{B-1}-1} X_{01}^m \delta_D[l\Delta + 2k - Y^i - m\Delta], \tag{4.67}$$

para $-2^{B-1} \leq l \leq 2^{B-1} - 1$.

4.4 Análise usando não-linearidade de saturação

4.4.1 Escalamento da entrada

A partir do uso da matriz de transição estendida com os estados saturados, fica visível o efeito da saturação na saída de um filtro digital, indicando que o escalamento do sinal de entrada é necessário para minimizar a distorção devido à saturação. Uma abordagem tradicional para escalar a entrada é o uso de normas L_p, conforme apresentado na seção 3.2.5 (pág. 53). Contudo, usando-se $\mathbb{P}$ e $\mathbb{P}^{\infty}$, pode-se determinar um fator de escalamento iterativamente, o que pode fornecer resultados menos conservadores que as normas L_p. Como encontrar esse fator de escala é apresentado a seguir, com o auxílio de rotinas desenvolvidas em *Matlab* e tendo por base as equações desenvolvidas na Seção 4.2.

Exemplo 1: Considere um filtro IIR de primeira ordem de equação

$$y(n) = R\{Q[x(n)] + Q[-0.75y(n-1)]\},$$

cujos coeficientes e sinais são representados com 3 e 2 bits, respectivamente. Supõe-se, novamente, que a densidade de probabilidade da entrada é uniforme e de média nula ($P(x(n) = -0.5) = P(x(n) = 0) = P(x(n) = 0.5) = 1/3$ e $P(x(n) = -1) = 0$) e que o arredondamento é para cima.

Se forem usadas as normas L_p para determinar o escalamento, é necessário usar a função de transferência

$$H(z) = \frac{1}{1 + 0.75z^{-1}}$$

para encontrar o fator de escala m_{esc}, isto é,

$$m_{esc} \leq \frac{0.5}{\|h(n)\|_p \|x(n)\|_q}, \text{ com } \frac{1}{p} + \frac{1}{q} = 1, \tag{4.68}$$

em que $h(n)$ é a resposta impulsiva do filtro. Como $x(n)$ possui energia ilimitada

neste exemplo, usa-se $q = \infty$ e $p = 1$, por onde se calcula

$$\|h(n)\|_1 = 1 + \sum_{n=1}^{\infty} |0.75^n| = 4 \tag{4.69}$$

e

$$\|x(n)\|_\infty = \max|x(n)| = 0.5. \tag{4.70}$$

Usando (4.69) e (4.70) em (4.68), encontra-se $m_{esc} \leq 0.25$ para que não haja saturação de nenhuma saída do filtro.

Por outro lado, pode-se calcular as matrizes $\mathbb{P}$ e $\mathbb{P}^\infty$, ou seja,

$$\mathbb{P} = \begin{array}{c} \begin{array}{cccccc} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.333 & 0.333 \\ 0 & 0 & 0 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0.333 & 0 & 0 \\ 0.333 & 0.333 & 0.333 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array} ,$$

e a matriz em estado estacionário, calculada de forma aproximada por $\mathbb{P}^{100}$,

$$\mathbb{P}^\infty = \begin{array}{c} \begin{array}{cccccc} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.111 & 0.111 & 0.111 & 0.111 & 0.111 & 0.111 \\ 0.222 & 0.222 & 0.222 & 0.222 & 0.222 & 0.222 \\ 0.333 & 0.333 & 0.333 & 0.333 & 0.333 & 0.333 \\ 0.222 & 0.222 & 0.222 & 0.222 & 0.222 & 0.222 \\ 0.111 & 0.111 & 0.111 & 0.111 & 0.111 & 0.111 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array} .$$

Testando iterativamente valores de escalamento na entrada do filtro digital, tendo como valor de referência inicial o escalamento calculado com normas L_p, encontra-

se um fator de escala $m_{esc} = 0.375$ (de 3 bits) que elimina as probabilidades de saturação em $\mathbb{P}$ e $\mathbb{P}^{\infty}$, como pode ser visto nas matrizes.

$$\mathbb{P} = \begin{array}{c} \begin{array}{cccccc} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{array} \quad \textbf{Estados} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.000 & 1.000 \\ 0 & 0 & 0 & 1.000 & 0 & 0 \\ 1.000 & 1.000 & 1.000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{array}{c} -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array} \end{array}$$

e

$$\mathbb{P}^{\infty} = \begin{array}{c} \begin{array}{cccccc} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{array} \quad \textbf{Estados} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1.000 & 1.000 & 1.000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.000 & 1.000 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{array}{c} -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array} \end{array} .$$

Se for usada a matriz $\mathbb{P}$ para a escolha do fator de escalamento, de forma que sejam eliminadas as probabilidades de saturação, certamente a matriz $\mathbb{P}^{\infty}$ ficará livre das probabilidades de saturação. Contudo, a escolha do fator por meio de $\mathbb{P}$ pode fornecer um valor muito conservador, o que acarreta a diminuição da relação sinal-ruído. Por esse motivo, se o objetivo for eliminar a probabilidade de saturação no estado estacionário, deve-se observar as linhas de $\mathbb{P}^{\infty}$ para escolher o melhor fator de escalamento possível. Portanto, a aplicação de cadeias de Markov pode ser usada para a escolha de fatores de escalamento menos conservadores, melhorando a relação sinal-ruído da saída do filtro digital.

Exemplo 2: Se, para o mesmo filtro do exemplo anterior, fosse assumido que o filtro possui sinais de 3 bits, o valor do escalamento calculado via norma L_p seria o mesmo, enquanto que a nova $\mathbb{P}$ de 3 bits seria

$$\mathbb{P} = \begin{array}{c} \begin{array}{cccccccccc} -1.0_s & -1.0 & -0.75 & -0.5 & -0.25 & 0 & 0.25 & 0.5 & 0.75 & 0.75_s \end{array} \\ \left[\begin{array}{cccccccccc} 0.286 & 0.286 & 0.143 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0 & 0 & 0 & 0 & 0 \\ 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0 & 0 & 0 & 0 \\ 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0 & 0 & 0 \\ 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 \\ 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 \\ 0 & 0 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 \\ 0 & 0 & 0 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 \\ 0 & 0 & 0 & 0 & 0 & 0.143 & 0.143 & 0.143 & 0.143 & 0.143 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.143 & 0.286 & 0.286 & 0.286 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0_s \\ -1.0 \\ -0.75 \\ -0.5 \\ -0.25 \\ 0 \\ 0.25 \\ 0.5 \\ 0.75 \\ 0.75_s \end{array} .$$

Iterativamente calculando o fator de escala, seria encontrado $m_{esc} = 0.375$, cuja matriz de transição de estados após o escalamento corresponde a

$$\mathbb{P} = \begin{array}{c} \begin{array}{cccccccccc} -1.0_s & -1.0 & -0.75 & -0.5 & -0.25 & 0 & 0.25 & 0.5 & 0.75 & 0.75_s \end{array} \\ \left[\begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.286 & 0.286 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.429 & 0.429 & 0.286 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.285 & 0.285 & 0.429 & 0.286 & 0.286 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.285 & 0.429 & 0.429 & 0.286 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.285 & 0.285 & 0.429 & 0.286 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.285 & 0.429 & 0.286 & 0.286 & 0.286 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.285 & 0.429 & 0.429 & 0.429 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.285 & 0.285 & 0.285 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0_s \\ -1.0 \\ -0.75 \\ -0.5 \\ -0.25 \\ 0 \\ 0.25 \\ 0.5 \\ 0.75 \\ 0.75_s \end{array} ,$$

que também é livre de saturação. A mesma ideia pode ser aplicada em filtros de

ordem 2, com resultados semelhantes. Não é apresentado um exemplo aqui em função das dimensões das matrizes obtidas, que precisam ser reduzidas demais para serem comportadas no espaço de uma folha, tornando difícil a visualização. (Por exemplo: uma matriz calculada para 2 bits tem dimensões de 36×36 para a matriz de transição de estados estendida.)

4.4.2 Comparação com o modelo linear

Em uma abordagem tradicional, as não-linearidades oriundas da quantização são modeladas como um erro $e(n)$ de distribuição uniforme que é somado ao sinal após cada multiplicador. Dessa forma, lineariza-se o efeito da quantização através da adição de diversos sinais de erro após cada um dos multiplicadores, e assume-se que não há saturação ou *overflow*. Essa linearização pode alterar muito a variância da saída do filtro, tornado esse modelo pouco preciso para análises com poucos bits. Nesse caso, cadeias de Markov podem ser usadas para um resultado mais preciso. Tomando como exemplo o filtro de primeira ordem do Exemplo 1 (pág. 93), onde $a_1 = 0.75$, $b_0 = 1$ e $b_1 = 0$, existe apenas uma fonte de erro de quantização depois da multiplicação por a_1. Dessa forma, a variância da saída pode ser calculada recursivamente por meio de

$$\begin{aligned} \mathrm{E}\{y(n)^2\} = \; & a_1^2 E\{y(n-1)^2\} - 2a_1 b_0 \,\mathrm{E}\{y(n-1)x(n)\} \\ & + b_0^2 \,\mathrm{E}\{x(n)^2\} + \mathrm{E}\{e(n)^2\}. \end{aligned} \tag{4.71}$$

Se for assumido que $x(n)$ é independente de $y(n-1)$, o que é válido se o processo $\{x(n)\}$ for independente, de média zero e identicamente distribuído (iid), o termo $2a_1 b_0 \,\mathrm{E}\{y(n-1)x(n)\} = 0$. Além disso, os termos $b_0^2 \,\mathrm{E}\{x(n)^2\}$ e $\mathrm{E}\{e(n)^2\}$ só precisam ser calculados uma vez, o que torna $\mathrm{E}\{y(n)^2\}$ dependente apenas de seus valores passados, simplificando o cálculo. Deve-se lembrar, ainda, que a média de $y(n)$ é igual a 0, já que todos os sinais envolvidos possuem média nula, e $\mathrm{E}\{y(0)\} = 0$, já que $y(0) = 0$.

Usando a matriz de transição de estados, também é possível calcular a média e a variância recursivamente. Escolhendo uma condição inicial $\boldsymbol{\pi}(0)$ para a saída do filtro (por exemplo, iniciar o filtro com $y(0) = 0$, que corresponde a um $\boldsymbol{\pi}(0)$ em que $P(y(0) = 0) = 1$ e $P(y(0) \neq 0) = 0$), e usando um vetor coluna $\boldsymbol{s}$ com os valores dos estados, a média μ é calculada por

$$\begin{aligned}
\boldsymbol{\pi}(1) &= \mathbb{P}\boldsymbol{\pi}(0) \\
\mu(1) &= \boldsymbol{s}^T\boldsymbol{\pi}(1) \\
\boldsymbol{\pi}(2) &= \mathbb{P}^2\boldsymbol{\pi}(0) \\
\mu(2) &= \boldsymbol{s}^T\boldsymbol{\pi}(2) \quad , \\
&\vdots \\
\boldsymbol{\pi}(n) &= \mathbb{P}^n\boldsymbol{\pi}(0) \\
\mu(n) &= \boldsymbol{s}^T\boldsymbol{\pi}(n)
\end{aligned} \tag{4.72}$$

em que, para o exemplo, $\boldsymbol{\pi}(0)$ e $\boldsymbol{s}$ correspondem à

$$\boldsymbol{\pi}(0) = [0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0]^T$$

e

$$\boldsymbol{s} = [-1 \;\; -1 \;\; -0.5 \;\; 0 \;\; 0.5 \;\; 0.5]^T,$$

e o vetor $\boldsymbol{\pi}(0)$ corresponde a $P(y(0) = 0) = 1$.

A variância pode ser calculada de forma semelhante, através de

$$\sigma_y^2(k) = \boldsymbol{\pi}(k)^T \begin{bmatrix} (-1-\mu(k))^2 \\ (-1-\mu(k))^2 \\ (-0.5-\mu(k))^2 \\ (0-\mu(k))^2 \\ (0.5-\mu(k))^2 \\ (0.5-\mu(k))^2 \end{bmatrix}. \tag{4.73}$$

A figura 28 apresenta os valores da aproximação linearizada e por cadeias de

Markov.

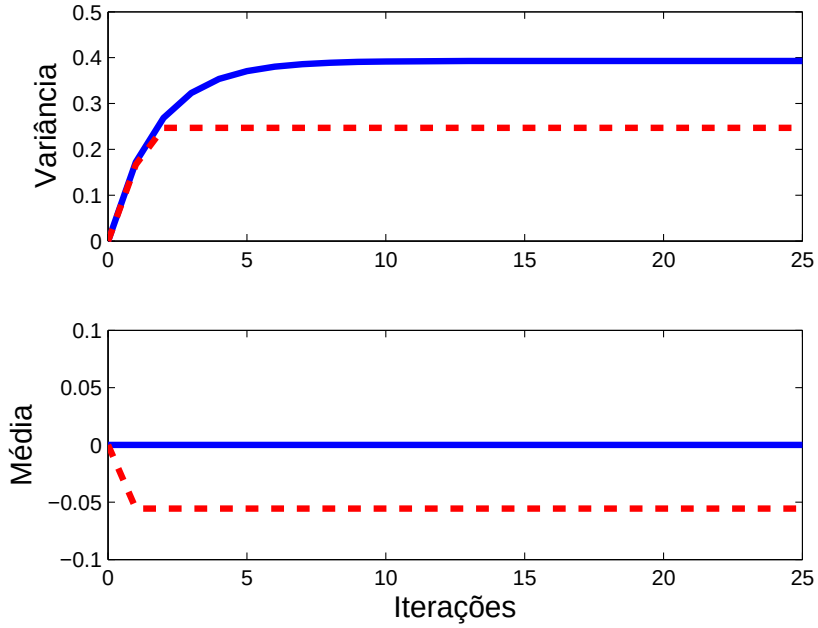

Figura 28: Média e variância da saída saturada, obtidas por meio de cadeias de Markov (linha pontilhada) e pelo modelo linear (curva contínua)

Observando a figura, nota-se que a análise dos efeitos de quantização para a aproximação linearizada é significativamente diferente da análise por meio da matriz de transição. De fato, essa diferença deve ser maior em filtros de palavras com um pequeno número de bits, dado que os erros de quantização se tornam mais relevantes para a variância da saída.

Os cálculos de média e de variância também podem ser estendidos para filtros de segunda ordem. A figura 29 compara a abordagem linear e a via cadeia de Markov, considerando um filtro dado pela equação

$$H(z) = \frac{0.8 - 0.4z^{-1} - 0.2z{-}2}{1 - 0.5z^{-1} - 0.2z^{-2}}, \tag{4.74}$$

implementado com 5 bits e supondo que a f.d.p de entrada é gaussiana, descorrelacionada, com média zero e variância 0.6455 (calculada assumindo que os valores da entrada estão entre -1 e $1 - \Delta$). Assume-se que a entrada é discretizada de

acordo com o apresentado na seção 2.7.3.

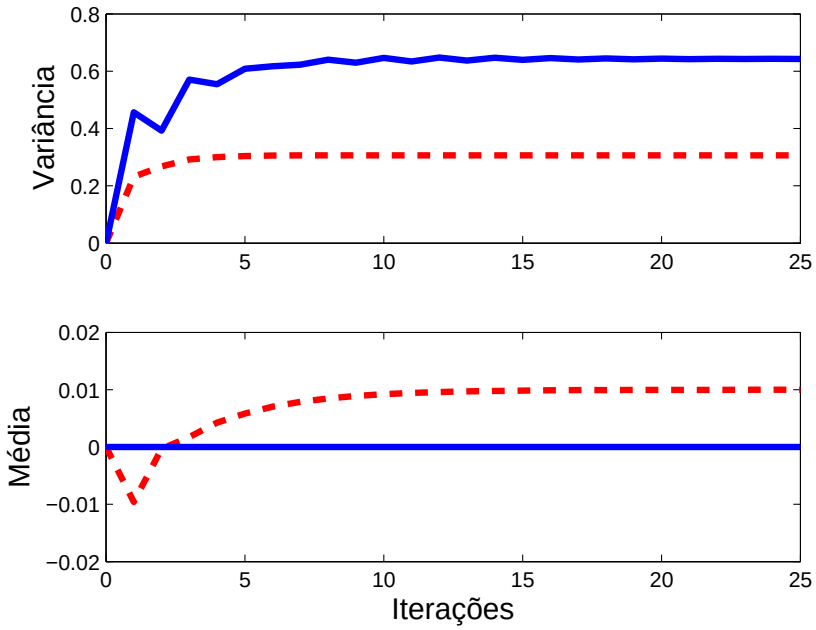

Figura 29: Média e variância da saída saturada, obtidas por meio de cadeias de Markov (linha pontilhada) e pelo modelo linear (curva contínua) para um filtro de segunda ordem

Para calcular a variância e a média via cadeia de Markov, usa-se a matriz de transição de estados não-expandida e as equações

$$E\{y^2(n)\} = \sum_{i=-2^{-B+1}}^{2^{-B+1}-1} (i\Delta)^2 \sum_{j=-2^{-B+1}}^{2^{-B+1}-1} \sum_{k=-2^{-B+1}}^{2^{-B+1}-1} P(i\Delta | y(n-1) = j\Delta, y(n-2) = k\Delta) \times$$

$$\sum_{l=-2^{-B+1}}^{2^{-B+1}-1} \sum_{m=-2^{-B+1}}^{2^{-B+1}-1} P(y(n-1) = j\Delta, y(n-2) = k\Delta | y(0) = l\Delta, y(-1) = m\Delta) \times$$

$$P(y(0) = l\Delta, y(-1) = m\Delta),$$

e

$$E\{y(n)\} = \sum_{i=-2^{B-1}}^{2^{B-1}-1} i\Delta \sum_{k=-2^{B-1}}^{2^{B-1}-1} \sum_{l=-2^{B-1}}^{2^{B-1}-1} \sum_{m=-2^{B-1}}^{2^{B-1}-1}$$

$$P(y(n) = i\Delta, y(n-1) = k\Delta | y(0) = l\Delta, y(-1) = m\Delta) \times$$

$$P(y(0) = l\Delta, y(-1) = m\Delta),$$

onde $y(-1)$ e $y(0)$ são as condições iniciais e $y(n)$ é escrito em termos dos valores discretos que pode assumir ($y(n) = i\Delta$, para i inteiro e entre -2^{B-1} e $2^{B-1} - 1$). Os termos $P(y(n-1) = j\Delta, y(n-2) = k\Delta | y(0) = l\Delta, y(-1) = m\Delta)$ correspondem aos elementos da matriz de transição de estados $\mathbb{P}$ em sua n-ésima potência. O modelo linearizado pode ser calculado de forma iterativa, semelhante ao apresentado na equação 4.71, fornecendo

$$\begin{aligned} \mathrm{E}\{y(n-1)y(n-2)\} = \; & (b_0b_1 + b_1b_2 - b_0b_2a_1)\sigma_x^2 \\ & -a_1\,\mathrm{E}\{y^2(n-1)\} - a_1\,\mathrm{E}\{y(n-2)y(n-3)\} \end{aligned}$$

e

$$\begin{aligned} \mathrm{E}\{y^2(n)\} = \; & a_1^2\,\mathrm{E}\{y^2(n-1)\} + a_2^2\,\mathrm{E}\{y^2(n-2)\} + 2a_1a_2\,\mathrm{E}\{y(n-1)y(n-2)\} \\ & +\sigma_x^2(b_0^2 + b_1^2 + b_2^2 - 2(b_0b_1a_1 + b_1b_2a_1 - b_0b_2a_1^2 + b_0b_2a_2)) \\ & +\sigma_{\eta_1}^2 + \sigma_{\eta_2}^2 + \sigma_{\eta_3}^2 + \sigma_{\eta_4}^2 + \sigma_{\eta_5}^2, \end{aligned}$$

onde σ_x^2 e $\sigma_{\eta_k}^2$ correspondem à variância da entrada e à do erro de quantização, respectivamente. Os elementos b_0, b_1 e b_2 correspondem aos termos do numerador de (4.74), e $a_0 = 1$, a_1 e a_2 correspondem aos elementos do denominador.

Nesse caso, também para filtros de segunda ordem é possível notar a diferença existente entre o modelo via cadeias de Markov e a abordagem linearizada.

4.5 Análise usando não-linearidade de *overflow*

4.5.1 Comparação com o modelo linear

Exemplo: Suponha novamente um filtro IIR de primeira ordem em que os sinais são representados com 2 bits (ou seja, o conjunto de números representáveis é $\{-1,\ -0.5,\ 0,\ 0.5\}$) e os coeficientes são representados com 3 bits, descrito pela equação

$$y(n) = R\{Q[x(n)] + Q[-0.75y(n-1)]\},$$

em que se usa a não-linearidade de *overflow*. Quando a entrada possui uma função densidade de probabilidade uniforme e de média nula (ou seja, a probabilidade de $x(n) = -0.5$, 0 ou 0.5 é igual a 1/3 e a probabilidade de $x(n) = -1$ é igual a 0), e assume-se que os números são arredondados para cima, $\mathbb{P}$ é dada por

$$\mathbb{P} = \begin{array}{c} \begin{array}{cccc} -1.0 & -0.5 & 0 & 0.5 \end{array} \\ \left[\begin{array}{cccc} 0.333 & 0.333 & 0 & 0.333 \\ 0.333 & 0 & 0.333 & 0.333 \\ 0 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \end{array} .$$

Supondo uma condição inicial em que $P(y(0) = 0) = 1$, que corresponde a $\boldsymbol{\pi}(0) = [0\ 0\ 1\ 0]^T$, e que $\boldsymbol{s} = [-1\ \ -0.5\ 0\ 0.5]^T$ é um vetor com os valores que os estados podem assumir, pode-se usar a matriz de transição para encontrar a média μ e a variância σ_y^2 exatas da saída, ou seja,

$$\begin{aligned} \boldsymbol{\pi}(1) &= \mathbb{P}\boldsymbol{\pi}(0) \\ \mu(1) &= \boldsymbol{s}^T\boldsymbol{\pi}(1) \\ \boldsymbol{\pi}(2) &= \mathbb{P}^2\boldsymbol{\pi}(0) \\ \mu(2) &= \boldsymbol{s}^T\boldsymbol{\pi}(2) \\ &\vdots \\ \boldsymbol{\pi}(n) &= \mathbb{P}^n\boldsymbol{\pi}(0) \\ \mu(n) &= \boldsymbol{s}^T\boldsymbol{\pi}(n) \end{aligned} \tag{4.75}$$

e

$$\sigma_y^2(k) = \boldsymbol{\pi}^T(k) \begin{bmatrix} (-1-\mu(k))^2 \\ (-0.5-\mu(k))^2 \\ (0-\mu(k))^2 \\ (0.5-\mu(k))^2 \end{bmatrix}. \tag{4.76}$$

A figura 30 mostra a média e a variância obtidas iterativamente para o filtro

considerando $\mathbb{P}$ (curva pontilhada), que é bem diferente da abordagem tradicional (curva contínua).

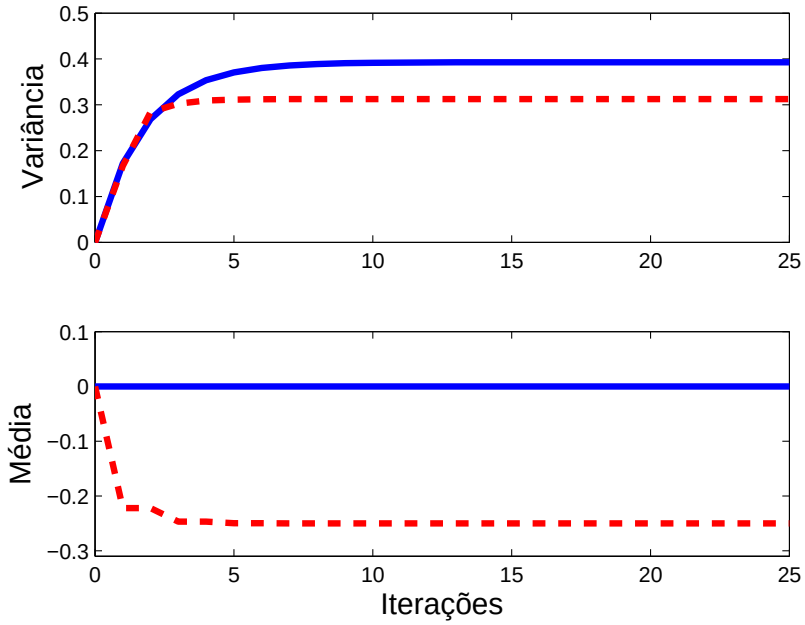

Figura 30: Média e variância da saída obtida por *overflow*, calculadas com de cadeias de Markov (curvas pontilhadas) e via modelo linear (curvas contínuas).

4.6 Identificação de ciclos-limite de entrada zero

Como mostrado na seção 2.11.1.2 (pág. 35), o Teorema de Perron-Frobenius garante que a matriz de transição de estados sempre possui ao menos um autovalor igual a 1. Além disso, em matrizes estocásticas como $\mathbb{P}$ (ou seja, uma matriz não-negativa em que a soma de cada uma das colunas ou de cada uma das linhas é igual a 1 [9]), os autovalores λ_k satisfazem a relação $\lambda_1 = 1 \geqslant |\lambda_2| \geqslant \ldots \geqslant |\lambda_N| \geqslant 0$. Com essa informação, pode-se escrever $\mathbb{P}$ e suas potências em termos de seus

autovalores e autovetores, isto é [2],

$$\mathbb{P}^n = \boldsymbol{V}\boldsymbol{D}^n\boldsymbol{V}^{-1} = \boldsymbol{V}\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \lambda_2^n & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_N^n \end{bmatrix}\boldsymbol{V}^{-1}, \tag{4.77}$$

onde $\boldsymbol{D}$ corresponde à matriz diagonal dos autovalores e $\boldsymbol{V}$ é a matriz dos autovetores de $\mathbb{P}$.

De (4.77) fica patente que a influência dos autovalores menores (em módulo) que 1 tende a se reduzir com o crescimento de n, até a situação em que $n \to \infty$ e esses autovalores tendem a 0. Esse conhecimento antecipa a informação de que a matriz $\mathbb{P}^\infty$ depende apenas de autovalores em que o módulo é igual a 1 e de seus autovetores associados para definir as densidades de probabilidade condicionada presentes em suas colunas. Dessa forma, o estudo desses elementos nas matrizes $\mathbb{P}$ e $\mathbb{P}^\infty$ leva à identificação dos ciclos-limite de entrada nula.

A seguir, mostra-se que para a ausência de ciclos-limite de entrada nula, é necessário que $\mathbb{P}$ possua apenas um autovalor igual a 1 e que os demais sejam $|\lambda_k| < 1$.

4.6.1 Ciclos-limite de entrada nula

Quando a entrada de um filtro se torna nula, é necessário que após um tempo finito a saída se torne zero, para que não haja ciclos-limite de entrada nula. Em linguagem de cadeias de Markov, isso significa que quando a probabilidade de entrada usada para calcular $\mathbb{P}$ é $P(x(n) = 0) = 1$, a f.d.p. da saída deve ser tal que $\lim_{n\to\infty} P(y(n) = 0) = 1$, para qualquer condição inicial $\boldsymbol{\pi}(0)$.

Suponha que o vetor coluna $\boldsymbol{\pi} = [1\ 0\ \dots\ 0]^T$ (de dimensão $N \times 1$) corres-

[2]Por simplicidade é assumido que $\mathbb{P}$ é diagonalizável. O argumento não se altera muito se isso não for verdade, dado que pode-se escrever $\boldsymbol{D}$ em termos de blocos de Jordan

ponda à densidade de probabilidade de $y(n)$, com $P(y(n) = 0) = 1$ (ou seja, a probabilidade do estado zero é igual a 1) e que a matriz $\mathbb{P}^{\infty}$ $(N \times N)$ seja reorganizada, de forma que a primeira linha corresponda ao estado zero. Se $\mathbb{P}^{\infty}$ não permitir ciclos limite de entrada nula, então para qualquer condição inicial $\boldsymbol{\pi}(0)$, deve acontecer

$$\mathbb{P}^{\infty}\boldsymbol{\pi}(0) = \boldsymbol{\pi}. \tag{4.78}$$

Escolhendo, por exemplo, $\boldsymbol{\pi}(0) = \boldsymbol{\pi}$,

$$\mathbb{P}^{\infty}\boldsymbol{\pi}(0) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1(N-1)} & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2(N-1)} & p_{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{N1} & p_{2N} & \cdots & p_{M(N-1)} & p_{NN} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{4.79}$$

e é possível notar que a primeira coluna de $\mathbb{P}^{\infty}$ deve ser

$$\begin{bmatrix} p_{11} \\ p_{21} \\ \vdots \\ p_{N1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

para que (4.78) seja válido. Escolhendo, uma outra condição inicial $\boldsymbol{\pi}(0) = [0\ 1\ 0 \ldots\ 0]^T$, é possível verificar que

$$\begin{bmatrix} p_{12} \\ p_{22} \\ \vdots \\ p_{N2} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Assim, para qualquer condição inicial da forma $\boldsymbol{\pi}(0) = [0\ \ldots\ 0\ 1\ 0\ \ldots\ 0]^T$, em que o valor 1 ocupa a posição k do vetor $(1 \leq k \leq N)$, verifica-se que a coluna

de $\mathbb{P}^\infty$ equivalente à posição k será igual a $\boldsymbol{\pi}$, fazendo com que $\mathbb{P}^\infty$ tenha a forma

$$\mathbb{P}^\infty = \begin{bmatrix} 1 & \dots & 1 & \dots & 1 \\ 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix}.$$

De fato, uma condição inicial qualquer pode ser obtida por meio de combinações lineares dessas condições $\boldsymbol{\pi}(0) = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T$, com 1 na posição k, de maneira que é possível mostrar que para qualquer $\boldsymbol{\pi}(0)$, a multiplicação $\mathbb{P}^\infty\boldsymbol{\pi}(0)$ será igual ao vetor $[1 \ 0 \ \dots \ 0]^T$,

$$\begin{bmatrix} 1 & \dots & 1 & \dots & 1 \\ 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \pi(1) \\ \pi(2) \\ \vdots \\ \pi(N) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} \pi(i) \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

em que $\sum_{i=1}^{N} \pi(i) = 1$ para que o vetor seja um vetor de probabilidades.

De (4.79), é fácil perceber que $\boldsymbol{\pi}$ é um autovetor de $\mathbb{P}^\infty$, associado ao autovalor 1. Se todos os autovalores de $\mathbb{P}^\infty$ forem calculados a partir da equação característica $\det(\mathbb{P}^\infty - \lambda I) = 0$ (em que I é a matriz identidade de dimensões adequadas e λ é um escalar), os autovalores serão tais que

$$(1 - \lambda)(-\lambda)^{(N-1)} = 0,$$

de onde vem que $\mathbb{P}^\infty$ possui um único autovalor 1 (ou seja, com multiplicidade um) e os demais são iguais a zero. Nesse caso, é necessário analisar $\mathbb{P}$ para definir as condições em que $\mathbb{P}^\infty$ satisfaz esse critério, evitando o cálculo de potências de $\mathbb{P}$ para a busca de ciclos-limite de entrada nula.

Para a matriz $\mathbb{P}$, pode-se encontrar sua forma de Jordan (vide [9]), que cor-

responde a uma matriz diagonal por blocos, em que cada bloco corresponde a um bloco de Jordan, com a forma geral

$$\boldsymbol{J}_k(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & \dots & 0 \\ 0 & \lambda & 1 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \lambda & 1 \\ 0 & \dots & 0 & 0 & \lambda \end{bmatrix}_{k\times k}, \tag{4.80}$$

em que λ é um autovalor de $\mathbb{P}$, com multiplicidade k. Nesse caso, $\mathbb{P}$ pode ser escrita como

$$\mathbb{P} = \boldsymbol{V} \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_{k_2}(\lambda_2) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J}_{k_3}(\lambda_3) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{J}_{k_M}(\lambda_M) \end{bmatrix} \boldsymbol{V}^{-1}, \tag{4.81}$$

em que os coeficientes k_2, ..., k_M correspondem às multiplicidades dos autovalores λ_2, ..., λ_M, respectivamente. $\boldsymbol{V}$ é a matriz que contém em suas colunas os autovetores de $\mathbb{P}$. Note que se a matriz for diagonalizável, os blocos de Jordan são substituídos pelos autovalores de $\mathbb{P}$, o que equivale ao caso em que os blocos de Jordan são todos do tipo $J_1(\lambda_k)$. As potências de $\mathbb{P}$ podem ser escritas como

$$\mathbb{P}^n = \boldsymbol{V} \begin{bmatrix} 1^n & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}^n_{k_2}(\lambda_2) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J}^n_{k_3}(\lambda_3) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{J}^n_{k_M}(\lambda_M) \end{bmatrix} \boldsymbol{V}^{-1}, \tag{4.82}$$

em que vale a propriedade

$$\boldsymbol{J}_k^n(\lambda) = \begin{bmatrix} \lambda^n & \binom{n}{1}\lambda^{n-1} & \binom{n}{2}\lambda^{n-2} & \dots & \binom{n}{k-1}\lambda^{n-k+1} \\ 0 & \lambda^n & \binom{n}{1}\lambda^{n-1} & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \binom{n}{2}\lambda^{n-2} \\ \vdots & \vdots & \vdots & \lambda^n & \binom{n}{1}\lambda^{n-1} \\ 0 & 0 & 0 & \dots & \lambda^n \end{bmatrix}, \tag{4.83}$$

onde

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Calculando $\mathbb{P}^\infty = \lim_{n\to\infty}\mathbb{P}^n$,

$$\mathbb{P}^\infty = \boldsymbol{V}\begin{bmatrix} \lim_{n\to\infty}1^n & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \lim_{n\to\infty}\boldsymbol{J}_{k1}^n(\lambda_2) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lim_{n\to\infty}\boldsymbol{J}_{k2}^n(\lambda_3) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \lim_{n\to\infty}\boldsymbol{J}_{kM}^n(\lambda_M) \end{bmatrix}\boldsymbol{V}^{-1}.$$

Para que haja apenas o autovalor 1 em $\mathbb{P}^n$, quando $n \to \infty$, é necessário que todos os blocos de Jordan se tornem blocos de zeros, para $n \to \infty$. Considerando a estrutura de (4.83), a única maneira de se obter um bloco de zeros corresponde à situação em que $|\lambda_i| < 1$, $2 \leq i \leq M$, de forma que $\lim_{n\to\infty}\lambda_i^n = 0$, $2 \leq i \leq M$. Portanto, a matriz $\mathbb{P}$ deve ter apenas um autovalor igual a 1 e os demais com módulo menor que 1 para que não exista ciclo-limite de entrada zero.

4.6.2 Encontrando os ciclos-limite de entrada nula

Exemplo 1: Deseja-se verificar a existência de ciclos-limite de entrada zero em um filtro IIR de primeira ordem, implementado com não-linearidades de *over-*

flow. Assume-se que os sinais e coeficientes envolvidos usam palavras binárias de 2 bits de comprimento e que os valores são arredondados para cima. O filtro a ser implementado é um passa-tudo descrito pela equação

$$H(z) = \frac{0.5 - z^{-1}}{1 - 0.5z^{-1}}.$$

Usando uma entrada em que $P(x(n) = 0) = 1$, encontra-se a matriz

$$\mathbb{P} = \begin{array}{c} \begin{array}{cccc} -1.0 & -0.5 & 0 & 0.5 \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{array}{c} \text{Estados} \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \end{array} .$$

Se forem calculados os autovalores de $\mathbb{P}$, são obtidos $\{0,\ 0,\ 1,\ 1\}$, mostrando que o autovalor 1 tem multiplicidade 2. A matriz de estado estacionário confirma a existência dos ciclos, como pode ser observado a seguir.

$$\mathbb{P}^{\infty} = \begin{array}{c} \begin{array}{cccc} -1.0 & -0.5 & 0 & 0.5 \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{array}{c} \text{Estados} \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \end{array}$$

Nesse caso, como uma das colunas difere de saída zero com 100% de certeza (a coluna correspondente a $y(n-1) = 0.5$), existe a possibilidade de a entrada ser nula e a saída não cair a zero ao longo das iterações, o que caracteriza ciclos-limite. De fato, simulando o filtro para as diversas condições iniciais possíveis de $y(0)$, observa-se o ciclo-limite. A figura 31 apresenta os resultados simulados em *Matlab*.

Exemplo 2: Deseja-se verificar a existência de ciclos-limite de entrada zero

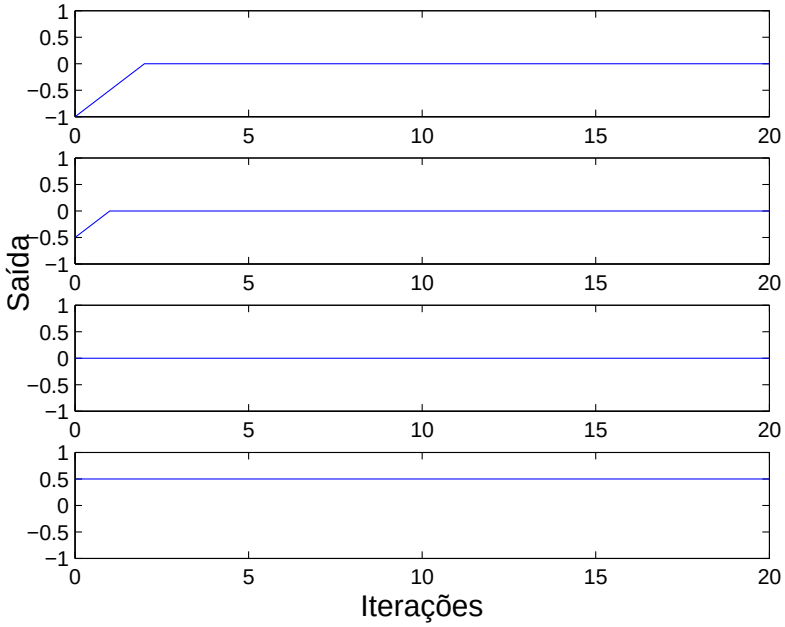

Figura 31: Saída do filtro digital do exemplo, para entrada nula. De cima para baixo: condição inicial $y(0) = -1,\ -0.5,\ 0$ e 0.5, respectivamente.

em um filtro IIR de segunda ordem, implementado com não-linearidades de *overflow*. Assume-se que os sinais e coeficientes envolvidos usam palavras binárias de 2 bits de comprimento e que os valores são arredondados para cima. O filtro a ser implementado é descrito pela equação

$$H(z) = \frac{1}{1 + 0.25z^{-1} + 0.5z^{-2}}.$$

Usando um vetor de condição inicial em que $P(y(0) = 0, y(-1) = 0) = 1$,

encontra-se a matriz

$$\mathbb{P} = \begin{array}{c} \\ \left[\begin{array}{cccccccccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} \text{Estados} \\ -1.0\ -1.0 \\ -0.5\ -1.0 \\ 0\ -1.0 \\ 0.5\ -1.0 \\ -1.0\ -0.5 \\ -0.5\ -0.5 \\ 0\ -0.5 \\ 0.5\ -0.5 \\ -1.0\ 0 \\ -0.5\ 0 \\ 0\ 0 \\ 0.5\ 0 \\ -1.0\ 0.5 \\ -0.5\ 0.5 \\ 0\ 0.5 \\ 0.5\ 0.5 \end{array} .$$

Os estados acima da matriz foram omitidos por problemas de espaço, enquanto os estados indicados à direita correspodem ao par $(y(n), y(n-1))$

Se forem calculados os autovalores de $\mathbb{P}$, são obtidos dois autovalores 1 e os demais são tais que $|\lambda_k| < 1$. A matriz de estado estacionário confirma a existência dos ciclos, como pode ser observado a seguir.

$$\mathbb{P}^{\infty} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \text{Estados} \\ \left[\begin{array}{cccccccccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \\ -1.0\ -1.0 \\ -0.5\ -1.0 \\ 0\ -1.0 \\ 0.5\ -1.0 \\ -1.0\ -0.5 \\ -0.5\ -0.5 \\ 0\ -0.5 \\ 0.5\ -0.5 \\ -1.0\ 0 \\ -0.5\ 0 \\ 0\ 0 \\ 0.5\ 0 \\ -1.0\ 0.5 \\ -0.5\ 0.5 \\ 0\ 0.5 \\ 0.5\ 0.5 \end{array} .$$

4.7 Aplicação de cadeias de Markov para a análise de filtros em cascata

Nesta seção, deseja-se verificar se é possível simplificar o modelo, aproximando o modelo completo de um filtro de segunda ordem por um modelo com dois filtros de primeria ordem em cascata, considerando cada termo da cascata separadamente. O que torna essa abordagem interessante é a redução do número de estados (que no modelo de segunda ordem corresponde a N^2 e no de primeira ordem, a N para cada filtro da cascata, com $N = 2^B$), possibilitando a redução

do número de operações para a obtenção da f.d.p. da saída do filtro.

Para a comparação, são analisadas implementações passa-tudo e passa-baixa realizadas em cascata, que são comparadas aos filtros de segunda ordem equivalentes. A posição dos filtros da cascata é variada, revezando a posição dos filtros com maiores e menores pólos e zeros. O intuito é investigar:

1. Se para filtros passa-tudo com entrada descorrelacionada, o modelo com filtros em cascata pode ser uma boa alternativa ao modelo de segunda ordem. Isso porque para filtros passa-tudo, se a entrada for descorrelacionada, a saída também é descorrelacionada. Nesse caso, a saída do primeiro filtro, que também é a entrada do segundo filtro da cascata, se mantém descorrelacionada, o que permitiria usar os modelos desenvolvidos nas seções 4.3 e 4.2 como uma aproximação razoável, já que esses modelos pressupõem entrada iid. Com isso, cada filtro poderia ser analisado com uma matriz de estados própria. Note que o modelo de cascata de filtros é uma simplificação, pois o modelo desenvolvido anteriormente exigiria que a entrada do segundo filtro fosse iid, não apenas descorrelacionada.

2. Se para filtros passa-baixa, a abordagem em cascata seria razoável quando pólos e zeros estão bem próximos da circunferência unitária, já que entrada descorrelacionada não implica saída descorrelacionada nesses filtros.

Para encontrar a f.d.p. da saída da cascata de filtros, inicialmente calcula-se a matriz de Markov $\mathbb{P}_1$ para o filtro mais próximo do sinal de entrada $x(n)$, que possui uma f.d.p. p_x. Considerando uma condição inicial $\boldsymbol{\pi}_{ci1}$, é encontrado um vetor $\boldsymbol{\pi} = \mathbb{P}_1\boldsymbol{\pi}_{ci1}$, que será usado como f.d.p. de entrada para o segundo filtro. Com $\boldsymbol{\pi}$, é possível calcular a matriz de Markov do segundo filtro, $\mathbb{P}_2$. Usando uma condição inicial $\boldsymbol{\pi}_{ci2}$, encontra-se, por fim, a densidade de probabilidade da saída $\boldsymbol{\pi}_{out} = \mathbb{P}_2\boldsymbol{\pi}_{ci2}$. Essa sequência de passos deve ser realizada de forma iterativa

para que a densidade de probabilidade dos estados correspondentes à saída possa ser encontrada após n iterações. Nessa situação, as matrizes $\mathbb{P}_1$ e $\mathbb{P}_2$ apresentam dimensão de $N \times N$ elementos cada uma e a f.d.p. da saída corresponde a um vetor de dimensão $N \times 1$. Em regime, a distribuição de probabilidades da saída do primeiro filtro fica estável, e a matriz $\mathbb{P}_2$ também se estabiliza. Para simplificar os cálculos é possível calcular a matriz $\mathbb{P}_2$ somente para o caso de regime permanente.

A f.d.p. da saída do modelo de segunda ordem é calculada pelo modelo descrito nas seções 4.2 (pág. 67) e 4.3 (pág. 84). Nesse caso, como existem N^2 estados, a f.d.p. da saída é descrita por um vetor de dimensão $N^2 \times 1$. Para tornar a f.d.p. da saída do modelo de segunda ordem comparável à do modelo da cascata de filtros, as probabilidades associadas aos estados são adicionadas para fornecer um vetor de dimensão $N \times 1$. Para isso, deve-se lembrar que os estados de um filtro de segunda ordem são compostos por um par $(y(n), y(n-1))$, em que $y(n)$ é a saída atual e $y(n-1)$ é a saída do instante anterior, enquanto no filtro de primeira ordem, os estados são fornecidos apenas pela saída atual. Logo, para descobrir a probabilidade da saída atual do filtro de segunda ordem ser igual a $y(n) = i\Delta$ (para $-2^{(B-1)} \leq i \leq 2^{(B-1)} - 1$), basta somar a probabilidade de todos os estados em que $y(n) = i\Delta$, i.é,

$$P(y(n) = i\Delta) = \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} P(y(n) = i\Delta, y(n-1) = k\Delta), \tag{4.84}$$

de onde se obtém um vetor de densidade de probabilidade com N elementos.

Essas abordagens para calcular as f.d.p. para filtros em cascata e de segunda ordem são usadas nos exemplos que se seguem, para filtros passa-tudo e passa-baixa.

4.7.1 Implementação de filtros passa-tudo

Um filtro passa-tudo digital de primeira ordem, assumindo precisão infinita, é descrito no domínio do tempo pela equação de diferenças

$$y(n) = ax(n) + x(n-1) - ay(n-1). \tag{4.85}$$

Para esse tipo de implementação, se a entrada for descorrelacionada ($\mathrm{E}\{x(n-k)x(n-j)\} = 0$, $\forall k \neq j$), então a saída também é descorrelacionada ($\mathrm{E}\{y(n-k)y(n-j)\} = 0$, $\forall k \neq j$). De fato, considerando o espectro de potência [10] desse filtro, a relação entre a entrada e a saída é calculada por

$$S_y(f) = |H(f)|^2 S_x(f), \tag{4.86}$$

em que $|H(f)|^2 = 1$. Com isso, $S_y(f) = S_x(f)$, de onde se conclui que a saída é descorrelacionada se a entrada for descorrelacionada.

Se for considerada uma implementação em precisão finita de um filtro passa-tudo, espera-se que quando os pólos e zeros forem pequenos o suficiente para evitar a saturação ou o *overflow*, a saída deva ser aproximadamente descorrelacionada. Com isso, a matriz do segundo estágio pode ser calculada sem que seja necessário considerar a correlação da entrada no cálculo dos estados do segundo filtro, o que torna vantajoso o uso modelo de cascata de filtros proposto. Essa aproximação é usada nos exemplos a seguir.

4.7.2 Exemplos com filtros passa-tudo

Neste exemplo, dois filtros passa-tudo de primeira ordem, colocados em cascata, são implementados assumindo que não existe um acumulador grande o suficiente para realizar a quantização ao final de todas as operações (isto é, a quantização é realizada após cada multiplicação – vide equação (4.2), pág. 64). Considera-se que os filtros são causais, o que significa que a condição inicial da

saída de cada um dos filtros concatenados é $P(y(0) = 0) = 1$, e que as matrizes de transição de estados são calculadas para entrada descorrelacionada, uniforme e de média nula. Após cada quantização, os valores obtidos são arredondados para cima. São apresentadas as funções de densidade de probabilidade da saída em regime estacionário (ou seja, quando $n \to \infty$). As funções de probabilidade são calculadas de duas formas distintas: em uma implementação de segunda ordem (vide equação (4.1), pág. 64) e usando duas matrizes de transição de estados, calculadas para o modelo de cascata de filtros. As funções de probabilidade obtidas são comparadas por meio de figuras e em termos de média e de variância, que são apresentadas em tabelas e comparadas ao modelo linearizado do filtro de segunda ordem.

4.7.2.1 Cálculo com o modelo linear

Para comparação com os resultados obtidos via cadeias de Markov, a média e a variância são calculadas para o modelo linear de um filtro de segunda ordem. Para isso, assume-se que a entrada $x(n)$ é iid e que o filtro é causal ($y(n) = 0$, quando $n \leq 0$), o que implica em $\mathrm{E}\{y(n)\} = 0$, se $n \leq 0$.

Para o filtro de segunda ordem descrito pelo modelo linear, a equação de diferenças é calculada por

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2) + \eta_1(n) + \eta_2(n) + \eta_3(n) - \eta_4(n) - \eta_5(n), \quad (4.87)$$

em que cada η_k corresponde ao ruído introduzido pelo modelo linear após cada multiplicação. Cada η_k é considerado independente e possui distribuição uniforme, de média zero e de variância $\sigma_\eta^2 = \Delta^2/12$.

A média de $y(n)$ será igual a 0, já que a entrada do filtro possui média zero

e trata-se de um sistema linear. A variância de $y(n)$ pode ser calculada por

$$\begin{aligned}
&\mathrm{E}\{y^2(n)\} = \mathrm{E}\{(b_0x(n) + b_1x(n-1) + b_2x(n-2))^2\} \\
&+ \mathrm{E}\{(\eta_1(n) + \eta_2(n) + \eta_3(n) - \eta_4(n) - \eta_5(n))^2\} + \mathrm{E}\{(a_1y(n-1) + a_2y(n-2))^2\} \\
&+ 2\,\mathrm{E}\{(b_0x(n) + b_1x(n-1) + b_2x(n-2))(\eta_1(n) + \eta_2(n) + \eta_3(n) - \eta_4(n) - \eta_5(n))\} \\
&- 2\,\mathrm{E}\{(b_0x(n) + b_1x(n-1) + b_2x(n-2))(a_1y(n-1) + a_2y(n-2))\} \\
&- 2\,\mathrm{E}\{(\eta_1(n) + \eta_2(n) + \eta_3(n) - \eta_4(n) - \eta_5(n))(a_1y(n-1) + a_2y(n-2))\}.
\end{aligned} \tag{4.88}$$

Analisando termo a termo,

$$\mathrm{E}\{(b_0x(n) + b_1x(n-1) + b_2x(n-2))^2\} = \sigma_x^2(b_0^2 + b_1^2 + b_2^2), \tag{4.89}$$

já que $x(n)$ é iid. De forma semelhante,

$$\mathrm{E}\{(\eta_1(n) + \eta_2(n) + \eta_3(n) - \eta_4(n) - \eta_5(n))^2\} = \sigma_{\eta_1}^2 + \sigma_{\eta_2}^2 + \sigma_{\eta_3}^2 + \sigma_{\eta_4}^2 + \sigma_{\eta_5}^2, \tag{4.90}$$

devido a independência entre os η_k. Os termos

$$\mathrm{E}\{(b_0x(n)+b_1x(n-1)+b_2x(n-2))(\eta_1(n)+\eta_2(n)+\eta_3(n)-\eta_4(n)-\eta_5(n))\} \tag{4.91}$$

e

$$\mathrm{E}\{(\eta_1(n) + \eta_2(n) + \eta_3(n) - \eta_4(n) - \eta_5(n))(a_1y(n-1) + a_2y(n-2))\} \tag{4.92}$$

resultam em zero. Isso ocorre porque, após a expansão das equações, aparecem esperanças do tipo

$$\mathrm{E}\{x(n-j)\eta_k(n)\} = \mathrm{E}\{x(n-j)\}\,\mathrm{E}\{\eta_k(n)\} = 0 \tag{4.93}$$

e

$$\mathrm{E}\{y(n-j)\eta_k(n)\} = \mathrm{E}\{y(n-j)\}\,\mathrm{E}\{\eta_k(n)\} = 0. \tag{4.94}$$

O termo que considera a correlação entre a entrada e a saída pode ser expan-

dido em

$$\begin{aligned}
&\mathrm{E}\{(b_0x(n)+b_1x(n-1)+b_2x(n-2))(a_1y(n-1)+a_2y(n-2))\}=\\
&b_0a_1\,\mathrm{E}\{x(n)y(n-1)\}+b_1a_1\,\mathrm{E}\{x(n-1)y(n-1)\}+b_2a_1\,\mathrm{E}\{x(n-2)y(n-1)\}\\
&+b_0a_2\,\mathrm{E}\{x(n)y(n-2)\}+b_1a_2\,\mathrm{E}\{x(n-1)y(n-2)\}+b_2a_2\,\mathrm{E}\{x(n-2)y(n-2)\},
\end{aligned} \tag{4.95}$$

onde

$$\begin{aligned}
&b_0a_1\,\mathrm{E}\{x(n)y(n-1)\}=\\
&b_0a_1\,\mathrm{E}\{x(n)(b_0x(n-1)+b_1x(n-2)+b_2x(n-3)-a_1y(n-2)-a_2y(n-3)\\
&\qquad+\eta_1(n-1)+\eta_2(n-1)+\eta_3(n-1)-\eta_4(n-1)-\eta_5(n-1))\}=\\
&\qquad\qquad=-b_0a_1(a_1\,\mathrm{E}\{x(n)y(n-2)\}+a_2\,\mathrm{E}\{x(n)y(n-3)\}).
\end{aligned} \tag{4.96}$$

As esperanças $\mathrm{E}\{x(n)y(n-2)\}$ e $\mathrm{E}\{x(n)y(n-3)\}$ também podem ser escritas em função de versões anteriores de $y(n)$, e isso pode ser realizado continuamente, até que (4.96) esteja em função apenas de $\mathrm{E}\{x(n)y(0)\}$ e $\mathrm{E}\{x(n)y(-1)\}$. Nesse caso,

$$\mathrm{E}\{x(n)y(0)\}=\mathrm{E}\{x(n)\}\,\mathrm{E}\{y(0)\}=0 \tag{4.97}$$

e

$$\mathrm{E}\{x(n)y(-1)\}=\mathrm{E}\{x(n)\}\,\mathrm{E}\{y(-1)\}=0, \tag{4.98}$$

já que os filtros são assumidos causais e nos instantes 0 e -1 é sabido que $y(0)=y(-1)=0$, independentemente de $x(n)$. Com isso, $b_0a_1\,\mathrm{E}\{x(n)y(n-1)\}=0$.

Usando um argumento semelhante para os demais elementos de (4.95), obtém-se

$$\begin{aligned}
&b_0a_1\,\mathrm{E}\{x(n)y(n-1)\} = 0\\
&b_1a_1\,\mathrm{E}\{x(n-1)y(n-1)\} = b_0b_1a_1\sigma_x^2\\
&b_2a_1\,\mathrm{E}\{x(n-2)y(n-1)\} = (b_1b_2a_1 - b_0b_2a_1^2)\sigma_x^2\\
&b_0a_2\,\mathrm{E}\{x(n)y(n-2)\} = 0\\
&b_1a_2\,\mathrm{E}\{x(n-1)y(n-2)\} = 0\\
&b_2a_2\,\mathrm{E}\{x(n-2)y(n-2)\} = b_0b_2a_2\sigma_x^2.
\end{aligned} \tag{4.99}$$

O cálculo de $\mathrm{E}\{(a_1y(n-1)+a_2y(n-2))^2\}$ fornece

$$\begin{aligned}
\mathrm{E}\{(a_1y(n-1)+a_2y(n-2))^2\} = \;& a_1^2\,\mathrm{E}\{y^2(n-1)\} + a_2^2\,\mathrm{E}\{y^2(n-2)\}\\
&+2a_1a_2\,\mathrm{E}\{y(n-1)y(n-2)\}.
\end{aligned} \tag{4.100}$$

Mas

$$\begin{aligned}
\mathrm{E}\{y(n-1)y(n-2)\} = \mathrm{E}\{&(b_0x(n-1)+b_1x(n-2)+b_2x(n-3)\\
&-a_1y(n-2)-a_2y(n-3)+\eta_1+\eta_2+\eta_3-\eta_4-\eta_5)y(n-2)\},
\end{aligned} \tag{4.101}$$

de onde se obtém uma equação de recorrência

$$\begin{aligned}
\mathrm{E}\{y(n-1)y(n-2)\} = \;& (b_0b_1+b_1b_2-b_0b_2a_1)\sigma_x^2\\
&-a_1\,\mathrm{E}\{y^2(n-1)\} - a_1\,\mathrm{E}\{y(n-2)y(n-3)\}
\end{aligned} \tag{4.102}$$

Juntando todos os resultados, pode-se calcular a variância através da relação de recorrência (4.102) e de

$$\begin{aligned}
\mathrm{E}\{y^2(n)\} = \;& a_1^2\,\mathrm{E}\{y^2(n-1)\} + a_2^2\,\mathrm{E}\{y^2(n-2)\} + 2a_1a_2\,\mathrm{E}\{y(n-1)y(n-2)\}\\
&+\sigma_x^2(b_0^2+b_1^2+b_2^2-2(b_0b_1a_1+b_1b_2a_1-b_0b_2a_1^2+b_0b_2a_2))\\
&+\sigma_{\eta_1}^2+\sigma_{\eta_2}^2+\sigma_{\eta_3}^2+\sigma_{\eta_4}^2+\sigma_{\eta_5}^2,
\end{aligned} \tag{4.103}$$

usando os valores passados de $\mathrm{E}\{y^2(n)\}$ e $\mathrm{E}\{y(n-1)y(n-2)\}$ para iterativamente encontrar a variância em regime (quando $n \to \infty$). Nessa situação, usam-se como

condições iniciais para um filtro causal $y(0) = 0$, $\mathrm{E}\{y(0)\} = 0$, $\mathrm{E}\{y^2(-1)\} = \mathrm{E}\{y^2(0)\} = 0$ e $\mathrm{E}\{y(0)y(-1)\} = 0$. Por meio das equações de recorrência, a variância dos filtros em regime é calculada e comparada aos valores obtidos pelo modelo com cadeias de Markov, como é apresentado nos exemplos seguintes.

4.7.2.2 Exemplo 1: Filtro passa-tudo com pólos próximos de zero

Para comparar as funções de probabilidade foram usados dois filtros passa-tudo de primeira ordem em cascata, cuja função resultante é dada por

$$H(z) = \frac{0.125 + z^{-1}}{1 + 0.125z^{-1}} \cdot \frac{0.125 + z^{-1}}{1 + 0.125z^{-1}}. \tag{4.104}$$

As figuras 32 e 33 mostram as f.d.p. em uma implementação de 3 bits, considerando as não-linearidades de saturação e de exceção de *overflow*, respectivamente.

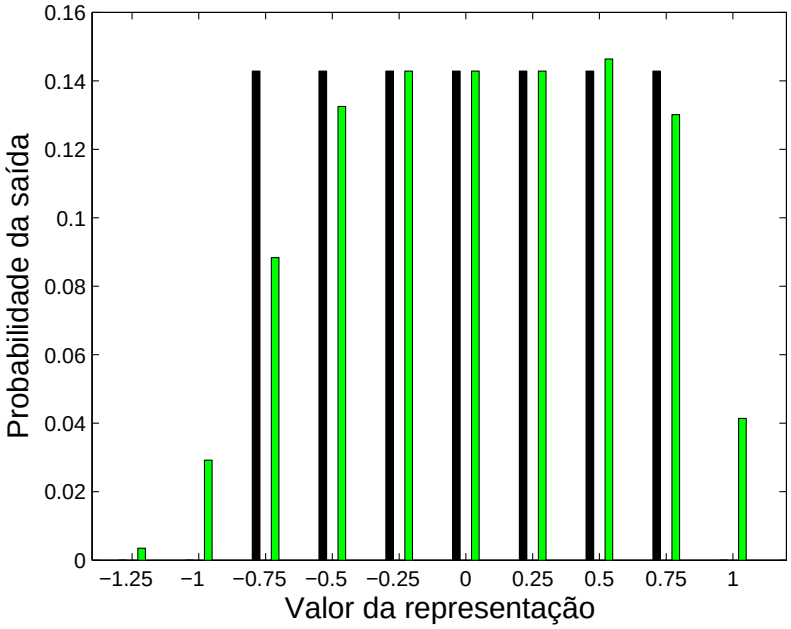

Figura 32: Densidade de probabilidade da saída de um filtro passa-tudo de segunda ordem, calculada com: modelo de Markov de segunda ordem (barras claras) e modelo calculando duas matrizes de transição de primeira ordem (barras escuras). Curvas calculadas assumindo a não-linearidade de saturação.

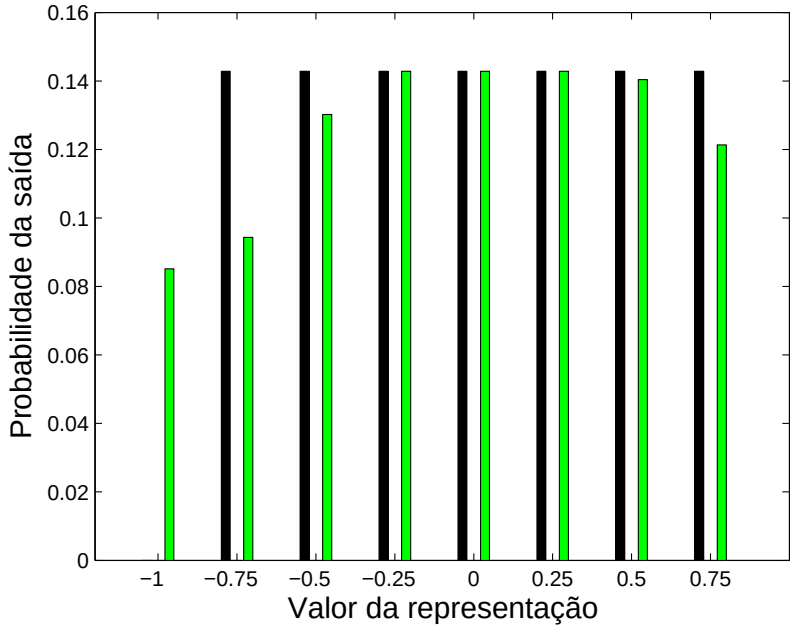

Figura 33: Densidade de probabilidade da saída de um filtro passa-tudo de segunda ordem, calculada com: modelo de Markov de segunda ordem (barras claras) e modelo calculando duas matrizes de transição de primeira ordem (barras escuras). Curvas calculadas assumindo a não-linearidade de *overflow*.

As duas formas de calcular a função de densidade de probabilidade da saída são apresentadas em forma de barras para facilitar a visualização das probabilidades. As funções são plotadas juntas, mas em cores diferentes, com o intuito de facilitar a comparação visual das probabilidades, que são discretas e estão associadas a cada valor possível na representação de 3 bits. As duas barras sobre o valor zero (vide figuras 32 e 33), por exemplo, correspondem à probabilidade de a saída ser igual a zero, segundo o modelo de segunda ordem (barras claras) e usando duas matrizes de transição de estados de primeria ordem (barras escuras). Nas figuras dos próximos exemplos, adota-se a mesma premissa.

Como se pode observar nas figuras 32 e 33, as duas densidades de probabilidade são diferentes. Na figura 32, as probabilidades associadas à saturação são apresentadas como barras adicionais, mostrando a importância da saturação na saída do filtro. As tabelas 2 e 3 apresentam as médias e as variâncias associadas

aos filtros implementados.

Tabela 2: Média e variância para os três modelos, considerando não-linearidade de saturação

	Média	Variância
Modelo de segunda ordem	0.0366	0.2651
Modelo com duas matrizes de Markov	0	0.2500
Modelo linear de segunda ordem	0	0.2569

Tabela 3: Média e variância para os três modelos, considerando não-linearidade por exceção de *overflow*

	Média	Variância
Modelo de segunda ordem	-0.0598	0.2884
Modelo com duas matrizes de Markov	0	0.2500
Modelo linear de segunda ordem	0	0.2569

Comparando a aproximação via cascata de filtros e via modelo de segunda ordem, é fácil notar que diferentes formas de implementação acarretam f.d.p. diferentes. Tanto para a abordagem considerando a não-linearidade de saturação quanto para a abordagem de *overflow*, o modelo de cascata de filtros manteve-se mais próximo da abordagem linearizada do que do modelo de segunda ordem: ambos apresentaram média zero e diferença menor que 3% na variância. Neste exemplo, nota-se que a abordagem de segunda ordem, que prevê média diferente de zero e maior variância, é a única que capta adequadamente os efeitos de exceções. Nesse caso, substituir a análise de segunda ordem pela abordagem com filtros em cascata causa um erro absoluto no valor da média e uma redução na variância medida. A abordagem linearizada apresenta um resultado ainda próximo do modelo de segunda ordem porque os pólos e zeros do filtro são pequenos, reduzindo o efeito das não-linearidades na saída.

4.7.2.3 Exemplo 2: Filtro passa-tudo com pólos próximos da circunferência unitária

Para comparar as funções de probabilidade, foram usados dois filtros passa-tudo de primeira ordem em cascata, cuja função é dada por

$$H(z) = \frac{0.875 + z^{-1}}{1 + 0.875z^{-1}} \cdot \frac{0.875 + z^{-1}}{1 + 0.875z^{-1}}. \tag{4.105}$$

As figuras 34 e 35 mostram as funções de densidade de probabilidade em uma implementação de 3 bits, considerando as não-linearidades de saturação e de *overflow*, respectivamente. As tabelas 4 e 5 mostram as médias e as variâncias

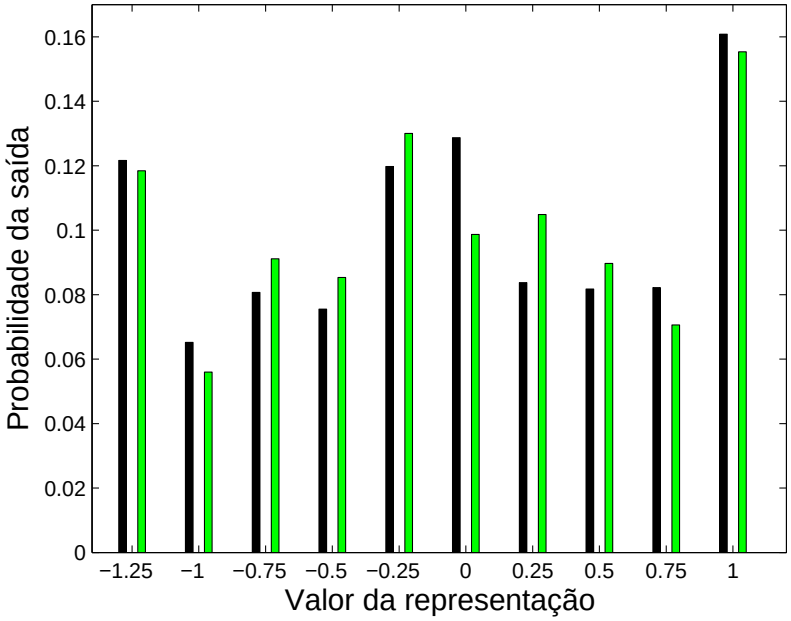

Figura 34: Densidade de probabilidade da saída de um filtro passa-tudo de segunda ordem, calculada com: modelo de Markov de segunda ordem (barras claras) e modelo calculando duas matrizes de transição de primeira ordem (barras escuras). Curvas calculadas assumindo a não-linearidade de saturação.

calculadas para o filtro do exemplo.

Por meio da observação das figuras 34 e 35, nota-se que para as duas não-linearidades consideradas o modelo de cascata de filtros apresenta uma grande proximidade em relação ao modelo de segunda ordem. Para saturação, compa-

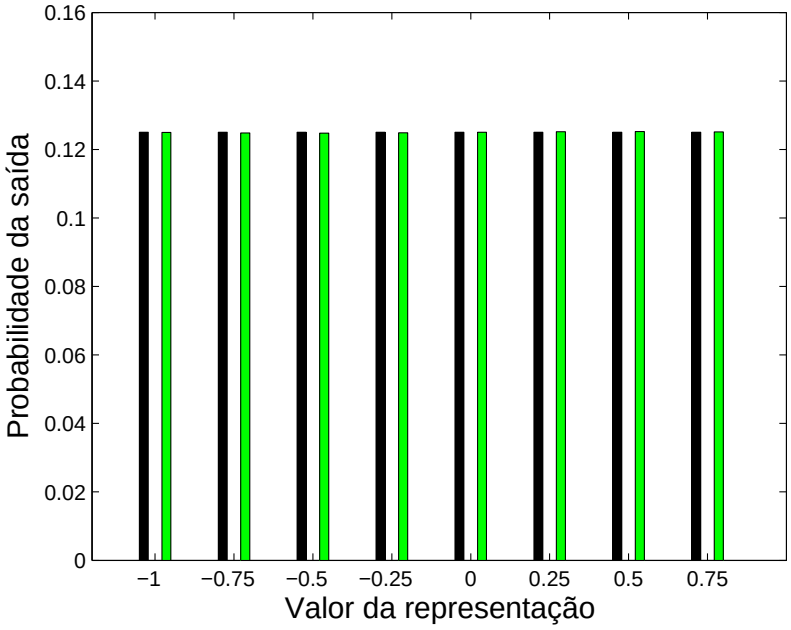

Figura 35: Densidade de probabilidade da saída de um filtro passa-tudo de segunda ordem, calculada com: modelo de Markov de segunda ordem (barras claras) e modelo calculando duas matrizes de transição de primeira ordem (barras escuras). Curvas calculadas assumindo a não-linearidade de *overflow*.

Tabela 4: Média e variância para os três modelos, considerando não-linearidade de saturação

	Média	Variância
Modelo de segunda ordem	-0.0774	0.4052
Modelo com duas matrizes de Markov	-0.0710	0.4159
Modelo linear de segunda ordem	0	1.1428

rando o modelo de segunda ordem e de cascata de filtros, a diferença na média e na variância é de 8 e $2,5\%$, respectivamente, enquanto que para *overflow*, a média difere de $0,4\%$ e a variância de $0,003\%$. Nas duas situações, o modelo linear apresenta média e variância muito distantes dos valores dos modelos de Markov. Para esse exemplo, o modelo de cascata é uma aproximação muito boa para filtros de segunda ordem, principalmente se a não-linearidade for *overflow*.

Tabela 5: Média e variância para os três modelos considerando exceção de *overflow*

	Média	Variância
Modelo de segunda ordem	-0.1245	0.3282
Modelo com duas matrizes de Markov	-0.1250	0.3281
Modelo linear de segunda ordem	0	1.1428

4.7.2.4 Exemplo 3: Filtro passa-tudo com um pólo próximo da circunferência unitária e outro distante

Deseja-se observar a diferença causada pela ordem de concatenação dos filtros na implementação. Nesse caso, o filtro é descrito pela concatenação

$$H(z) = \frac{0.125 + z^{-1}}{1 + 0.125z^{-1}} \cdot \frac{0.875 + z^{-1}}{1 + 0.875z^{-1}} = \frac{0.875 + z^{-1}}{1 + 0.875z^{-1}} \cdot \frac{0.125 + z^{-1}}{1 + 0.125z^{-1}}. \quad (4.106)$$

As funções de probabilidade encontradas, mostrando a diferença provocada pela ordem de concatenação, são apresentadas nas figuras 36 e 37, para não-linearidade de saturação, e nas figuras 38 e 39, para a exceção de *overflow*. São considerados filtros implementados em precisão finita de 3 bits.

As tabelas 6, 7, 8 e 9 comparam as médias e variâncias encontradas quando os filtros são cascateados com o primeiro filtro contendo o menor pólo e o menor zero (indicado pelo índice *menor*) e depois invertendo a posição da cascata (indicado pelo índice *maior* na tabela).

Tabela 6: Média para os três modelos, considerando saturação

	Média_{menor}	Média_{maior}
Modelo de segunda ordem	-0.0491	-0.0491
Modelo com duas matrizes de Markov	-0.0491	-0.0221
Modelo linear de segunda ordem	0	0

Das figuras e das tabelas, fica visível para este exemplo que o modelo de filtros em cascata implementados com não-linearidade de saturação sofre maior influência da ordem de implementação de pólos e zeros do que os filtros que

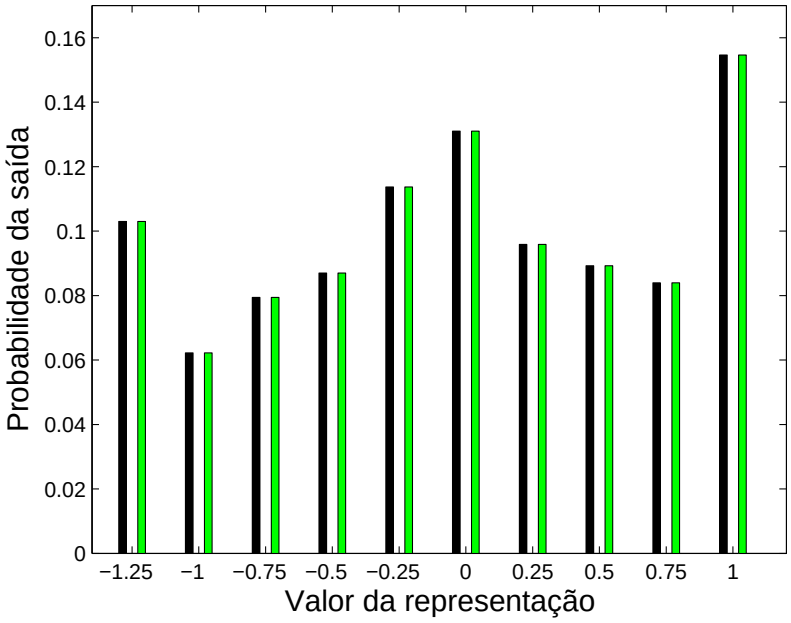

Figura 36: Densidade de probabilidade da saída de um filtro passa-tudo de segunda ordem, calculada com: modelo de Markov de segunda ordem (barras claras) e modelo calculando duas matrizes de transição de primeira ordem (barras escuras). Curvas obtidas para a não-linearidade de saturação, com o filtro de pólo menor implementado primeiro.

Tabela 7: Variância para os três modelos, considerando saturação

	Var_{menor}	Var_{maior}
Modelo de segunda ordem	0.3988	0.3988
Modelo com duas matrizes de Markov	0.3988	0.3913
Modelo linear de segunda ordem	0.2851	0.2851

lidam com as exceções por *overflow*. A análise usando duas matrizes de transição apresenta maior similaridade com o modelo de segunda ordem quando o filtro com pólos e zeros mais próximos da circunferência unitária é implementado no final da cascata. Isso está de acordo com a abordagem tradicionalmente empregada na implementação de filtros em cascata, de implementar primeiro filtros com menores pólos e zeros em uma cascata, com o intuito de reduzir o efeito de não-linearidades [1].

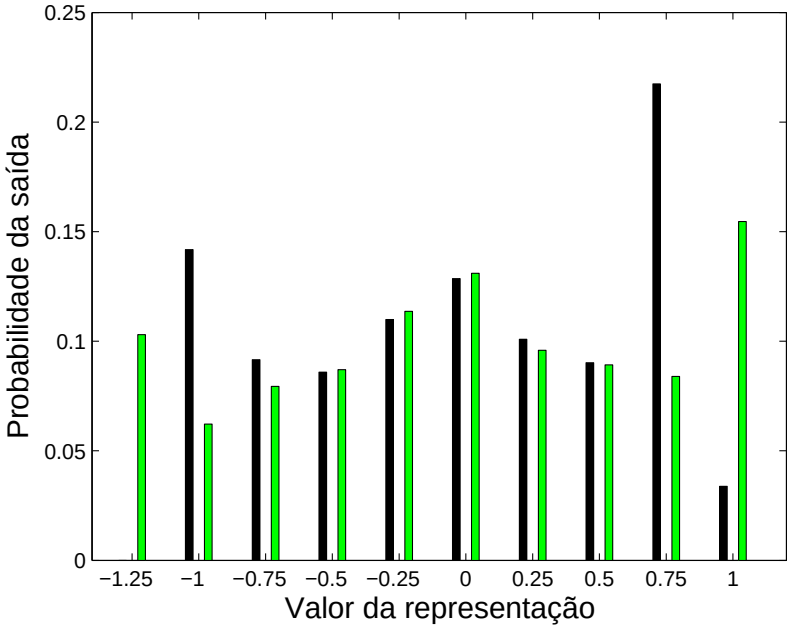

Figura 37: Densidade de probabilidade da saída de um filtro passa-tudo de segunda ordem, calculada com: modelo de Markov de segunda ordem (barras claras) e modelo calculando duas matrizes de transição de primeira ordem (barras escuras). Curvas obtidas para a não-linearidade de saturação, com o filtro de pólo maior implementado primeiro.

Tabela 8: Média para os três modelos, considerando *overflow*

	Média$_{menor}$	Média$_{maior}$
Modelo de segunda ordem	-0.1250	-0.1250
Modelo com duas matrizes de Markov	-0.1250	-0.1250
Modelo linear de segunda ordem	0	0

4.7.3 Exemplos com filtros passa-baixa

Filtros passa-baixa de primeira ordem são descritos por equações no domínio da transformada $\mathbb{Z}$

$$H(z) = \frac{C}{1 - az^{-1}}, \tag{4.107}$$

com $0 < a < 1$ e C correspondendo a uma constante positiva.

Diferentemente de filtros passa-tudo, para filtros passa-baixa não é válido afirmar que, se a entrada for descorrelacionada, a saída também será descorrelacionada. Nesse caso, para dois filtros passa-baixas colocados em cascata, a

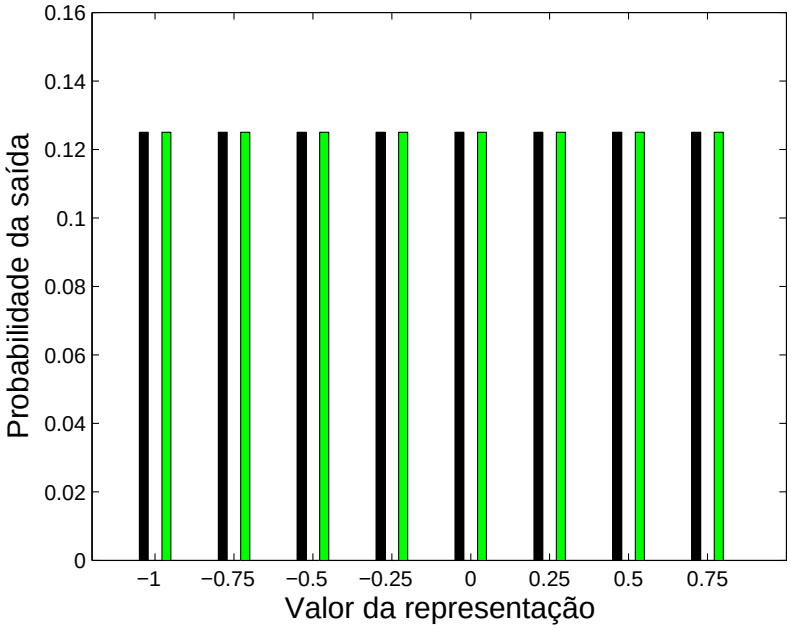

Figura 38: Densidade de probabilidade da saída de um filtro passa-tudo de segunda ordem, calculada com: modelo de Markov de segunda ordem (barras claras) e modelo calculando duas matrizes de transição de primeira ordem (barras escuras). Curvas obtidas para a exceção de *overflow*, com o filtro de pólo menor implementado primeiro.

Tabela 9: Variância para os três modelos, considerando *overflow*

	Var_{menor}	Var_{maior}
Modelo de segunda ordem	0.3281	0.3281
Modelo com duas matrizes de Markov	0.3281	0.3281
Modelo linear de segunda ordem	0.2851	0.2851

entrada do segundo filtro é correlacionada, o que contraria a hipótese usada para o modelo com cadeias de Markov, de que a entrada do filtro é iid.

Por exemplo, suponha dois filtros passa-baixa concatenados, descritos pela equação (4.107), cujas saídas são $y_1(n)$ e $y_2(n)$, respectivamente. Suponha que a entrada do primeiro filtro corresponde a $x_1(n)$, descorrelacionada e de média

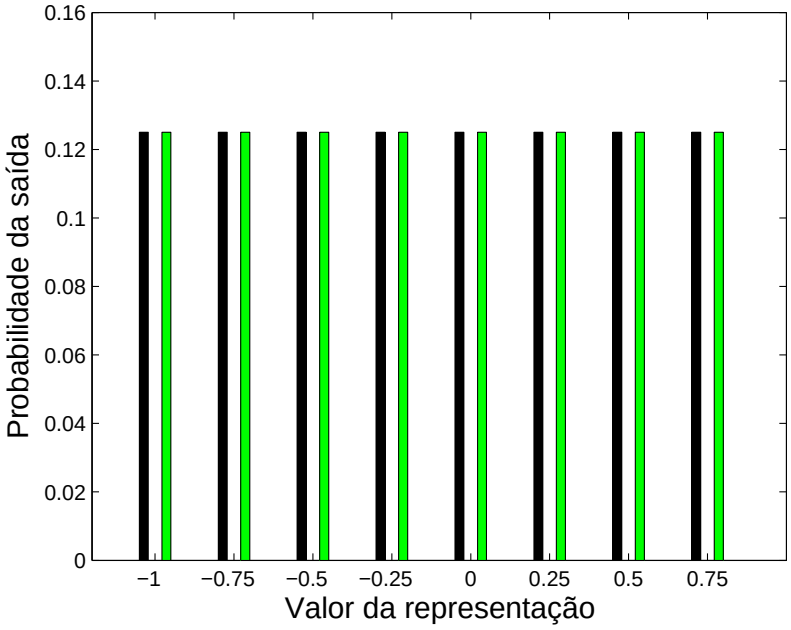

Figura 39: Densidade de probabilidade da saída de um filtro passa-tudo de segunda ordem, calculada com: modelo de Markov de segunda ordem (barras claras) e modelo calculando duas matrizes de transição de primeira ordem (barras escuras). Curvas obtidas para a exceção de *overflow*, com o filtro de pólo maior implementado primeiro.

zero. Pode-se calcular a correlação da saída do primeiro filtro como

$$\begin{aligned}
&\mathrm{E}\{y_1(n)y_1(n-i)\} = \\
&a\,\mathrm{E}\{y_1(n-1)y_1(n-i)\} + C\,\mathrm{E}\{x_1(n-1)y-1(n-i)\} = \\
&a^2\,\mathrm{E}\{y_1(n-2)y_1(n-i\} + C\,[a\,\mathrm{E}\{x_1(n-2)y_1(n-i)\} + \mathrm{E}\{x_1(n-1)y_1(n-i)\}] = \\
&\vdots \\
&a^i\,\mathrm{E}\{y_1^2(n-i)\} + C\sum_{j=0}^{i-1} a^j\,\mathrm{E}\{x_1(n-j)y_1(n-i)\}.
\end{aligned} \tag{4.108}$$

Cada elemento da somatória $\mathrm{E}\{x_1(n-j)y_1(n-i)\}$ pode ser expandido como

$$\begin{aligned}
&\mathrm{E}\{x_1(n-j)y_1(n-i)\} = \\
&a\,\mathrm{E}\{x_1(n-j)y_1(n-i-1)\} + C\,\mathrm{E}\{x_1(n-j)x_1(n-i)\} = \\
&a^2\,\mathrm{E}\{x_1(n-j)y_1(n-i-2)\} + C\,[a\,\mathrm{E}\{x_1(n-j)x_1(n-i-1)\} + \mathrm{E}\{x_1(n-j)x_1(n-i)\}] = \\
&\vdots \\
&a^K\,\mathrm{E}\{x_1(n-j)y_1(n-i-K)\} + C\left[\sum_{k=1}^{K} a^{k-1}\,\mathrm{E}\{x_1(n-j)x_1(n-i-k+1)\}\right].
\end{aligned} \tag{4.109}$$

Nesse caso, pode-se expandir até o ponto em que $y_1(n-i-K) = y_1(0)$. Nesse

ponto,

$$\mathrm{E}\{x_1(n-j)y_1(0)\} = \mathrm{E}\{x_1(n-j)\}\,\mathrm{E}\{y_1(0)\} = 0 \tag{4.110}$$

e

$$\mathrm{E}\{x_1(n-j)x_1(n-i-k+1)\} = r(i+k+j-1), \tag{4.111}$$

em que $i \geq 1$, $0 \leq j < i$ e $k \geq 1$, de onde vem que $i+k+j-1 > 0$. Mas a entrada do filtro, por hipótese, é iid, o que implica que $r(\alpha) = 0$, $\forall \alpha \neq 0$. Portanto, a equação (4.108) pode ser simplificada por

$$\mathrm{E}\{y_1(n)y_1(n-i)\} = a^i\,\mathrm{E}\{y_1^2(n-i)\}, \tag{4.112}$$

de onde fica claro que a saída do primeiro filtro é correlacionada, tornando a abordagem via cadeias de Markov para o segundo filtro da cascata (segundo o desenvolvido ao longo desse texto) imprecisa. Contudo, analisando (4.112), nota-se que a correlação é dependente do pólo filtro. Se for escolhido um filtro com pólo pequeno, próximo de zero, a correlação diminui e pode-se aplicar o modelo de cascata de filtros, como uma forma aproximada. Os exemplos seguintes apresentam filtros passa-baixa implementados com pólos próximos de zero e próximos da circunferência unitária. Os resultados obtidos são comparados com modelo linear.

4.7.3.1 Modelo linear para filtro passa-baixa

De forma semelhante ao usado em (4.102) e (4.103), podem ser encontradas equações de recorrência para calcular a variância da saída do filtro implementado com o modelo linear, por meio das equações

$$\mathrm{E}\{y_1(n-1)y_1(n-2)\} = 2a\,\mathrm{E}\{y_1^2(n-2)\} - a^2\,\mathrm{E}\{y_1(n-2)y_1(n-3)\} \tag{4.113}$$

e

$$\mathrm{E}\{y_1^2(n)\} = C^4\sigma_x^2 + 4a^2\,\mathrm{E}\{y_1^2(n-1)\} + a^4\,\mathrm{E}\{y_1^2(n-2)\} - 4a^3\,\mathrm{E}\{y_1(n-1)y_1(n-2)\}, \tag{4.114}$$

usando como condições iniciais as esperanças $\mathrm{E}\{y_1^2(n-1)\} = 0$, $\mathrm{E}\{y_1^2(n-2)\} = 0$ e $\mathrm{E}\{y_1(0)y_1(-1)\} = 0$.

4.7.3.2 Exemplo 1: Filtro passa-baixas com pólos próximos de zero

Neste exemplo, dois filtros passa-baixa de primeira ordem, descritos pela equação

$$H(z) = \frac{0.75}{1 - 0.25z^{-1}}, \tag{4.115}$$

são implementados com 3 bits, assumindo que a entrada do primeiro filtro possui distribuição de probabilidade uniforme descorrelacionada, de média 0 e variância 0.25 (calculadas a partir dos valores que a entrada pode assumir na representação discreta, entre -1 e 0.75), discretizada segundo o apresentado na seção 2.7.3 (pág. 25). As figuras 40 e 41 apresentam as densidades de probabilidade quando a não-linearidade é a saturação e a exceção por *overflow*, respectivamente.

Visualmente, comparando as funções de probabilidade, é possível notar que a densidade de probabilidade calculada com o auxílio de duas matrizes de transição de estados fornece um resultado distinto, mas que ainda apresenta semelhanças com o obtido com o modelo de segunda ordem (note que como os pólos são pequenos, as não-linearidades têm menor influência na forma final da função de probabilidade). As tabelas 10 e 11 apresentam uma comparação entre a média e a variância das funções obtidas.

Para as duas não-linearidades consideradas, o modelo usando a cascata de filtros apresenta diferenças razoáveis em relação ao modelo de segunda ordem, embora ainda seja uma abordagem melhor do que o modelo linear, com relação

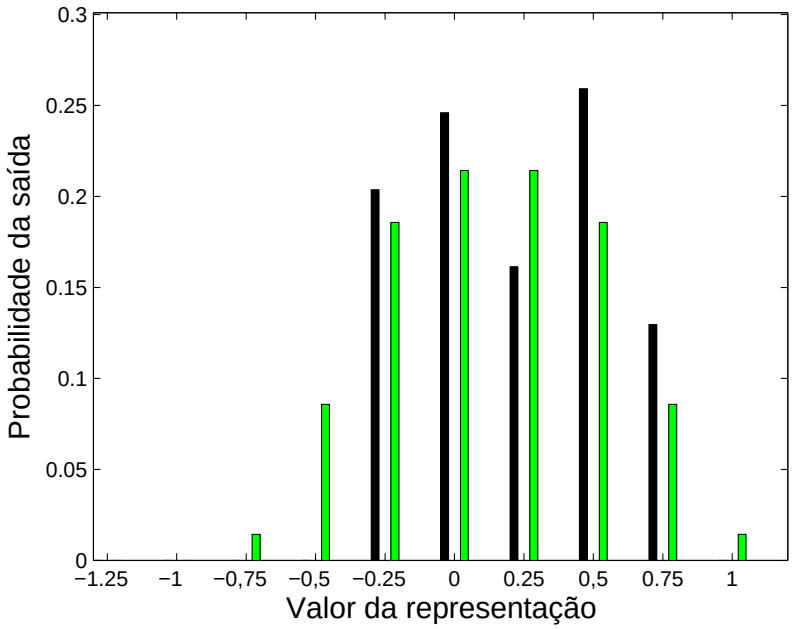

Figura 40: Função densidade de probabilidade de um filtro passa-baixa com um pólo próximos de zero: implementação de segunda ordem (barras claras) e usando duas matrizes de transição de estados (barras escuras). Curvas obtidas para a não-linearidade de saturação.

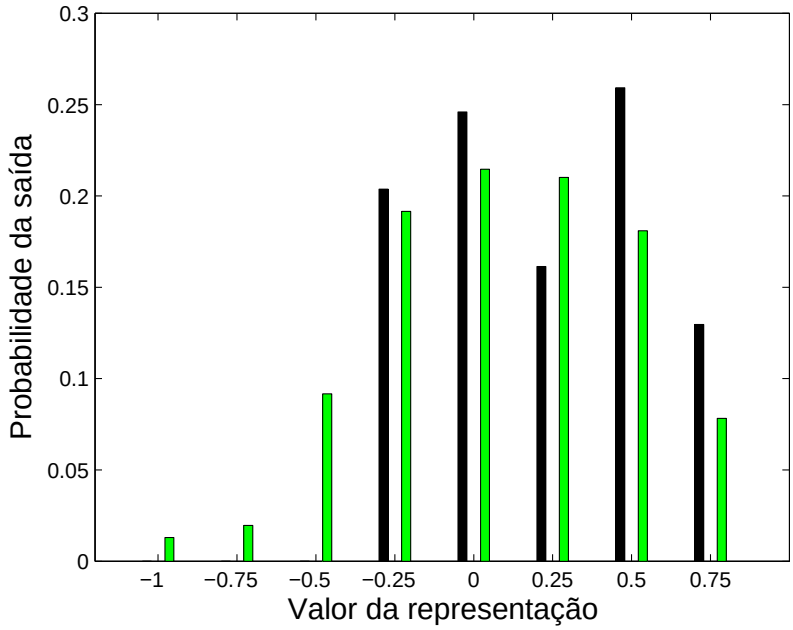

Figura 41: Função densidade de probabilidade de um filtro passa-baixa com um pólo próximos de zero: implementação de segunda ordem (barras claras) e usando duas matrizes de transição de estados (barras escuras). Curvas obtidas para a exceção de *overflow*.

Tabela 10: Média e variância para os três modelos considerando saturação

	Média	Variância
Modelo de segunda ordem	0.1214	0.1424
Modelo com duas matrizes de Markov	0.2163	0.1138
Modelo linear de segunda ordem	0	0.1070

Tabela 11: Média e variância para os três modelos considerando exceção de *overflow*

	Média	Variância
Modelo de segunda ordem	0.0802	0.1549
Modelo com duas matrizes de Markov	0.2163	0.1138
Modelo linear de segunda ordem	0	0.1070

à variância da saída (com relação ao modelo de segunda ordem, a variância da cascata de filtros difere de 20% para saturação e 27% para *overflow*, enquanto o modelo linear difere de 25% para saturação e 31% para *overflow*). Se fosse desejado um cálculo mais preciso no modelo de cascata de filtros, seria necessário usar um modelo de cadeias de Markov que considerasse a correlação na entrada do segundo filtro, o que iria introduzir mais estados, relacionando a entrada e a saída dos filtros, o que acarretaria mais cálculos e maior complexidade computacional.

4.7.3.3 Exemplo 2: Filtro passa-baixas com pólos próximos da circunferência unitária

Para observar as funções de probabilidade obtidas para filtros com pólos próximos da circunferência unitária, dois filtros passa-baixa de primeira ordem, descritos cada um por

$$H(z) = \frac{0.25}{1 - 0.75z^{-1}}, \tag{4.116}$$

são implementados com 3 bits. Assume-se que a entrada do primeiro filtro possui distribuição de probabilidade uniforme, de média 0 e variância 0.25, da mesma maneira como foi usado na seção 4.7.3.2. As figuras 42 e 43 apresentam as densidades de probabilidade quando a não-linearidade é a saturação e a exceção

por *overflow*, respectivamente.

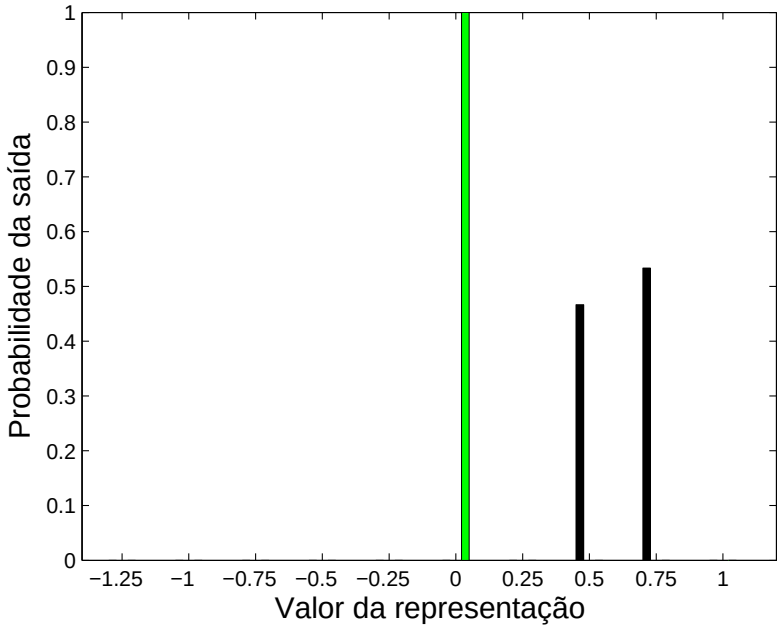

Figura 42: Função densidade de probabilidade de um filtro passa-baixa com pólos próximos da circunferência unitária: implementação de segunda ordem (barras claras) e usando duas matrizes de transição de estados (barras escuras). Curvas obtidas para a não-linearidade de saturação.

Nas figuras 42 e 43 fica claro que as funções de probabilidade obtidas são bem distintas da curva obtida com o modelo de segunda ordem. Nesse caso, o efeito das não-linearidades é mais pronunciado (devido aos pólos mais próximos da circunferência unitária) e os modelos de filtros em cascata se tornam inadequados para a análise, conforme previsto anteriormente.

4.7.4 Comentários finais da análise com filtros concatenados

Nesta seção foram analisados filtros passa-tudo e passa-baixa de segunda ordem implementados como uma cascata de filtros de primeira ordem. Foram comparados o modelo de segunda ordem do filtro todo, o modelo linearizado, e um modelo aproximado, em que cada filtro da cascata é modelado por uma cadeia de Markov própria. O intuito foi procurar um modelo simplificado, em

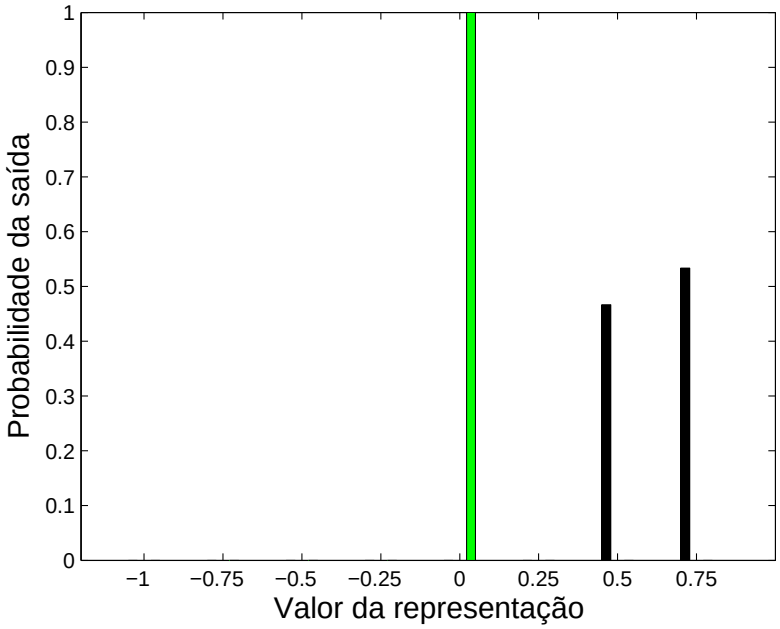

Figura 43: Função densidade de probabilidade de um filtro passa-baixa com pólos próximos da circunferência unitária: implementação de segunda ordem (barras claras) e usando duas matrizes de transição de estados (barras escuras). Curvas obtidas para a exceção de *overflow*.

que o número de estados das cadeias de Markov não aumente exageradamente com a ordem do filtro completo.

Para filtros passa-tudo, os exemplos apresentados mostraram que o uso do modelo da cascata tem resultados algumas vezes semelhantes aos do modelo linear (no caso de pólos próximos de zero) e algumas vezes melhores e mais próximos dos obtidos com o modelo de segunda ordem (no exemplo com pólos próximos da circunferência unitária). Os exemplos variando a posição dos pólos e zeros com relação à ordem de implementação mostraram que estruturas em que os pólos e zeros menores são implementados primeiro apresentam f.d.p. de saída mais semelhante à f.d.p. do modelo de segunda ordem.

Para filtros passa-baixa, os exemplos mostraram que o modelo de cascata de filtros não é adequado para substituir o modelo de segunda ordem: mesmo quando os pólos são pequenos, as f.d.p. dos modelos apresentam grandes diferenças.

Para resultados mais conclusivos sobre a aplicação do modelo de cascata de filtros, seria necessário realizar mais simulações. Considerando os exemplos aqui apresentados, pode-se dizer que a aproximação não é pior do que a abordagem linear e, às vezes, chega a ser bem melhor, como no caso de filtros passa-tudo. Portanto, ainda são necessárias mais simulações para definir as limitações exatas do modelo simplificado.

5 FILTROS DIGITAIS COM ACUMULADOR DE PRECISÃO DUPLA

Em geral, filtros digitais costumam ser implementados em DSPs e computadores com acumulador de precisão suficiente para que a quantização possa ser realizada após a soma de diversos elementos, de forma que as equações (4.1) e (4.2) (vide pág. 65) podem ser simplificadas por

$$y(n) = Q[b_0x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2)] \quad (5.1)$$

e

$$y(n) = Q[b_0x(n) + b_1x(n-1) - a_1y(n-1)]. \quad (5.2)$$

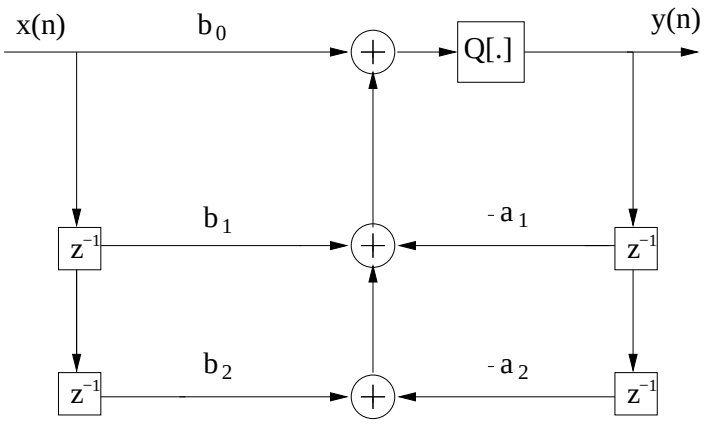

Figura 44: Filtro IIR de segunda ordem implementado na forma direta I, mostrando não-linearidades de precisão finita

Nessa situação, o cálculo teórico e a programação em linguagem *Matlab* ficam facilitadas, já que existe apenas uma operação de quantização.

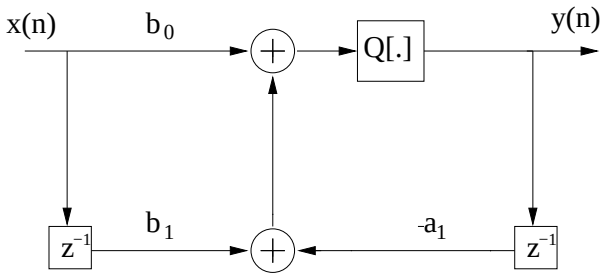

Figura 45: Filtro IIR de primeira ordem implementado na forma direta I, mostrando não-linearidades de precisão finita

5.1 Probabilidades associadas ao filtro de segunda ordem, assumindo saturação

Se a entrada do filtro tiver uma função de densidade de probabilidade discreta, tal como na equação (4.3), ou seja,

$$f_x(x(n)) = \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \delta(x(n) - k\Delta), \tag{5.3}$$

e se $x(n)$, $x(n-1)$ e $x(n-2)$ forem estatísticas independentes, a função de densidade de probabilidade da soma de $x_0 = b_0 x(n)$, $x_1 = b_1 x(n-1)$ e $x_2 = b_2 x(n-2)$, que para simplificar notação será escrita $x_s = x_0 + x_1 + x_2$, será dada por

$$f(x_s) = f(x_0) * f(x_1) * f(x_2), \tag{5.4}$$

o que fornece

$$f(x_s) = \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \sum_{l=-2^{(B-1)}}^{2^{(B-1)}-1} \sum_{m=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \gamma_l \gamma_m \delta(x_s - [k+l+m]\Delta). \tag{5.5}$$

Se for lembrado que as versões passadas de $y(n)$ estão quantizadas, pode-se escrever $y(n-1) = h\Delta$ e $y(n-2) = j\Delta$ (com h, j inteiros e entre -2^{B-1} e $2^{B-1} - 1$). Com isso, a função de probabilidade de $P(y(n) = i\Delta | y(n-1), y(n-2))$ pode ser calculada por

$$P(y(n) = i\Delta|h\Delta, j\Delta) = P(i\Delta - 0.5\Delta \leq y(n)) < i\Delta + 0.5\Delta|h\Delta, j\Delta)$$
$$= P(i\Delta - 0.5\Delta \leq x_s - a_1 h\Delta - a_2 j\Delta < i\Delta + 0.5\Delta|h\Delta, j\Delta). \quad (5.6)$$

Escrevendo em termos de (5.5) e considerando a não-linearidade de saturação, obtém-se

$$P(y(n) = i\Delta|h\Delta, j\Delta) = \begin{cases} \int_{-\infty}^{L_{S_{sat}}} f(x_s)dx_s & \text{se } i\Delta = -1 \\ \int_{L_I}^{L_S} f(x_s)dx_s & \text{se } -1 < i\Delta < 1 - \Delta \\ \int_{L_{I_{sat}}}^{\infty} f(x_s)dx_s & \text{se } i\Delta = 1 - \Delta \end{cases} \quad (5.7)$$

onde L_I, L_S, $L_{I_{sat}}$ e $L_{S_{sat}}$ correspondem a

$$L_I = a_1 h\Delta + a_2 j\Delta + i\Delta - 0.5\Delta,$$
$$L_S = a_1 h\Delta + a_2 j\Delta + i\Delta + 0.5\Delta,$$
$$L_{I_{sat}} = a_1 h\Delta + a_2 j\Delta + 1 - 1.5\Delta$$

e

$$L_{S_{sat}} = a_1 h\Delta + a_2 j\Delta - 1 + 0.5\Delta,$$

respectivamente.

Finalmente, a função de probabilidade fica definida por

$$f(y(n)|y(n-1) = h\Delta, y(n-2) = j\Delta) =$$
$$= \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{h=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} P(y(n) = i\Delta|h\Delta, j\Delta)\delta(y(n) - i\Delta), \quad (5.8)$$

encerrado a determinação da probabilidade condicionada para um filtro de segunda ordem.

5.2 Probabilidades associadas ao filtro de primeira ordem, assumindo saturação

Para um filtro de primeira ordem, os cálculos para a determinação de $f(y(n)|y(n-1))$ são similares aos realizados na seção 5.1. A probabilidade associada à soma de $x_0 = b_0 x(n)$ e $x_1 = b_1 x(n-1)$ corresponde à convolução

$$f(x_s) = f(x_0) * f(x_1) = \sum_{k=-2^{(B-1)}}^{2^{(B-1)}-1} \sum_{l=-2^{(B-1)}}^{2^{(B-1)}-1} \gamma_k \gamma_l \delta(x_s - [k+l]\Delta), \tag{5.9}$$

em que $x_s = x_0 + x_1$. As probabilidades $P(y(n) = i\Delta | y(n-1) = h\Delta)$ são calculadas por

$$P(y(n) = i\Delta | h\Delta) = \begin{cases} \int_{-\infty}^{L_{S_{sat}}} f(x_s)dx_s & \text{se } i\Delta = -1 \\ \int_{L_I}^{L_S} f(x_s)dx_s & \text{se } -1 < i\Delta < 1-\Delta \\ \int_{L_{I_{sat}}}^{\infty} f(x_s)dx_s & \text{se } i\Delta = 1-\Delta \end{cases} \tag{5.10}$$

onde L_I, L_S, $L_{I_{sat}}$ e $L_{S_{sat}}$ correspondem a

$$L_I = a_1 h\Delta + i\Delta - 0.5\Delta,$$
$$L_S = a_1 h\Delta + i\Delta + 0.5\Delta,$$
$$L_{I_{sat}} = a_1 h\Delta + 1 - 0.5\Delta$$

e

$$L_{S_{sat}} = a_1 h\Delta - 1 + 0.5\Delta,$$

respectivamente. Com isso, pode-se definir a função de probabilidade condicionada como

$$f(y(n)|y(n-1) = h\Delta) = \\ = \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{h=-2^{B-1}}^{2^{B-1}-1} \sum_{k=-2^{B-1}}^{2^{B-1}-1} P(y(n) = i\Delta | h\Delta)\delta(y(n) - i\Delta), \tag{5.11}$$

encerrando os cálculos.

5.3 Probabilidades associadas ao filtro de segunda ordem, assumindo exceção de *overflow*

Quando se considera exceções de *overflow* em um filtro de segunda ordem, implementado como descrito pela equação (5.1), a probabilidade de $P(y(n) = i\Delta | y(n-1) = h\Delta, y(n-2) = j\Delta)$ será calculada por

$$P(y(n) = i\Delta | h\Delta, j\Delta) = \sum_{k=-\infty}^{\infty} P(i\Delta - 0.5\Delta + 2k \leq y(n) < i\Delta + 0.5\Delta + 2k)$$

$$= \sum_{k=-\infty}^{\infty} P(i\Delta - 0.5\Delta + 2k \leq x_s - a_1 h\Delta - a_2 j\Delta < i\Delta + 0.5\Delta + 2k), \quad (5.12)$$

em que $x_s = x_0 + x_1 + x_2$ (com $x_0 = b_0 x(n)$, $x_1 = b_1 x(n-1)$ e $x_2 = b_2 x(n-2)$) e a densidade de probabilidade $f(x_s)$ é obtida por meio de (5.5). Para determinar cada $P(y(n) = i\Delta | h\Delta, j\Delta)$ basta fazer

$$P(y(n) = i\Delta | h\Delta, j\Delta) = \sum_{k=-\infty}^{\infty} \int_{L_I}^{L_S} f(x_s) dx_s, \quad (5.13)$$

com

$$L_I = a_1 h\Delta + a_2 j\Delta + i\Delta - 0.5\Delta + 2k$$

e

$$L_S = a_1 h\Delta + a_2 j\Delta + i\Delta + 0.5\Delta + 2k.$$

Dessa forma,

$$f(y(n) | y(n-1) = h\Delta, y(n-2) = j\Delta) =$$

$$= \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{h=-2^{B-1}}^{2^{B-1}-1} \sum_{j=-2^{B-1}}^{2^{B-1}-1} P(y(n) = i\Delta | h\Delta, j\Delta) \delta(y(n) - i\Delta), \quad (5.14)$$

definindo a função de densidade de probabilidade.

5.4 Probabilidades associadas ao filtro de primeira ordem, assumindo exceção de *overflow*

Semelhante ao filtro de primeira ordem considerando saturação, calcula-se a função de probabilidade da equação (5.9). A probabilidade $P(y(n) = i\Delta|y(n-1) = h\Delta)$ pode ser escrita em temos de $x_s = b_0x(n) + b_1x(n-1)$ e $h\Delta$, isto é,

$$P(y(n) = i\Delta|h\Delta) = \sum_{k=-\infty}^{\infty} P(i\Delta - 0.5\Delta + 2k \leq x_s - a_1h\Delta < i\Delta + 0.5\Delta + 2k), \tag{5.15}$$

que resulta em

$$P(y(n) = i\Delta|h\Delta) = \sum_{k=-\infty}^{\infty} \int_{L_I}^{L_S} f(x_s)dx_s, \tag{5.16}$$

com

$$L_I = a_1h\Delta + i\Delta - 0.5\Delta + 2k$$

e

$$L_S = a_1h\Delta + i\Delta + 0.5\Delta + 2k.$$

Dessa forma,

$$f(y(n)|y(n-1) = h\Delta) = \\ = \sum_{i=-2^{B-1}}^{2^{B-1}-1} \sum_{h=-2^{B-1}}^{2^{B-1}-1} P(y(n) = i\Delta|h\Delta)\delta(y(n) - i\Delta), \tag{5.17}$$

definindo a função de densidade de probabilidade.

5.5 Simulações

As análises realizadas no capítulo 4 são também aplicáveis para filtros implementados com acumulador maior. Portanto, é possível encontrar a matriz de transição de estados (com ou sem a introdução de estados de saturação, ou supondo a exceção de *overflow*) e usá-la para escalar o sinal de entrada. Além

disso, também é possível comparar a média e a variância do modelo de cadeias de Markov com o modelo teórico linearizado, o que é apresentado a seguir.

5.5.1 Exemplo considerando a não-linearidade de saturação

Supondo o mesmo filtro de primeira ordem do Exemplo 1 (pág. 93), descrito pela equação

$$y(n) = Q[x(n) - 0.75y(n-1)],$$

com sinais de $B = 2$ bits e com uma entrada descorrelacionada, de média nula e distribuição uniforme, a matriz de transição de estados estendida $\mathbb{P}$ é dada por

$$\begin{array}{c} \\ \mathbb{P} = \end{array} \begin{array}{c} \begin{array}{cccccc} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.333 & 0.333 \\ 0 & 0 & 0 & 0.333 & 0.333 & 0.333 \\ 0 & 0 & 0.333 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0.333 & 0 & 0 \\ 0.667 & 0.667 & 0.333 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array} .$$

A matriz de estado estacionário equivalente corresponde a

$$\begin{array}{c} \\ \mathbb{P}^{\infty} = \end{array} \begin{array}{c} \begin{array}{cccccc} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.122 & 0.122 & 0.122 & 0.122 & 0.122 & 0.122 \\ 0.220 & 0.220 & 0.220 & 0.220 & 0.220 & 0.220 \\ 0.293 & 0.293 & 0.293 & 0.293 & 0.293 & 0.293 \\ 0.211 & 0.211 & 0.211 & 0.211 & 0.211 & 0.211 \\ 0.155 & 0.155 & 0.155 & 0.155 & 0.155 & 0.155 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array} .$$

Procurando iterativamente um fator de escala para a entrada, encontra-se $m_{esc} = 0.375$ (escrito com 3 bits), de onde se obtém

$$\mathbb{P}^{\infty} = \begin{array}{c} \begin{array}{cccccc} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1,000 & 1,000 & 1,000 & 1,000 & 1,000 & 1,000 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array} .$$

O fator de escala $m_{esc} = 0.375$ encontrado é menor do que o obtido via critério de normas L_P (vide exemplo 1, pág. 93), demonstrando que o modelo via cadeias de Markov pode ser vantajoso para fazer o escalamento da entrada e evitar a saturação da saída. A figura 46 ressalta a diferença entre o modelo linearizado

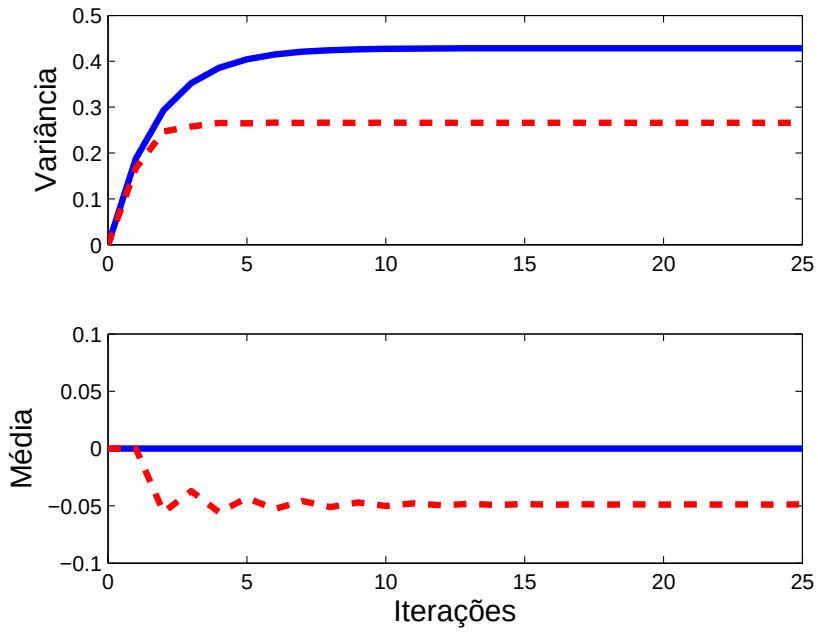

Figura 46: Variância e média via cadeias de Markov (linha pontilhada) e via modelo linearizado (linha contínua), considerando a não-linearidade de saturação

e via cadeias de Markov, onde é possível observar a diferença entre a média e a

variância da saída. As equações (4.72) e (4.73) (vide pág. 98) foram usadas para obter a curva do modelo via cadeias de Markov. A curva da variância do modelo linearizado corresponde à calculada com (4.71) (pág. 97).

5.5.2 Exemplo considerando a exceção de *overflow*

Considerando o mesmo filtro usado no exemplo da seção 5.5.1, e supondo as mesmas condições (sinais de 2 bits e distribuição de entrada uniforme, descorrelacionada e de média nula), deseja-se encontrar a matriz de transição de estados correspondente.

Para um filtro de primeira ordem, implementado segundo a equação (5.2), o valor da saída ($y(n)$) antes da quantização pode alcançar valores entre $-3+3\Delta$ e 3. Esses valores podem ser representados no intervalo $[-3+3\Delta,\ 3]$, com passo de quantização de tamanho Δ (supondo que os sinais e coeficientes são representados por valores entre -1 e $1-\Delta$). Os valores encontrados podem ser, então, usados como os estados de uma matriz de transição de estados expandida. Com essa matriz, tornam-se visíveis as probabilidades de ocorrerem exceções de *overflow* a cada instante n. (De fato, esse processo de escrever a matriz de transição de estados expandida corresponde a não usar a somatória na equação (5.13) e usar as probabilidades das situações de exceção como estados adicionais na cadeia.)

Para o filtro desejado, a matriz expandida corresponde a

$$\mathbb{P}_{exp} = \begin{array}{c} \begin{matrix} -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 & 2 & 2.5 & 3 \end{matrix} \\ \left[\begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.333 & 0 & 0 & 0 & 0.333 & 0 & 0 & 0 & 0.333 & 0 \\ 0.333 & 0 & 0 & 0.333 & 0.333 & 0 & 0 & 0.333 & 0.333 & 0 \\ 0.3333 & 0 & 0.333 & 0.333 & 0.333 & 0 & 0.333 & 0.333 & 0.333 & 0 \\ 0 & 0.333 & 0.333 & 0.333 & 0 & 0.333 & 0.333 & 0.333 & 0 & 0.333 \\ 0 & 0.333 & 0.333 & 0 & 0 & 0.333 & 0.333 & 0 & 0 & 0.333 \\ 0 & 0.333 & 0 & 0 & 0 & 0.333 & 0 & 0 & 0 & 0.333 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.5 \\ -1 \\ -0.5 \\ 0 \\ 0.5 \\ 1 \\ 1.5 \\ 2 \\ 2.5 \\ 3 \end{array} .$$

Em $\mathbb{P}_{exp}$, por exemplo, o elemento $p_{6,2} = 0.333$ corresponde a $P(y(1) = 1|y(0) = -1)$. Como 1 não está contido na representação de $B = 2$ bits, a exceção de *overflow* representa esse valor como -1. Como na matriz de transição de estados tradicional não existe diferenciação entre as probabilidades, não é possível distinguir o efeito da não-linearidade, o que pode tornar atrativo o uso de $\mathbb{P}_{exp}$.

Se for calculada a matriz de transição de estados (agora representando a saída apenas dentro dos possíveis valores da representação), obtém-se

$$\mathbb{P} = \begin{array}{c} \begin{matrix} -1.0 & -0.5 & 0 & 0.5 \end{matrix} \\ \left[\begin{array}{cccc} 0.333 & 0.333 & 0 & 0.333 \\ 0.333 & 0 & 0.333 & 0.333 \\ 0 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0 \end{array}\right] \end{array} \begin{array}{c} \textbf{Estados} \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \end{array} .$$

Usando $\mathbb{P}$, pode-se calcular a média e a variância dessa implementação, conforme apresentado na figura 47. Nessa figura, as curvas do modelo com cadeias de

Markov são calculadas por meio das equações (4.75) e (4.76) (pág. 102). As curvas obtidas são comparadas com as do modelo linearizado.

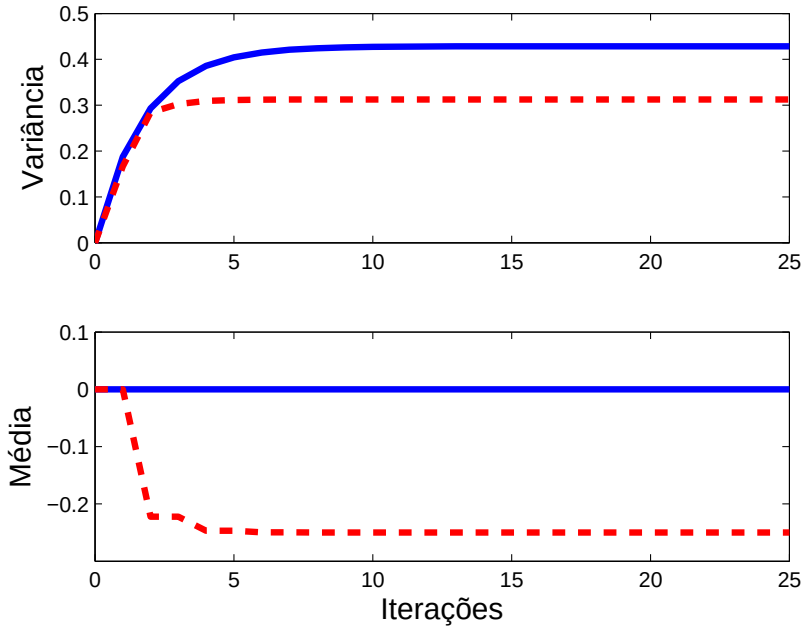

Figura 47: Variância e média via cadeias de Markov (linha pontilhada) e via modelo linearizado (linha contínua), considerando a exceção de *overflow*

Como pode ser observado, a abordagem linearizada não é exata em sua modelagem do filtro implementado, mostrando que a abordagem via cadeias de Markov pode ser uma forma mais precisa de prever o funcionamento do filtro.

6 EFEITOS DE PRECISÃO FINITA NO ALGORITMO LMS

Neste capítulo, cadeias de Markov são usadas para estudar o efeito da precisão finita no algoritmo LMS, quando existe correlação na entrada. Considera-se o algoritmo LMS unidimensional e calcula-se a matriz de transição de estados, em que os estados são determinados pelas possíveis representações do coeficiente do algoritmo e pela entrada correlacionada.

6.1 O algoritmo LMS e a precisão finita

Se $d(n)$ for uma sequência desejada e $\boldsymbol{x}(n)$ um vetor de dados, o problema de estimação em termos dos mínimos quadrados é obter a melhor estimativa linear de $d(n)$ em função de $\boldsymbol{x}(n)$, através da minimização da variância do erro. Isso equivale a resolver a função custo

$$J(\boldsymbol{w}(n)) = \mathrm{E}\{|e(n)|^2\} = \mathrm{E}\{|d(n) - \hat{d}(n)|^2\}, \tag{6.1}$$

em que

$$\hat{d}(n) = \boldsymbol{w}^T(n)\mathbf{x}(n) \tag{6.2}$$

é a estimativa, $\boldsymbol{x}(n)$ é o vetor regressor e $\boldsymbol{w}(n)$ é o vetor de pesos.

O algoritmo LMS (Least-Mean Squares) atualiza seus coeficientes $\boldsymbol{w}(n)$ com o intuito de minimizar (6.1), calculando

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu e(n)\boldsymbol{x}(n) \tag{6.3}$$

e

$$e(n) = d(n) - \hat{d}(n), \tag{6.4}$$

em que μ corresponde ao passo da adaptação. Com as equações (6.3) e (6.4), e considerando precisão infinita dos termos envolvidos, o algoritmo levará $\boldsymbol{w}(n)$ em direção ao ponto de ótimo da função custo, ou seja, $\boldsymbol{w}_{op}$, de modo que $\lim_{n\to\infty} \mathrm{E}\{\boldsymbol{w}(n)\} = \boldsymbol{w}_{op}$. Todavia, em aplicações práticas o algoritmo é implementado em *hardware* com comprimento de palavra finito, introduzindo não-linearidades que podem alterar seu funcionamento.

Os efeitos indesejados em precisão finita ocorrem devido à realimentação do erro de quantização. Uma situação indesejada pode acontecer, por exemplo, quando o gradiente se torna menor do que o bit menos significativo, podendo provocar o que é chamado de *stopping phenomenon* [24]. Nessa situação, a adaptação pode virtualmente parar. Esse efeito não é capturado com modelos linearizados e, por esse motivo, pode-se esperar que um modelo não-linear deva ser mais preciso do que a tradicional abordagem linearizada.

Em [3] e [4], o efeito da precisão finita no algoritmo LMS unidimensional foi estudado com o auxílio de cadeias de Markov. Nesses trabalhos, assumiu-se que $w(n)$ seria representado com B bits e em notação de ponto fixo, permitindo que $w(n)$ assumisse 2^B valores distintos. O conjunto desses 2^B valores, limitados entre -1 e $(1-\Delta)$ (com $\Delta = 2^{-B+1}$), foi usado para definir os estados da cadeia. A análise que se seguiu calculou a matriz de transição de estados, usada para prever o funcionamento do algoritmo para $n \to \infty$. As simulações demonstraram que essa era uma abordagem mais precisa para a análise de precisão finita do que a tradicional modelagem linearizada, em que um ruído uniforme é adicionada após cada multiplicação. Em particular, foi observado que o modelo com cadeias de Markov é mais preciso do que o linearizado quando a representação usa um número pequeno de bits, já que a influência das não-linearidades de quantização é

maior quando as palavras são menores. (Vale pensar na situação em que se usam 2 ou 4 bits para representar um mesmo sinal: no primeiro caso, o erro máximo que se comete ao quantizar uma grandeza é $2^{-2+1}/2 = 0.25$. No segundo caso, o erro de quantização máximo é de $2^{-4+1}/2 = 0.0625$.)

Neste capítulo, a abordagem de [3, 4] é retomada e estendida, introduzindo na análise a premissa de que a entrada atual do algoritmo ($x(n)$) pode apresentar correlação com seu valor anterior ($x(n-1)$). Nesse caso, cada estado passa a ser representado por um par ($w(n+1)$,$x(n)$), em que $w(n+1)$ e $x(n)$ podem assumir todos os diferentes valores da representação binária. Diferentemente de [3, 4], onde é usada saturação dos valores que excedem os limites da representação, a não-linearidade assumida nos cálculos é o *overflow* oriundo da representação em complemento-a-dois.

Na análise que se se segue, considera-se que a correlação ($r_x(k) = \mathrm{E}\{x(n)x(n-k)\}$) existe apenas entre a entrada atual e seu valor no instante anterior, isto é, $r_x(0)$ e $r_x(1) \neq 0$ e $r_x(\alpha) = 0$, $\forall \alpha \neq 0, 1$. Os cálculos são realizados para o algoritmo LMS mas podem ser estendidos para outros algoritmos, sendo necessário apenas algumas adaptações na análise e na forma de programação das simulações.

6.2 Modelo em precisão finita

Considerando $Q[\cdot]$ um operador de quantização após a multiplicação e $R\{\cdot\}$ o ajuste após a soma, é possível descrever as não-linearidades da implementação do algoritmo LMS unidimensional por meio de

$$\begin{aligned} d_Q(n) &= R\{w_{op}x_Q(n) + v(n)\} \\ \hat{d_Q}(n) &= Q[w(n)x_Q(n)] \\ e_Q(n) &= R\{d_Q(n) - \hat{d}_Q(n)\} \\ y_Q(n) &= Q[\mu e_Q(n)x_Q(n)], \end{aligned} \tag{6.5}$$

onde a atualização do peso do algoritmo é descrita por

$$w(n+1) = R\{w(n) + y_Q(n)\}, \tag{6.6}$$

como apresentado em [3, 4]. Nas equações (6.5) e (6.6), o subscrito Q indica a quantização da grandeza (por exemplo, $d_Q(n) = Q[d(n)]$).

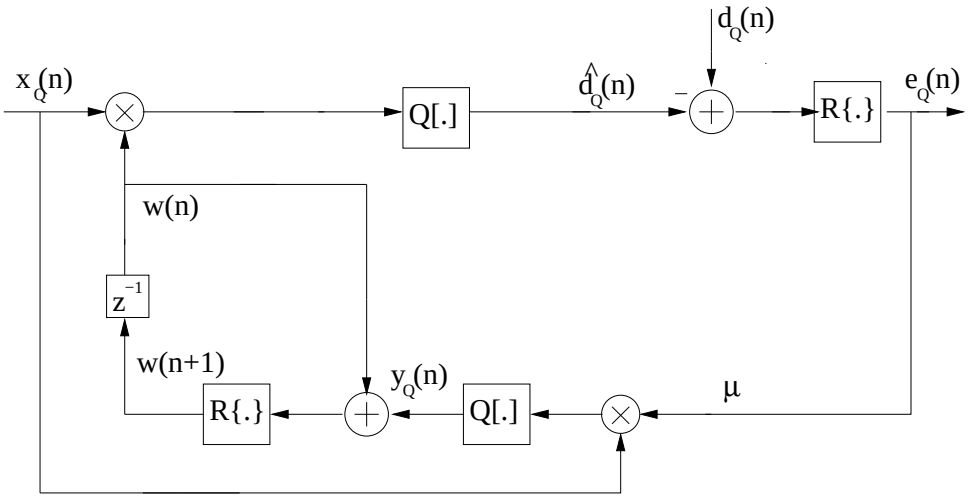

Figura 48: Algoritmo LMS unidimensional apresentando não-linearidades de quantização (representadas pelos blocos $Q[\cdot]$ e $R\{\cdot\}$)

6.2.1 Função de probabilidade correlacionada

Para o desenvolvimento da análise do algoritmo LMS em precisão finita, foi usada uma entrada $x(n)$ com distribuição gaussiana de média nula (vide pág. 24), correlacionada apenas com a entrada anterior $x(n-1)$. Usando $x(n) = x$ e $x(n-1) = x_1$ e definindo $\mathrm{E}\{xx\} = \sigma_x^2$ e $\mathrm{E}\{xx_1\} = \sigma$, a distribuição de probabilidade conjunta é calculada como

$$f_{x,x_1}(x, x_1) = \frac{1}{2\pi(\sigma_x^4 - \sigma^2)^{1/2}} e^{-\frac{(\sigma_x^2(x-x_1)-2\sigma x x_1)^2}{2(\sigma_x^4-\sigma^2)}}.$$

Para o cálculo das probabilidades desejadas, é necessário obter $f_x(x|x_1)$. Contudo, x_1 corresponde a uma versão passada de x, discretizada. Nesse caso, a

função de probabilidade condicionada pode ser calculada por

$$f_x(x|x_1) = \frac{f_{x,x_1}(x, x_1)}{f_{x_1}(x_1)}, \tag{6.7}$$

que deve ser calculada numericamente para fornecer as probabilidades dos valores discretos de x, assumindo x_1. Dessa forma,

$$P(x = i\Delta|x_1 = k\Delta) = \frac{\int_{l_I}^{l_s}\int_{L_I}^{L_s} f_{x,x_1}(x, x_1)dx\, dx_1}{\int_{L_I}^{L_s} f_{x_1}(x_1)dx_1}, \tag{6.8}$$

com l_I e l_S dados por

$$l_I = \begin{cases} i\Delta - \Delta/2, & \text{se} \quad i \geq -2^{B-1} + 1 \\ -\infty, & \text{se} \quad i = -2^{B-1} \end{cases}$$

e

$$l_s = \begin{cases} \infty, & \text{se} \quad i = 2^{B-1} - 1 \\ i\Delta + \Delta/2, & \text{se} \quad i \leq 2^{B-1} - 2 \end{cases},$$

e os extremos de integração L_I e L_s calculados como

$$L_I = \begin{cases} k\Delta - \Delta/2, & \text{se} \quad k \geq -2^{B-1} + 1 \\ -\infty, & \text{se} \quad k = -2^{B-1} \end{cases}$$

e

$$L_s = \begin{cases} \infty, & \text{se} \quad k = 2^{B-1} - 1 \\ k\Delta + \Delta/2, & \text{se} \quad k \leq 2^{B-1} - 2 \end{cases}.$$

Se for definido $\Theta_{ik} = P(x = i\Delta|x_1 = k\Delta)$, para i e k inteiros e contidos no conjunto $\{-2^{B-1}, ..., 2^{B-1} - 1\}$, a função de probabilidade condicionada pode ser finalmente reescrita em função dos valores discretos que x e x_1 podem assumir, originando

$$f_x(x|x_1 = k\Delta) = \sum_{i=-2^{B-1}}^{2^{B-1}-1} \Theta_{ik}\delta(x - i\Delta). \tag{6.9}$$

6.2.2 Cálculo da probabilidade de $d_Q(n)$

A partir de (6.9), obtém-se a função de probabilidade condicionada de $x_0 = w_{op}x(n)$, isto é,

$$f_{x_0}(x_0|x_1 = k\Delta) = \sum_{i=-2^{B-1}}^{2^{B-1}-1} \Theta_{ik}\delta(x_0 - i\Delta w_{op}). \tag{6.10}$$

A probabilidade associada a $d_Q(n) = R\{w_{op}x(n) + v(n)\}$ pode ser escrita como (considerando a ideia da pág. 88 e a equação (6.10))

$$P(d_Q(n) = l\Delta|x_1 = k\Delta) = \sum_{j=-\infty}^{\infty} P(d(n) = l\Delta + 2j|x_1 = k\Delta) =$$
$$\sum_{i=-2^{B-1}}^{2^{B-1}-1} \Theta_{ik} \sum_{j=-\infty}^{\infty} \frac{1}{2\sqrt{2}} \left[erf\left(\frac{d_{l_2} - i\Delta w_{op}}{\sqrt{2}\sigma_v}\right) - erf\left(\frac{d_{l_1} - i\Delta w_{op}}{\sqrt{2}\sigma_v}\right)\right],$$

com

$$\begin{cases} d_{l_2} = l\Delta + 2j + \Delta/2 \\ d_{l_1} = l\Delta + 2j - \Delta/2 \end{cases}. \tag{6.11}$$

Definindo

$$D_{lk} = P(d_Q(n) = l\Delta|x_1 = k\Delta), \tag{6.12}$$

obtém-se a função de probabilidade $f_{d_{Q_n}}(d_Q(n)|x_1 = k\Delta)$,

$$f_{d_{Q_n}}(d_Q(n)|x_1 = k\Delta) = \sum_{l=-2^{B-1}}^{2^{B-1}-1} D_{lk}\delta(d_Q(n) - l\Delta) \tag{6.13}$$

que corresponde à função de probabilidade desejada.

6.2.3 Cálculo da probabilidade de $e_Q(n)$

A probabilidade de $e_Q(n)$ pode ser escrita como uma função dependente de x, x_1 e $w(n)$, ou seja,

$$P(e_Q(n) = k_e\Delta|x, x_1, w(n)) = \sum_{j=-\infty}^{\infty} P(e(n) = k_e\Delta + 2j|x, x_1, w(n)), \tag{6.14}$$

em que k_e é um número inteiro entre -2^{B-1} e $2^{B-1}-1$. O lado direito da equação (6.14) pode ser alterado de maneira que apareça o termo $d_Q(n)$ (já que $e(n) = d_Q(n) - \hat{d}(n)$),

$$P(e_Q(n) = k_e\Delta|x, x_1, w(n)) = \sum_{j=-\infty}^{\infty} P(d_Q(n) = \hat{d}(n) + k_e\Delta + 2j|x, x_1, w(n)),$$

a partir de onde se define a função de probabilidade condicionada

$$f_{e_Q(n)}(e_Q(n)|x, x_1, w(n)) = \\ = \sum_{k_e=-2^{B-1}}^{2^{B-1}-1} P(e_Q(n) = k_e\Delta|x, x_1, w(n))\delta(e_Q(n) - k_e\Delta). \quad (6.15)$$

6.2.4 Cálculo da probabilidade de $y_Q(n)$

Semelhante a $e_Q(n)$, a probabilidade de $y_Q(n)$ corresponder a um valor discreto $k_q\Delta$, para k_q inteiro e entre -2^{B-1} e $2^{B-1}-1$, é calculada por

$$P(y_Q(n) = k_q\Delta|x, x_1, w(n)) = \sum_{j=-\infty}^{\infty} P(y(n) = k_q\Delta + 2j|x, x_1, w(n)). \quad (6.16)$$

Usando $y(n) = \mu e_Q(n)x_Q(n)$, que pode ser reescrita como $e_Q(n) = y(n)/(\mu x_Q(n))$ e substituindo em (6.16),

$$P(y_Q(n) = k_q\Delta|x, x_1, w(n)) = \sum_{j=-\infty}^{\infty} P\left(e_Q(n) = \frac{k_q\Delta + 2j}{\mu x}|x, x_1, w(n)\right). \quad (6.17)$$

A função de probabilidade condicionada pode ser descrita, então, por

$$f_{y_Q(n)}(y_Q(n)|x, x_1, w(n)) = \\ = \sum_{k_e=-2^{B-1}}^{2^{B-1}-1} P(y_Q(n) = k_q\Delta|x, x_1, w(n))\delta(y_Q(n) - k_q\Delta). \quad (6.18)$$

6.2.5 Cálculo da probabilidade de $w(n+1)$

O coeficiente $w(n+1)$ é calculado em termos de $w(n)$ e $y_Q(n)$. Se essa informação for usada, a probabilidade de $w(n+1) = k_w\Delta$, onde k_w é um inteiro entre -2^{B-1} e $2^{B-1}-1$, corresponde a

$$P(w(n+1) = k_w\Delta|x, x_1, w(n)) = \sum_{j=-\infty}^{\infty} P(y_q(n) = k_w\Delta - w(n) + 2j|x, x_1, w(n)).$$

A função de densidade de probabilidade condicionada fica definida, por

$$f_{w(n+1)}(w(n+1)|x, x_1, w(n)) = \\ = \sum_{k_w=-2^{B-1}}^{2^{B-1}-1} P(w(n+1) = k_w\Delta|x, x_1, w(n))\delta(w(n+1) - k_w\Delta),$$

de onde se estabelece uma sequência de passos para calcular a probabilidade do próximo estado $w(n+1)$ em termos da probabilidade da entrada de $x(n)$, dado $x(n-1)$, e do coeficiente anterior $w(n)$.

6.3 Teste do modelo com cadeias de Markov

Nesta seção são apresentadas simulações comparando o desempenho do modelo proposto com implementações do algoritmo LMS em precisão finita e com o modelo linearizado.

6.3.1 O *overflow* no LMS

A figura 49 apresenta em detalhe a ocorrência da não-linearidade de *overflow* para o LMS implementado com 3 bits, $w_{op} = 0.5$, com entrada correlacionada ($r(0) = 0.34$, $r(1) = 0.1095$ e $r(\alpha) = 0\ \forall|\alpha| \neq 0, 1.$) e SNR = 20 dB. O ponto A corresponde a um valor de $w(k) = 0.75$, enquanto os pontos B e C correspondem a $w(k+1) = -1$ e $w(k+2) = 0.5$. Da equação do LMS quantizado, sabe-se que

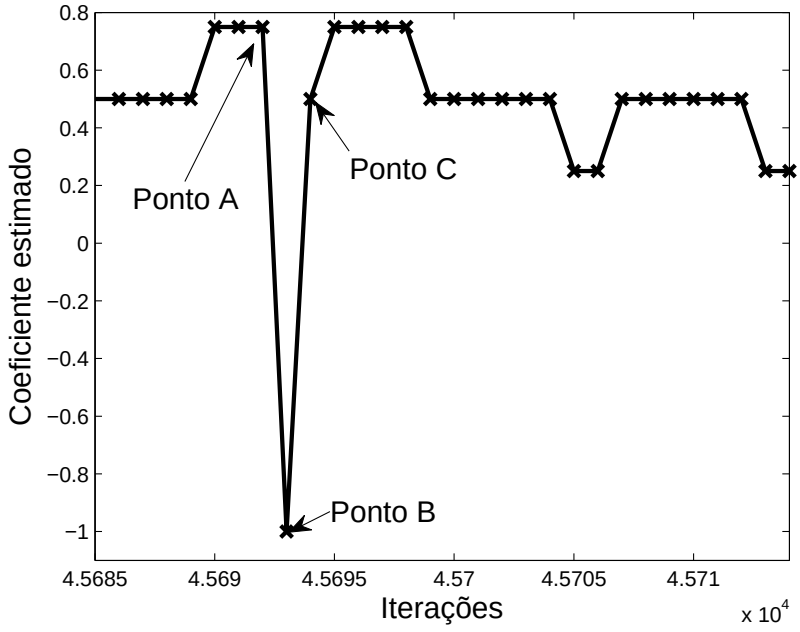

Figura 49: *Overflow* em implementação do LMS

o coeficiente estimado na iteração n corresponde à soma de dois termos,

$$w(n+1) = R\{w(n) + Q[\mu e_Q(n) x_Q(n)]\}, \tag{6.19}$$

ou seja, o valor de $w(n+1)$, antes de ser quantizado, está limitado por $-2 \leq w(n+1) \leq 2 - 2\Delta = 1.5$. Nesse caso, por exemplo, se $w(n) = 0.75$ e o outro termo for igual a 0.25, ocorrerá *overflow* e o resultado final será igual a -1 (vide figura 49, na transição do ponto A para o ponto B). De maneira semelhante, quando a soma excede o mínimo negativo -1, o resultado se torna positivo (vide a transição do ponto B para o ponto C, em que a soma de -1 e -0.5 é quantizada para $w(n) = 0.5$). Efeitos não-lineares como esse não são modelados de forma precisa por meio de linearizações e, por esse motivo, é interessante uma abordagem utilizando cadeias de Markov.

6.3.2 Comparação do erro quadrático médio (MSE)

Para comparar o resultado do modelo usando cadeias de Markov com uma implementação do algoritmo, implementou-se o algoritmo LMS unidimensional descrito pelas equações (6.5) e (6.6), assumindo a não-linearidade de *overflow*. As curvas obtidas foram comparadas com a curva teórica correspondente ao modelo linearizado (vide figura 50).

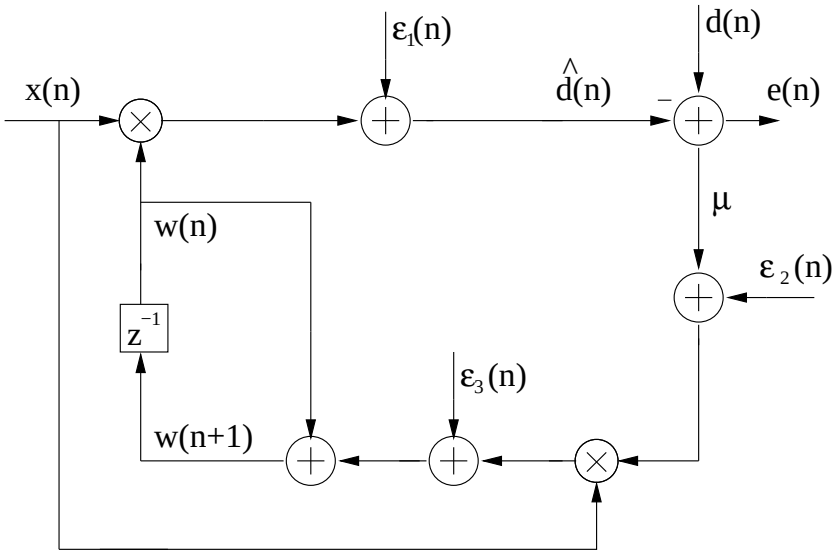

Figura 50: Algoritmo LMS unidimensional sob a ótica da abordagem linearizada ($\varepsilon_1(n)$, $\varepsilon_2(n)$ e $\varepsilon_3(n)$ correspondem aos ruídos de quantização)

Para obter a curva linearizada, ruídos de quantização de distribuição uniforme e descorrelacionados foram incluídos no algoritmo ($\varepsilon_1(n)$, $\varepsilon_2(n)$ e $\varepsilon_3(n)$, respectivamente). A variância de cada ruído foi calculada em função da quantidade de bits usada na representação de ponto fixo (B), como $\sigma^2_{\varepsilon_k} = \Delta^2/12$ (para $k = 1, 2, 3$). Com esses novos termos, as equações do algoritmo foram reescritas como

$$\begin{aligned} d_L(n) &= w_{op}x(n) + v(n) \\ \hat{d}_L(n) &= w_L(n)x(n) + \varepsilon_1(n) \\ e_L(n) &= d_L(n) - \hat{d}_L(n) \\ y_L(n) &= [\mu e_L(n) + \varepsilon_2(n)]x(n) + \varepsilon_3(n) \end{aligned} \tag{6.20}$$

e usadas em (6.3), fornecendo

$$w_L(n+1) = w_L(n) + \{\mu[w_{op}x(n) + v(n) - w_L(n)x_(n) - \varepsilon_1(n)] + \varepsilon_2(n)\}x(n) + \varepsilon_3(n). \tag{6.21}$$

Multiplicando (6.21) por -1 e somando w_{op} dos dois lados,

$$\begin{aligned} w_{op} - w_L(n+1) = \; & w_{op} - w_L(n) - \mu(w_{op} - w_L(n))x^2(n) - \mu v(n)x(n) \\ & + \{\mu x(n)\varepsilon_1(n) - x_(n)\varepsilon_2(n) - \varepsilon_3(n)\}. \end{aligned} \tag{6.22}$$

Fazendo

$$\tilde{w}(n) = w_{op} - w_L(n), \tag{6.23}$$

a equação (6.22) pode ser reorganizada como

$$\begin{aligned} \tilde{w}(n+1) = \; & (1 - \mu x^2(n))\tilde{w}(n) - \mu x(n)v(n) \\ & + \{\mu x(n)\varepsilon_1(n) - x(n)\varepsilon_2(n) - \varepsilon_3(n)\}. \end{aligned} \tag{6.24}$$

Elevando ao quadrado os dois lados de (6.24) e calculando a esperança,

$$\begin{aligned} \mathrm{E}\{\tilde{w}^2(n+1)\} = \; & \mathrm{E}\left\{[(1 - \mu x^2(n))\tilde{w}(n) - \mu x(n)v(n) \right. \\ & \left. + \{\mu x(n)\varepsilon_1(n) - x(n)\varepsilon_2(n) - \varepsilon_3(n)\}]^2\right\}, \end{aligned} \tag{6.25}$$

é possível encontrar uma forma recorrente para calcular $\mathrm{E}\{\tilde{w}^2(n+1)\}$.

De fato, $\mathrm{E}\{\tilde{w}^2(n+1)\}$ será composto por 2 termos,

$$\begin{aligned} A = \; & \mathrm{E}\left\{[(1 - \mu x^2(n))\tilde{w}(n) - \mu x(n)v(n)]^2\right\} \\ B = \; & \mathrm{E}\left\{[\mu x(n)\varepsilon_1(n) - x(n)\varepsilon_2(n) - \varepsilon_3(n)]^2\right\}, \end{aligned} \tag{6.26}$$

já que o termo cruzado será igual a zero, dado que $\varepsilon_1(n)$, $\varepsilon_2(n)$ e $\varepsilon_3(n)$ são independentes de todos os outros termos. O termo A pode ser calculado segundo o apresentado em [25], fornecendo

$$A = \mathrm{E}\{\tilde{w}^2(n)[1 - 2\mu\sigma_x^2 + 3\mu\sigma_{x]}^4] + \mu^2\sigma_v^2\sigma_x^2\}, \tag{6.27}$$

em que σ_x^2 e σ_v^2 correspondem às potências de $x(n)$ e de $v(n)$, respectivamente.

O termo B é calculado expandindo seus elementos, ou seja,

$$\begin{aligned} B = \; & \mathrm{E}\{\mu^2 x^2(n)\varepsilon_1^2(n) + x^2(n)\varepsilon_2^2(n) + \varepsilon_3^2(n) \\ & -2\mu x^2(n)\varepsilon_1(n)\varepsilon_2(n) - 2\mu x(n)\varepsilon_1(n)\varepsilon_3(n) + 2x(n)\varepsilon_2(n)\varepsilon_3(n)\}. \end{aligned} \tag{6.28}$$

Analisando os termos de (6.28), apenas os 3 primeiros serão diferentes de zero. Com isso, pode-se fazer

$$B = \mu^2\sigma_{\varepsilon_1}^2\sigma_x^2 + \sigma_{\varepsilon_2}^2\sigma_x^2 + \sigma_{\varepsilon_3}^2, \tag{6.29}$$

o que leva a

$$\mathrm{E}\{\tilde{w}^2(n+1)\} = \mathrm{E}\{\tilde{w}^2(n)\}[1-2\mu\sigma_x^2+3\mu\sigma_x^4]+\mu^2\sigma_v^2\sigma_x^2+\mu^2\sigma_{\varepsilon_1}^2\sigma_x^2+\sigma_{\varepsilon_2}^2\sigma_x^2+\sigma_{\varepsilon_3}^2, \tag{6.30}$$

que fornece a equação teórica para o cálculo de $\mathrm{E}\{\tilde{w}^2(n+1)\}$.

O erro quadrático médio ($\mathrm{E}\{e^2(n)\}$) pode ser calculado por

$$\begin{aligned} \mathrm{E}\{e^2(n)\} &= \mathrm{E}\{(d_L(n) - \hat{d}_L(n)^2\} \\ &= \mathrm{E}\{([w_{op} - w_L(n)]x(n) - \varepsilon_1(n) + v(n))^2\} \\ &= \mathrm{E}\{\tilde{w}^2(n)\}\sigma_x^2 + \sigma_v^2 + \sigma_{\varepsilon_1}^2, \end{aligned} \tag{6.31}$$

lembrando que, por hipótese, os ruídos de quantização e $v(n)$ são independentes dos demais termos e substituindo em (6.31) $\mathrm{E}\{w_L^2(n)x^2(n)\}$ por $\mathrm{E}\{w_L^2(n)\}\sigma_x^2(n)$ (o que é usualmente usado na análise do LMS [26]). Portanto, usando uma condição inicial para $\mathrm{E}\{w_L^2(n)\}$, pode-se calcular analiticamente o MSE do modelo linearizado com as equações (6.30) e (6.31).

Para obter o valor do erro quadrático médio utilizando cadeias de Markov, é necessário usar

$$E\{e^2(n)\} = \sum_{i=-2^{-B+1}}^{2^{-B+1}-1} (i\Delta)^2 \sum_{j=-2^{-B+1}}^{2^{-B+1}-1} \sum_{k=-2^{-B+1}}^{2^{-B+1}-1} P(i\Delta|w(n) = j\Delta, x(n-1) = k\Delta)\times$$
$$\sum_{l=-2^{-B+1}}^{2^{-B+1}-1} \sum_{m=-2^{-B+1}}^{2^{-B+1}-1} P(w(n) = j\Delta, x(n-1) = k\Delta|w(0) = l\Delta, x(-1) = m\Delta)\times$$
$$P(w(0) = l\Delta, x(-1) = m\Delta), \quad (6.32)$$

onde $x(-1)$ e $w(0)$ são as condições iniciais e o erro é escrito em termos dos valores discretos que pode assumir ($e(n) = i\Delta$, para i inteiro e entre -2^{-B+1} e $2^{-B+1} - 1$). Os termos $P(w(n) = j\Delta, x(n-1) = k\Delta|w(0) = l\Delta, x(-1) = m\Delta)$ correspondem aos elementos da matriz de transição de estados $\mathbb{P}$ em sua n-ésima potência.

6.3.3 Simulações

Para demonstrar a validade do modelo proposto, calculou-se o MSE da abordagem via cadeias de Markov e do modelo linearizado. As curvas obtidas foram comparadas com o MSE de uma implementação afetada por quantização.

Tal como desenvolvido nas seções anteriores, assume-se que o sinal de entrada $x(n)$ possui distribuição gaussiana de média nula, e que $x(n)$ é correlacionado com $x(n-1)$, com $r(0) = 0.34$, $r(1) = 0.1095$ e $r(\alpha) = 0\ \forall|\alpha| \neq 0, 1$. A relação sinal-ruído (SNR) considerada é de $20dB$. O passo de adaptação do algoritmo é escolhido igual a 1, de maneira que não ocorra a parada da adaptação (*stopping phenomenon*), segundo apresentado em [4]. O coeficiente ótimo foi definido como $w_{op} = 0.5$. Para definir um estado inicial para a cadeia de Markov, o valor inicial do peso é feito $w(0) = -0.5$ e $x(-1) = -1$. Os estados são representados com $B = 2$ bits, assumindo a não-linearidade de *overflow*. A figura 51 apresenta o MSE calculado de 3 maneiras diferentes e compara o modelo de Markov, linearizado e a implementação do filtro. As equações (6.30) e (6.31) foram usadas na obtenção

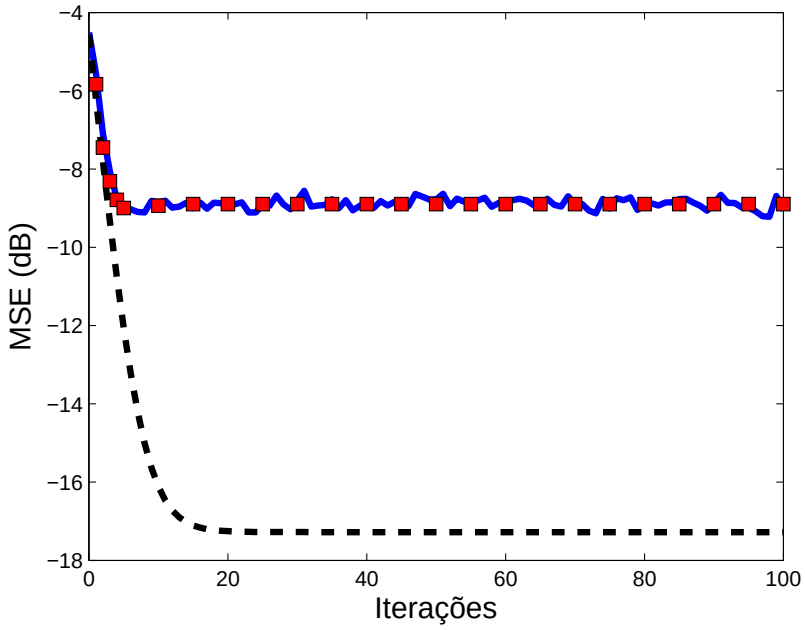

Figura 51: MSE calculado para o algoritmo LMS: implementação real (curva contínua), curva teórica do modelo linearizado (pontilhada) e calculado com cadeias de Markov (quadrados)

da curva da linearização, com a condição inicial

$$\mathrm{E}\{\tilde{w}^2(0)\} = \mathrm{E}\{[w_{op} - w(0)]^2\} = \mathrm{E}\{1\} = 1. \tag{6.33}$$

As curvas do filtro simulado (linha contínua) e do modelo via cadeias de Markov (representada por quadrados) apresentam grande semelhança, enquanto que o modelo linearizado (curva pontilhada) não é capaz de modelar o funcionamento do filtro de forma precisa. Vale lembrar que no cálculo da curva teórica do modelo linearizado são realizadas diversas simplificações, as quais são aplicadas durante a análise teórica do próprio LMS (por exemplo, simplificar $\mathrm{E}\{w(n)^2x^2(n)\}$ por $\mathrm{E}\{w(n)^2\}\sigma_x^2(n)$). Isso mostra que o modelo via cadeias de Markov pode ser uma alternativa mais precisa ao modelo linearizado, já que considera de forma exata os efeitos não-lineares.

6.3.4 Comparação da estimativa do coeficiente

A estimativa de $w(n)$ através de $\mathrm{E}\{w(n)\}$ também pode ser comparada ao modelo teórico e à implementação do algoritmo. Retomando a equação (6.21) e calculando a média,

$$\mathrm{E}\{w(n+1)\} = (1 - \mu\sigma_x^2)\,\mathrm{E}\{w(n)\} + \mu\sigma_x^2 w_{op}, \tag{6.34}$$

obtém-se uma forma recursiva para calcular $\mathrm{E}\{w(n)\}$ do modelo teórico linearizado. Para o modelo de cadeias de Markov, a média pode ser calculada por

$$\begin{aligned} E\{w(n)\} = \sum_{i=-2^{B-1}}^{2^{B-1}-1} i\Delta \sum_{k=-2^{B-1}}^{2^{B-1}-1} \sum_{l=-2^{B-1}}^{2^{B-1}-1} \sum_{m=-2^{B-1}}^{2^{B-1}-1} \\ P(w(n) = i\Delta, x(n-1) = k\Delta | w(0) = l\Delta, x(-1) = m\Delta) \times \\ P(w(0) = l\Delta, x(-1) = m\Delta) \end{aligned} \tag{6.35}$$

Usando as condições apresentadas na seção 6.3.3, obtém-se a figura 52, confirmando a melhor precisão do modelo de cadeias de Markov para avaliar efeitos de precisão finita.

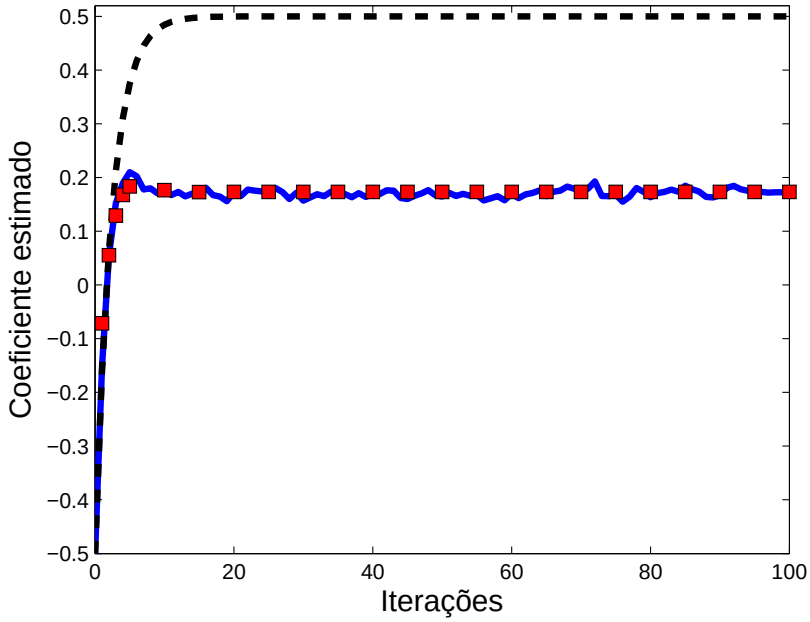

Figura 52: Estimativa do coeficiente: implementação real (curva contínua), modelo teórico linearizado (curva pontilhada) e usando cadeias de Markov (quadrados)

7 CONCLUSÕES

Ao longo deste texto, foi proposto usar cadeias de Markov para elaborar uma forma mais precisa de estudar o efeito de aritmética de precisão finita na implementação de filtros digitais fixos e adaptativos.

Para filtros digitais fixos de primeira e de segunda ordem, foi desenvolvida uma maneira de calcular as probabilidades condicionadas da saída $y(n)$ em função de saídas anteriores, através de manipulações nas funções de probabilidades da entrada $x(n)$. Os resultados obtidos permitiram calcular a matriz de transição de estados, por meio da qual algumas análises puderam ser realizadas. Neste texto, uma implementação na forma direta I foi usada como referência. Foram considerados filtros implementados em formas extremamente econômicas, em que o acumulador tem comprimento de palavra reduzido, tornando necessária a quantização após cada multiplicação, e filtros com acumulador de tamanho suficiente para que a quantização ocorresse apenas ao final das operações. Contudo, a metodologia usada pode ser estendida para outras implementações, bastando apenas modificar a forma de calcular as funções de probabilidade envolvidas.

Na análise de filtros fixos implementados com a não-linearidade de saturação, foi proposta uma nova forma de escrever $\mathbb{P}$, através da separação dos efeitos de saturação em dois novos estados, por onde foi possível observar o impacto da saturação em $y(n)$. Nesse ponto, foi proposta uma forma iterativa de calcular um fator de escalamento para $x(n)$, pela qual se obteve um resultado menos

conservador, aumentando a relação sinal-ruído na saída. Também foi possível encontrar uma maneira mais precisa de calcular a média e a variância da saída, usando $\mathbb{P}$ ao invés da abordagem linearizada de modelar a quantização como um erro somado após cada multiplicação.

A análise considerando a não-linearidade de *overflow* forneceu resultados semelhantes, embora não tenha sido usada para escalamento da entrada de filtros.

O estudo de ciclos-limites de entrada zero apresentou resultados interessantes. Foi possível identificar implementações livres dessas oscilações, através da busca por autovalores unitários na matriz de transição de estados. Nesse caso, foi mostrado que apenas os filtros cuja matriz de transição de estados possui apenas um autovalor igual a 1 e os demais com módulo menor que 1, estão livres de ciclos-limite de entrada nula.

Ainda no contexto de filtros fixos, o modelo via cadeias de Markov foi aproveitado para estudar filtros colocados em cascata. Para isso, filtros de primeira ordem foram colocados em cascata e foram encontradas as cadeias de Markov associadas a cada filtro da cascata, o que foi usado para comparações com filtros implementados com o modelo de Markov de segunda ordem. O objetivo era verificar se a implementação em cascata forneceria resultados semelhantes aos do modelo de segunda ordem, com um custo computacional menor, devido à redução do número de estados envolvidos. Em filtros passa-tudo, em que vale o fato de que a saída se mantém descorrelacionada se a entrada é descorrelacionada, o modelo de cascata de filtros apresentou resultados semelhantes aos da abordagem linear ou melhores, mais próximos do modelo de segunda ordem. Em filtros passa-baixa, o modelo mostrou-se ruim para a determinação da f.d.p. da saída. Isso aconteceu porque o modelo de cadeia de Markov foi calculado assumindo entrada descorrelacionada, o que não é válido para a entrada do segundo filtro da cascata. Contudo, essas observações são referentes a poucos exemplos, o

que torna necessário a realização de mais testes para determinar a validade desse modelo.

No contexto de filtros adaptativos implementados em ponto fixo, foi realizada uma análise no algoritmo LMS unidimensional, introduzindo o conceito de entrada correlacionada. Essa abordagem levou à definição dos estados da cadeia de Markov em função da entrada e do coeficiente estimado. As comparações com o modelo linearizado e com simulações de filtros implementados mostrou que cadeias de Markov fornecem um modelo mais preciso, tornando atrativa sua aplicação.

Este trabalho apresentou algumas contribuições para o estudo de filtros digitais, por meio da aplicação da teoria de cadeias de Markov. Para torná-las mais claras, pode-se enumerá-las de forma resumida nos seguintes itens:

1. Extensão do modelo de cadeias de Markov para filtros IIR.
2. Consideração da exceção de *overflow* no modelo desenvolvido.
3. Extensão do modelo desenvolvido para filtros IIR para qualquer função de densidade de probabilidade de entrada.
4. Publicação de um artigo sobre o estudo de filtros IIR por meio do modelo de cadeia de Markov.
5. Desenvolvimento da análise com entrada correlacionada para o LMS unidimensional.

Como resultado dessas contribuições, foi apresentado um artigo em um congresso internacional. Intitulado *Applying Markov chains to calculate the probability of saturation in digital IIR filters* (vide Anexo A), este trabalho foi apresentado no *International Telecommunication Symposium* (ITS 2010), realizado em Manaus. O artigo expõe os resultados obtidos com a matriz de transição

de estados expandida com os estados saturados, segundo apresentado na seção 4.4.1. Nele também são apresentados os resultados obtidos no cálculo da média e da variância da saída do filtro, devido à não-linearidade de saturação, como mostrado na seção 4.4.2.

Este trabalho não esgota todas as possibilidades de estudo através de cadeias de Markov. Ao contrário, existem diversos tópicos de interesse que podem ser estudados em pesquisas futuras, como

1. Extensão da análise com cadeias de Markov para filtros implementados em ponto flutuante. Nesse caso, os estados seriam definidos em termos dos valores da mantissa e do expoente, permitindo análises semelhantes às aqui apresentadas.

2. Introdução de entrada correlacionada nos modelos desenvolvidos neste trabalho para filtros fixos. Essa abordagem geraria matrizes de transição de estados com um maior número de elementos, dado que haveria mais estados, relacionando a saída e a entrada dos filtros.

3. Estudo da matriz de transição de estados obtida a partir de análise de filtros implementados em paralelo.

4. Identificação de padrões nas matrizes de transição de estados, de forma que os cálculos e os dados que necessitam ser armazenados sejam reduzidos.

Dessa maneira, ainda existem muitas possibilidades para explorar as propriedades da matriz de transição de estados no estudo de filtros digitais, fixos ou adaptativos. O assunto possui vários tópicos que ainda podem ser estudados e diversos resultados interessantes podem ser obtidos em outras pesquisas futuras.

REFERÊNCIAS

[1] DINIZ, P. S. R.; SILVA, E. A. B. da; NETTO, S. L. *Processamento Digital de Sinais: Projeto e Análise de Sistemas*. São Paulo: Bookman, 2004.

[2] ANTONIOU, A. *Digital signal processing: signals, systems, and filters*. [S.l.]: McGraw-Hill, New York, 2006.

[3] MONTENEGRO-MALUENDA, Y. *Propriedades do Algoritmo LMS em Precisão Finita*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2005.

[4] MONTENEGRO MALUENDA, Y.; BERMUDEZ, J. C. M.; NASCIMENTO, V. H. Modeling finite precision LMS behavior using Markov chains. In: *Proc., ICASSP 2006*. Toulouse, France: [s.n.]. III, p. 97–100.

[5] PAPOULIS, A. *Probability & statistics*. [S.l.]: Prentice-Hall, 1990.

[6] BERTSEKAS, D.; TSITSIKLIS, J. *Introduction to probability*. [S.l.]: Athena Scientific Belmont, Massachusetts, 2002.

[7] BRÉMAUD, P. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. [S.l.]: Springer Verlag, 1999.

[8] NORRIS, J. R. *Markov Chains*. [S.l.]: Cambridge University Press, 1998.

[9] MEYER, C. D. *Matrix Analysis and Applied Linear Algebra*. Philadelphia, USA: SIAM, 2000.

[10] OPPENHEIM, A.; SCHAFER, R.; BUCK, J. *Discrete-time signal processing*. [S.l.]: Prentice Hall Englewood Cliffs, NJ, 1989.

[11] CARLETTA, J. et al. Determining appropriate precisions for signals in fixed-point IIR filters. ACM, 2003.

[12] AVENHAUS, E. On the design of digital filters with coefficients of limited word length. *IEEE Transactions on Audio and Electroacoustics*, v. 20, n. 3, p. 206–212, 1972.

[13] CROCHIERE, R. A new statistical approach to the coefficient word length problem for digital filters. *IEEE Transactions on Circuits and Systems*, v. 22, n. 3, p. 190–196, 1975.

[14] DEBRUNNER, L.; DEBRUNNER, V.; PINAULT, P. Variable wordlength IIR filter implementations for reduced space designs. In: *2000 IEEE Workshop on Signal Processing Systems, 2000. SiPS 2000*. [S.l.: s.n.], 2000. p. 326–335.

[15] BAICHER, G. Optimization of Finite Word Length Coefficient IIR Digital Filters Through Genetic Algorithms–A Comparative Study. *Advances in Natural Computation*, Springer, p. 641–650, 2006.

[16] ARSLAN, T.; HORROCKS, D. A genetic algorithm for the design of finite word length arbitraryresponse cascaded IIR digital filters. In: *Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995. GALESIA. First International Conference on (Conf. Publ. No. 414)*. [S.l.: s.n.], 1995. p. 276–281.

[17] HAN, K.; EVANS, B. Optimum wordlength search using sensitivity information. *EURASIP Journal on Applied Signal Processing*, Hindawi Publishing Corp., v. 2006, p. 76, 2006.

[18] SZCZUPAK, J.; GREEN, C. On the elimination of zero-input limit cycles digital filters. In: *IEEE International Symposium on Circuits and Systems, 1990.* [S.l.: s.n.], 1990. p. 129–132.

[19] BOSE, T.; BROWN, D. Limit cycles in zero input digital filters due to two's complement quantization. *IEEE Transactions on Circuits and Systems*, v. 37, n. 4, p. 578–571, 1990.

[20] CAMPO, J.; CRUZ-ROLDAN, F.; UTRILLA-MANSO, M. Tighter limit cycle bounds for digital filters. *IEEE Signal Processing Letters*, v. 13, n. 3, p. 149–152, 2006.

[21] PREMARATNE, K. et al. An exhaustive search algorithm for checking limit cycle behavior of digital filters. *IEEE Transactions on Signal Processing*, v. 44, n. 10, p. 2405–2412, 1996.

[22] BAUER, P.; LECLERC, L. A computer-aided test for the absence of limit cycles infixed-point digital filters. *IEEE Transactions on Signal Processing*, v. 39, n. 11, p. 2400–2410, 1991.

[23] YAN-ZHONG, Z.; YA-FEN, Y. Detection and removal of LCOs in IIR filters. In: *1992 IEEE International Symposium on Circuits and Systems, 1992. ISCAS'92. Proceedings.* [S.l.: s.n.], 1992. v. 1.

[24] BERMUDEZ, J.; BERSHAD, N. Nonlinear quantization effects an the LMS algorithm-analytical models for the MSE transient and convergence behavior. In: IEEE. *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on.* [S.l.], 2002. v. 1, p. 627–631.

[25] NASCIMENTO, V. H. *Stability and Performance of Adaptive Filters without Slow Adaptation Approximations.* Tese (Doutorado) — University of California, Los Angeles, 1999.

[26] HAYKIN, S. *Adaptive Filter Theory.* 4th. ed. [S.l.]: Prentice Hall, 2002.

ANEXO A – ARTIGO PARA O ITS2010

Applying Markov chains to calculate the probability of saturation in digital IIR filters

Fernando G. Almeida Neto and Vítor H. Nascimento
Electronic Systems Engineering Department
Escola Politécnica, University of São Paulo
São Paulo, Brazil
E-mails: {fganeto, vitor}@lps.usp.br

José Carlos M. Bermudez
Electrical Engineering Department
Federal University of Santa Catarina
Florianópolis, Brazil
E-mail: bermudez@eel.ufsc.br

***Abstract*— We propose a new method to model the effect of finite-precision arithmetic in infinite impulse response (IIR) digital filters. As an application, we use the proposed model to compute the probability of saturation or overflow in IIR filters implemented in fixed-point arithmetic. The transition from the current filter output to the next output is modeled as a first-order Markov chain. The Markov chain transition probability matrix is then used to evaluate the probabilities of saturation or overflow for first and second-order IIR filters.**

***Keywords*— Saturation, IIR filters, Markov chains, finite precision arithmetic.**

I. Introduction

Modeling the behavior of algorithms when implemented in finite-precision arithmetic is important for practical designs. Such models are however difficult to develop due to the highly nonlinear characteristic of the quantization operation. Infinite impulse response (IIR) filters may be considerably sensitive to finite-precision effects, given their feedback structure. This is specially true when the number of bits is small. In this case, the usual modeling of quantization noise as a uniformly-distributed random variable is not appropriate. Nevertheless, designs using short wordlengths are required in applications where low power consumption is paramount, such as cellular phones and other portable devices. In these cases, a more precise model for the quantization effect is desirable.

During its operation, the output of a filter can also exceed its allowed range, severely degrading the filter performance. In finite-precision arithmetic, such an exception may be dealt with simply by disregarding the most significant bits of the output (which we will call "overflow") or by saturating the output to its most positive or most negative value. In both cases, a large error results, which should be avoided for proper system operation. One way to avoid saturation is to scale the filter coefficients to avoid, or at least reduce, the probability of exceeding the output range [1], [2]. The approach is to determine the transfer function from the input of the filter to the input of each multiplier and then use the inverse of the p-norm of this function as a scaling factor, where p is chosen according to the signal in the input of the filter. However, this is a worst-case approach, which may lead to conservative designs (i.e., a scale factor that is smaller than necessary, leading to a lower signal-to-noise ratio). A model that incorporates the highly nonlinear overflow and saturation effects during the filter operation can be of great help for the designer. To the best of our knowledge, no precise models for predicting their probability of occurrence are available.

This work is partly supported by CNPq under Grants 136050/2008-5, 303.361/2004-2 and 305377/2009-4, and by FAPESP under Grants 2008/00773-1, 2008/04828-5 and 2009/03609-0

In this paper we propose to model the behavior of first and second-order IIR filters using a first-order Markov chain. This approach is an extension of [3] and [4], which investigated the impact of finite precision in the performance of the least mean squares (LMS) algorithm. We consider a fixed-point implementation and use no linearization in the description of the signal quantizations. We apply a Markov chain to model the transition probabilities from the current output to the possible future outputs of the filter. We take advantage of the fact that the output may assume only a finite number of values in finite-precision. These values are interpreted as states of a Markov chain. We introduce extra states in the transition matrix to calculate the probability of saturation or overflow.

This paper is organized as follows: Section II presents the nonlinear IIR filters models used here, while Section III makes a brief introduction to the Markovian concepts needed. Section IV introduces the approach to calculate the probability of saturation, while Section V shows some examples of the proposed method. Section VI concludes the paper.

II. Nonlinear effects in IIR filters

Digital IIR filters are, in general, implemented as a cascade of first and second-order filters, which are described by the difference equations (1) and (2), respectively,

$$y_1(n) = b_0 u(n) + b_1 u(n-1) - a(1) y_1(n-1) \tag{1}$$

and

$$\begin{aligned} y_2(n) = {} & b_0 u(n) + b_1 u(n-1) + b_2 u(n-2)) \\ & - a_1 y_2(n-1) - a_2 y_2(n-2), \end{aligned} \tag{2}$$

where the filter coefficients are given by a_k, for $k = 1, 2$, and b_k, for $k = 0, 1, 2$. Equations (1) and (2) consider coefficients and signals represented in finite precision [5]. In this paper, we consider fixed-point representation.

A fixed-point implementation uses a fixed number of bits to represent the integer and the fractional parts of a number. A filter has thus a finite number of codes to describe quantities, which is given by $N = 2^B$, where B is the word length in bits. If the range of representable number is from -1 to $+1-\Delta$, the quantization step will be $\Delta = 2^{-B+1}$, and the result of all operations must be rounded or truncated to fit to the numerical representation. If a sum or a multiplication result exceeds the representation, another nonlinear operation is used to find a representation within the established bounds. For instance, saturation limits the exceeding quantities to the bounds of the representation, while overflow disregards the most significant bits outside the allowed range, resulting in large errors. Figure 1 shows saturation and overflow for a two's complement 2-bit signal representing the set $\{-1, -0.5\ 0\ 0.5\}$.

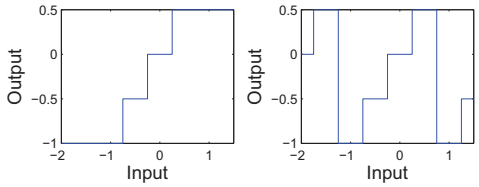

Fig. 1. Saturation (left) and overflow (right) for a 2-bit signal

Equations (3) and (4) describe the application of the nonlinear operations to the filter equations, i.e.,

$$y_1(n) = R[R[Q\{b_0u(n)\} + Q\{b_1u(n-1)\}] + Q\{-a_1y_1(n-1)\}] \tag{3}$$

and

$$y_2(n) = R[R[Q\{b_0u(n)\} + R[Q\{b_1u(n-1)\} + Q\{b_2u(n-2)\}]] + R[Q\{-a_1y_2(n-1)\} + Q\{-a_2y_2(n-2)\}]], \tag{4}$$

where $R[\cdot]$ can be the saturation or the overflow operator after a sum and $Q\{\cdot\}$ represents quantization after a multiplication. Here we consider the most economical implementation, where accumulators are not available to perform multiply-accumulation operations. Figures 2 and 3 show the quantized filters in a direct form I implementation. Note that the $R[\cdot]$ operator may be applied only once in (3) and (4) if the processor has a register with guard bits for intermediate computation, such as is found in most DSP. The method presented here may be easily modified to consider all details of a specific implementation.

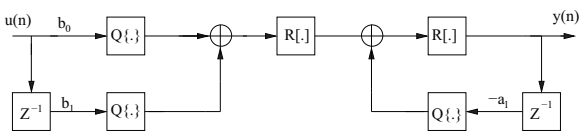

Fig. 2. Quantized first-order IIR filter implemented in direct-form I

The possible outputs of IIR filters implemented in fixed-point are contained in a finite set, and the current output clearly depends on the last outputs and on the current and last inputs, as we can observe in (3) and (4). We can take advantage of these two characteristics and use a first-order discrete Markov chain [6] to find a probabilistic description of the filter output as a function of the past output and the input. In this case, we can define each possible output of the filter as a state of a Markov chain. In the next section, we introduce some concepts of Markov chains used in this paper. To clarify the calculation of the transition matrix, we also present an example with a first-order filter.

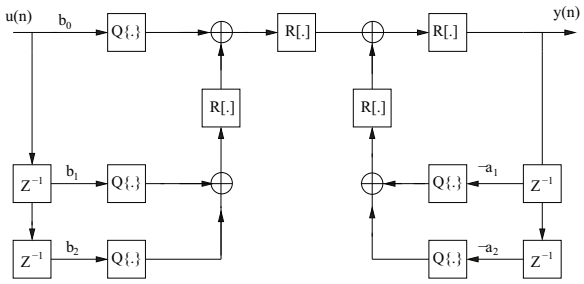

Fig. 3. Quantized second-order IIR filter implemented in direct-form I

III. Discrete-time Markov chains

A discrete-first-order Markov chain [6] is a discrete stochastic process where the probability of the next state, given the current and the past states, only depends on the current state. That means that given a *stochastic process* $\{X_n\}_{n=0}^{\infty}$,

$$P(X_n = i_n | X_{n-1} = i_{n-1}, \cdots, X_0 = i_0) = P(X_n = i_n | X_{n-1} = i_{n-1}),$$

where $P(a|b)$ is the conditional probability of *a given b* and i_n represents the possible states for X_n. The subscript n represents time instants ($n = 1, 2, \ldots$) and we consider that the states i_n belong to the finite set $\{1, 2, \ldots, N\}$. In general, the notation used in (5) is abbreviated as $P(X_n = i | X_{n-1} = j) = p_{ij}$, where p_{ij} is defined as the probability to reach state i when the current state is j. This notation is used to define an $N \times N$ matrix $\mathbb{P}$ with elements p_{ij}. $\mathbb{P}$ is called the *transition matrix* and its main characteristic, by construction, is that all the columns add to 1, since

$$\sum_{i=1}^{N} p_{ij} = \sum_{i=1}^{N} P(X_n = i | X_{n-1} = j) = 1. \tag{5}$$

Indeed, each column represents a conditional probability distribution for each state, in a specific instant. Therefore, if we consider that the transition probabilities are independent of n, we can find the transition probabilities after n instants, which corresponds to

$$p_{ij}^{(n)} = P(X_n = i | X_0 = j), \tag{6}$$

where $p_{ij}^{(n)}$ is the probability to begin in the state j and reach the state i after n steps. We can use the Chapman-Kolmogorov equation [7] to calculate $p_{ij}^{(n)}$ as

$$p_{ij}^{(n)} = \sum_{k=0}^{N} p_{kj}^{(n-1)} p_{ik}. \tag{7}$$

Equation (7) shows an iterative method to calculate $p_{ij}^{(n)}$ given the past probabilities. Writing in matrix form [6],

$$\mathbb{P}^{(n)} = \mathbb{P}^{(n-1)} \cdot \mathbb{P}^{(1)} = \mathbb{P}^{(1)} \cdot \mathbb{P}^{(n-1)} = \mathbb{P}^{n}, \text{ n} = 1, 2, \ldots \tag{8}$$

where $\mathbb{P}(0) = I_{N\times N}$. We conclude that $\mathbb{P}^{(n)}$ is equivalent to $\mathbb{P}^n$. Thus, given a initial probability distribution vector π_0 for X_0, we obtain

$$\pi_n = \mathbb{P}^n \pi_0, \tag{9}$$

and we notice that the knowledge of $\mathbb{P}$ and π_0 allows us to know the distribution after n steps.

If we look to $\mathbb{P}^n$ when $n \to \infty$, we obtain the process steady-state (SS) matrix. This matrix differs from the initial matrix $\mathbb{P}$ because of the absence of transient states, which are the states that stop receiving visits after a finite number of steps. The SS matrix contains the information about the long term process, and therefore about the saturation of the output in steady-state.

Computing $\mathbb{P}$ for IIR filters is simple, as we show in the next example.

Example 1: Suppose a 2-bit filter, described by the coefficients $a_1 = 0.5$, $b_0 = 0.5$ and $b_1 = 0$. We want to calculate the 4×4 matrix $\mathbb{P}$ when there is an input $u(n)$ with uniform distribution and zero mean, (i.e., the probability of $u(n) = -0.5$, 0 and 0.5 are $1/3$ and the probability of $u(n) = -1$ is zero, and the input is white). We consider here that $R[\cdot]$ is the saturation operator and that we round up, i.e., $Q\{0.25\} = 0.5$ and $Q\{-0.25\} = 0$. Let us, for example, find the element $p_{34} = P(y(n) = 0|y(n-1) = 0.5)$. For this element, equation (3) is modified as

$$y(n) = R[Q\{0.5u(n)\} + Q\{-0.5\cdot 0.5\}],$$

since $b_1 = 0$ and $y(n-1) = 0.5$. Varying $u(n)$ for all the possible values, if we calculate $y(n)$, we notice that

$$\begin{aligned} y(n) &= R[Q\{0.5(-1)\} + Q\{-0.5(0.5)\}] = -0.5 \\ y(n) &= R[Q\{0.5(-0.5)\} + Q\{-0.5(0.5)\}] = 0 \\ y(n) &= R[Q\{0.5(0)\} + Q\{-0.5(0.5)\}] = 0 \\ y(n) &= R[Q\{0.5(0.5)\} + Q\{-0.5(0.5)\}] = 0.5 \end{aligned},$$

where the last result comes from the saturation of 1 to 0.5. Therefore, there are two possibilities to reach $y(n) = 0$ (when $u(n) = -0.5$ and $u(n) = 0$), and we should use the distribution of the input to calculate p_{34}, as

$$p_{13} = P(u(n) = -0.5) + P(u(n) = 0) = \frac{1}{3} + \frac{1}{3} = 0.667.$$

Using the same procedure to determine the other elements of $\mathbb{P}$ yields

$$\begin{array}{c} \begin{array}{cccc} -1.0 & -0.5 & 0 & 0.5 \end{array} \quad \textbf{States} \\ \mathbb{P} = \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.667 & 0.667 \\ 1.000 & 1.000 & 0.333 & 0.333 \end{array}\right] \begin{array}{c} -1.0 \\ -0.5 \\ 0 \\ 0.5 \end{array} \end{array}$$

where the numbers above and to the right of $\mathbb{P}$ show the values of $y(n)$ and $y(n-1)$ corresponding to each row and column.

Assume now that the current output has a distribution $\pi_n = [0\ 0.5\ 0\ 0.5]$, and we want to know the probability of $y(n+1) = 0.5$. Using (9), we obtain

$$\pi_{n+1} = \mathbb{P}\pi_n = [0\ \ 0\ \ 0.333\ \ 0.667]^T,$$

which is the distribution vector at instant $n+1$. Thus, we conclude that $P(y(n+1) = 0.5) = 0.333$. We can also find the distribution for the instant $n+2$, calculating

$$\pi_{n+2} = \mathbb{P}\pi_{n+1} = \mathbb{P}^2\pi_n = [0\ \ 0\ \ 0.667\ \ 0.333]^T.$$

Therefore, we can calculate any probability for any time instant if we have $\mathbb{P}$ and an initial distribution vector for $y(0)$. In the next section, we include extra states to describe overflow or saturation. For convenience, we refer to these extra states as "saturation states".

IV. Modelling saturation

In section III, $\mathbb{P}$ was described with states related to the output of the filter, considering that the output is limited to a range (e.g., the range of the past example is the set $\{-1, -0.5, 0, 0.5\}$), and using a nonlinear saturation to guarantee this limitation. This means that when the output exceeds the range, this value is limited to the bounds of the range (e.g., for the past example if the filter calculates an output of 1, the saturation limits the output to 0.5). In this case, although $\mathbb{P}$ takes saturation into account, we cannot distinguish saturated from nonsaturated outputs. However, we can introduce more states to model the saturation and obtain the exact probability of saturation in a filter. For this purpose, we add two states in the transition matrix, corresponding to the saturation to the positive and negative limits of the output. The matrix obtained will be $(N+2)\times(N+2)$ if we are using a first-order filter and $(N+2)^2\times(N+2)^2$ when a second-order filter is analyzed. (Although the matrices are large, we can apply sparse matrix computation to reduce the calculations, since the matrices have a large number of zero elements [1].)

Consider again, the first example. To observe the saturation, we define two extra states: -1_s and 0.5_s. The state -1_s corresponds to an output -1, but that is reached through saturation. In the same way, the state 0.5_s corresponds to an output 0.5 obtained by saturation. Let us find $P(y(n) = e|y(n-1) = -1)$, when $e\in\{-1_s\ -1\ -0.5\ 0\ 0.5\ 0.5_s\}$, we notice that

$$\begin{aligned} y(n) &= Q\{0.5(-1)\} + Q\{-0.5(-1)\} = 0 \\ y(n) &= Q\{0.5(-0.5)\} + Q\{-0.5(-1)\} = 0.5 \\ y(n) &= Q\{0.5(0)\} + Q\{-0.5(-1)\} = 0.5 \\ y(n) &= Q\{0.5(0.5)\} + Q\{-0.5(-1)\} = 1 = 0.5_s. \end{aligned}$$

Therefore, we conclude that the element of the transition matrix

$$p_{42} = P(x(n) = -0.5) + P(x(n) = 0) + P(x(n) = 0.5) = 1$$

includes one part related to the output saturation (when $x(n) = 0.5$). To include the two saturation states, we use an expanded matrix $\mathbb{P}$, where the column $P(y(n)|y(n-1) = -1)$, for $y(n) \in \{-1_s\ -1\ -0.5\ 0\ 0.5\ 0.5_s\}$, corresponds to

$$P(y(n)|y(n-1) = -1) = [0\ \ 0\ \ 0\ \ 0\ \ 0.667\ \ 0.333]^T.$$

[1] In fact, the use of sparse matrices is more efficient when we calculate the transition matrix for a second-order filter, since the matrix dimension is larger and zero elements appear more frequently.

If we calculate the full expanded matrix $\mathbb{P}$, we obtain

$$\mathbb{P}=\begin{array}{c} \begin{matrix} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.667 & 0.667 & 0.667 \\ 0.667 & 0.667 & 0.667 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \textbf{States} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array}$$

The rows corresponding to -1_s and 0.5_s show the conditional probabilities of saturation. We notice from the transition matrix that the columns related to -1_s and -1 have the same distribution. This happens because when we have state -1_s or -1, the output is -1. Therefore, $P(y(n)|y(n-1)=-1_s)=P(y(n)|y(n-1)=-1)$, and the columns must be equal. The same argument is valid for the states 0.5_s and 0.5. (It is important to note that the columns corresponding to -0.5 and 0, in general, do not need to be equal to the columns of the states -1 and 0.5, as we observe in this example.)

V. Examples

In order to calculate the probability of saturation with the proposed method, we wrote two programs in *Matlab*. The programs calculate the transition matrix based on the probability distribution of the inputs, and they describe first and second order IIR filters, as presented in equations (3) and (4). For simplicity, we present only one example, assuming a 2-bit first order filter We use saturation after sums. The program inputs are the filter coefficients, the input word length B and the distribution of $u(n)$. We use sparse matrix calculation to reduce the number of operations.

A. Probability of saturation

Consider the filter $a_1=0.75$, $b_0=1$ and $b_1=0$, where the coefficients have a 3-bit description, while we still have a 2-bit input and output (this choice is made only to keep the example simple). Using the input distribution $[0\ 1/3\ 1/3\ 1/3]$, the transition matrix is given by

$$\mathbb{P}=\begin{array}{c} \begin{matrix} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.333 & 0.333 \\ 0 & 0 & 0 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0.333 & 0.333 & 0.333 \\ 0.333 & 0.333 & 0.333 & 0.333 & 0 & 0 \\ 0.333 & 0.333 & 0.333 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \textbf{States} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array},$$

and the SS matrix is

$$\mathbb{P}^{\infty}=\begin{array}{c} \begin{matrix} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.111 & 0.111 & 0.111 & 0.111 & 0.111 & 0.111 \\ 0.222 & 0.222 & 0.222 & 0.222 & 0.222 & 0.222 \\ 0.333 & 0.333 & 0.333 & 0.333 & 0.333 & 0.333 \\ 0.222 & 0.222 & 0.222 & 0.222 & 0.222 & 0.222 \\ 0.111 & 0.111 & 0.111 & 0.111 & 0.111 & 0.111 \end{bmatrix} \end{array} \begin{array}{c} \textbf{States} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array}.$$

We conclude from the SS matrix that there are outputs which exceed the superior limit of the representation, with a probability of 0.111, no matter what is the initial condition. We can scale the coefficient b_0 to avoid saturation — in this simple example, this means guaranteeing that $|y(n)|\leq 0.5$. If we use the traditional approach of the p-norm (which we indicate by $\|\cdot\|_p$), as presented in [1], [2], we must calculate the transfer function from the input to the output of the filter, i.e.,

$$H(z)=\frac{1}{1+0.75z^{-1}},$$

to calculate the scaling factor as

$$\lambda \leq \frac{0.5}{\|h(n)\|_p \|u(n)\|_q}, \text{ for } \frac{1}{p}+\frac{1}{q}=1, \tag{10}$$

where $h(n)$ is the filter's impulse response. This approach is based on Hölder's inequality [8],

$$|y(n)|=\sum_{k=0}^{\infty}|h(k)u(n-k)|\leq \|h(n)\|_p\|u(n)\|_q, \text{ for } \frac{1}{p}+\frac{1}{q}=1.$$

In this example, as $u(n)$ has unlimited energy, we should use $q=\infty$ and $p=1$. If we calculate the 1-norm for $h(n)$ and the infinity norm for $x(n)$, we obtain

$$\|h(n)\|_1=1+\sum_{n=1}^{\infty}|0.75^n|=4$$

and

$$\|x(n)\|_{\infty}=\max|x(n)|=0.5.$$

Using these in (10), we find that $\lambda\leq 0.25$. However, if we iteratively apply our method, we find $\lambda_{\text{opt}}=0.375$ (for coefficients with 3 bits). The new transition matrix is given by

$$\mathbb{P}=\begin{array}{c} \begin{matrix} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.000 & 1.000 \\ 0 & 0 & 0 & 1.000 & 0 & 0 \\ 1.000 & 1.000 & 1.000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \textbf{States} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array}.$$

while the SS matrix is

$$\mathbb{P}^{\infty}=\begin{array}{c} \begin{matrix} -1.0_s & -1.0 & -0.5 & 0 & 0.5 & 0.5_s \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1.000 & 1.000 & 1.000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.000 & 1.000 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \textbf{States} \\ -1.0_s \\ -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 0.5_s \end{array}.$$

We conclude that,using the transition matrix, one may iteratively search for the largest scaling factor that avoids saturation or overflow, avoiding conservative designs based on worst-case considerations.

B. Quantization noise

We now use the model proposed here to compute the mean and variance of the filter output, and compare with predictions based on the linearized approach, in which quantization errors are modelled as noise with uniform distribution. In the example presented, there is one error related to quantization, after the multiplication by a_1 (since $b_0 = 1$, there is no quantization error). We can model this signal $e(n)$ with a uniform distribution, zero mean and variance equal to [2]

$$\sigma_e^2 = \frac{\Delta^2}{12}.$$

Assume that the initial condition is $y(n-1) = 0$ with probability 1 and that the input is an independent sequence, with distribution as before.

$$\begin{aligned} \mathrm{E}\{y(n)^2\} = a_1^2\,\mathrm{E}\{y(n-1)^2\} + b_0^2\,\mathrm{E}\{u(n)^2\} + \\ 2a_1 b_0\,\mathrm{E}\{y(n-1)x(n)\} + \mathrm{E}\{e(n)^2\}. \end{aligned} \tag{11}$$

From the independence of the input sequence, it follows that $e(n)$ is independent of $y(n-1)$, so $2a_1 b_0 E\{y(n-1)u(n)\} = 0$. We calculated the mean and the variance of the output for the filter in the last example. Since $u(n)$ and $e(n)$ have zero mean, the output mean is also zero. The variance was calculated with (11) and is presented in figure 4.

We calculated the exact mean $\mu(n)$ of the output with our approach, for the same initial condition, using the extended transition matrix, i.e.,

$$\begin{aligned} &\pi(1) = \mathbb{P}[0\,0\,0\,1\,0\,0\,0]^T \\ &\mu(1) = \pi^T(1)\,[-1\ -1\ -0.5\ 0\ 0.5\ 0.5]^T \\ &\pi(2) = \mathbb{P}^2[0\,0\,0\,1\,0\,0\,0]^T \\ &\mu(2) = \pi^T(2)\,[-1\ -1\ -0.5\ 0\ 0.5\ 0.5]^T \\ &\vdots \\ &\pi(n) = \mathbb{P}^n[0\,0\,0\,1\,0\,0\,0]^T \\ &\mu(n) = \pi^T(n)\,[-1\ -1\ -0.5\ 0\ 0.5\ 0.5]^T \end{aligned}, \tag{12}$$

where $\mu(k)$ is the mean for iteration k. Similarly, for the variance, we calculated

$$\sigma_y^2(n) = \pi^T(n) \begin{bmatrix} (-1-\mu(n))^2 \\ (-1-\mu(n))^2 \\ (-0.5-\mu(n))^2 \\ (0-\mu(n))^2 \\ (0.5-\mu(n))^2 \\ (0.5-\mu(n))^2 \end{bmatrix} \tag{13}$$

for k from 0 to 25. Figure 5 shows the results.

We note from figures 4 and 5 that for this example, the linearized approach to analyze the quantization effects in digital filters produces significantly different results than the more precise approach proposed in this paper. This difference is expected to be large for filters with short wordlengths, and to diminish as the wordlength increases.

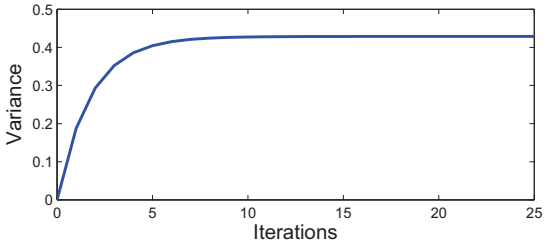

Fig. 4. Output variance for the linearized approach

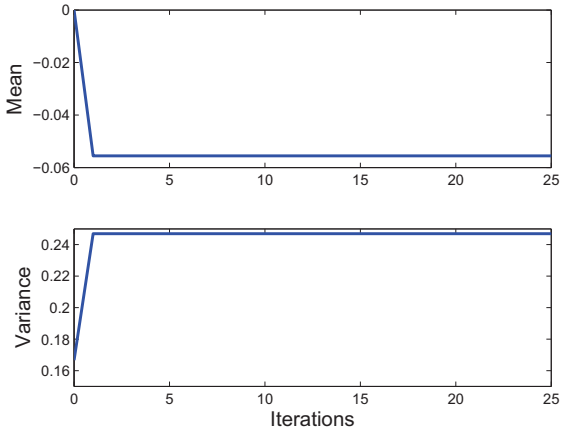

Fig. 5. Mean and variance of the output

VI. Conclusions

In this paper, we used Markov chains to describe the behavior of first and second-order IIR filters. The model allows a more precise prediction of the effect of quantization errors in digital filters implemented with short wordlengths. We calculated the transition matrix for the Markov chain with the addition of two states to represent saturation of the output. The saturation states allow one to find the best scaling factor to avoid saturation. The use of the new model was exemplified with a simple first-order filter.

References

[1] A. Antoniou, *Digital signal processing: signals, systems, and filters*. McGraw-Hill, New York, 2006.

[2] P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Processamento Digital de Sinais: Projeto e Análise de Sistemas*. São Paulo: Bookman, 2004.

[3] Y. Montenegro Maluenda, J. C. M. Bermudez, and V. H. Nascimento, "Modeling finite precision LMS behavior using Markov chains," in *Proc., ICASSP 2006*, vol. III, Toulouse, France, pp. 97–100.

[4] Y. R. Montenegro Maluenda, J. C. M. Bermudez, and V. H. Nascimento, "Propriedades do algoritmo LMS operando em precisão finita," in *Anais do XXII Simpósio Brasileiro de Telecomunicações*, Campinas, SP, 2005, pp. 1–7.

[5] A. Antoniou, *Digital Filters: Analysis, Design, and Applications*, 2nd ed. McGraw-Hill, 1993.

[6] J. R. Norris, *Markov Chains*. Cambridge University Press, 1998.

[7] D. Bertsekas and J. Tsitsiklis, *Introduction to probability*. Athena Scientific Belmont, Massachusetts, 2002.

[8] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, USA: SIAM, 2000.