



Dissertação de Mestrado

**PROJETO DE QUANTIZADORES NÃO LINEARES PARA
CONVERSORES ANALÓGICO-DIGITAIS**

Laryssa Lorrany Olinda Costa

Brasília, Dezembro de 2020

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**DESIGN OF NONLINEAR QUANTIZERS FOR
ANALOG-TO-DIGITAL CONVERTERS**

LARYSSA LORRANY OLINDA COSTA

ORIENTADOR: SANDRO AUGUSTO PAVLIK HADDAD, PROF. DR.

**DISSERTAÇÃO DE MESTRADO EM
ENGENHARIA DE SISTEMAS ELETRÔNICOS E
DE AUTOMAÇÃO**

PUBLICAÇÃO: PPGEA.TD-756/20

BRASÍLIA/DF: DEZEMBRO - 2020

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**PROJETO DE QUANTIZADORES NÃO LINEARES PARA
CONVERSORES ANALÓGICO-DIGITAIS**

LARYSSA LORRANY OLINDA COSTA

DISSERTAÇÃO DE Mestrado submetida ao Departamento de Engenharia Elétrica da Faculdade de Tecnologia da Universidade de Brasília como parte dos requisitos necessários para a obtenção do grau de Mestre.

APROVADA POR:

**Prof. Dr. Sandro Augusto Pavlik Haddad – FGA/Universidade de Brasília
Orientador**

**Prof. Dr. José Camargo da Costa – ENE/Universidade de Brasília
Membro Interno**

**Prof. Dr. Wellington Avelino do Amaral – FGA/Universidade de Brasília
Membro Externo**

BRASÍLIA, 11 DE DEZEMBRO DE 2020.

FICHA CATALOGRÁFICA

COSTA, LARYSSA LORRANY

Projeto de Quantizadores Não Lineares para Conversores Analógico-Digitais [Distrito Federal] 2020.

xvi, 103p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia de Sistemas Eletrônicos e de Automação, 2020).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Quantização não linear

2. Compandor

3. Transformada da Incerteza

4. Conversores analógico-digitais

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

COSTA, L.L. (2020). Projeto de Quantizadores Não Lineares para Conversores Analógico-Digitais . Dissertação de Mestrado em Engenharia de Sistemas Eletrônicos e de Automação, Publicação PPGEA.TD-756/20, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 103p.

CESSÃO DE DIREITOS

AUTOR: Laryssa Lorrany Olinda Costa

TÍTULO: Projeto de Quantizadores Não Lineares para Conversores Analógico-Digitais .

GRAU: Mestre ANO: 2020

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Laryssa Lorrany Olinda Costa

Departamento de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Dedico este trabalho a mim.

AGRADECIMENTOS

Agradeço a minha família, responsável por criar a pessoa que sou hoje. Agradeço ao Prof. Dr. Sandro Augusto Pavilik Haddad pelo incentivo, orientação e por me despertar novas ideias e possibilidades. Agradeço também ao Prof. Dr. José Edil Guimarães de Medeiros pela oportunidade de participar do seu trabalho. Por fim agradeço a meu amigo José Alberto, pela ajuda e conhecimento compartilhado.

RESUMO

Título: Projeto de Quantizadores Não Lineares para Conversores Analógico-Digitais

Autor: Laryssa Lorrany Olinda Costa

Orientador: Sandro Augusto Pavlik Haddad, Prof. Dr.

**Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos e de Automação
Brasília, 11 de dezembro de 2020**

No processo de conversão analógico-digital, a quantização linear é a mais utilizada. Essa técnica de quantização pode, no entanto, gerar grandes erros de quantização. Esses erros podem ser grandes a ponto da relação sinal-ruído não ser suficiente para a recuperação de toda a informação contida no sinal. Como alternativa para esse problema, pode-se utilizar uma técnica de quantização não linear. Este trabalho apresenta a modelagem de quantizadores não lineares para aplicação em conversores analógico-digitais com arquitetura Σ - Δ *multi-bit*. Os quantizadores projetados utilizam como ferramenta a Transformada da Incerteza e a técnica de *Compadding* Logarítmico e foram desenvolvidos com o auxílio das ferramentas Cadence e Matlab, seguindo a metodologia de projeto *Top-Down*. Na modelagem dos sistemas utilizou-se a linguagem de descrição de *hardware* Verilog-A, que possibilita a análise comportamental e simulação mista. A primeira parte deste trabalho apresenta a modelagem dos quantizadores não lineares e também de um quantizador linear para ser utilizado como referência. Para validar o funcionamento dos quantizadores realizou-se a análise no tempo e na frequência e comparou-se os resultados obtidos. A segunda parte apresenta a aplicação dos quantizadores projetados em moduladores Σ - Δ de 1ª e de 2ª ordem com a finalidade de validar o funcionamento dos mesmos. Nessa parte também realizou-se a análise no tempo e na frequência e extraiu-se os parâmetros dinâmicos de cada topologia. Os resultados das simulações mostraram que os quantizadores não lineares apresentaram desempenho superior ao quantizador linear nos parâmetros relacionados a distorção, ruído de quantização e resolução.

Palavras-chave: Quantização não linear, Compadding, Transformada da Incerteza, Conversores analógico-digitais.

ABSTRACT

Title: Design of Nonlinear Quantizers for Analog-to-digital Converters

Author: Laryssa Lorrany Olinda Costa

Supervisor: Sandro Augusto Pavlik Haddad, Prof. Dr.

Graduate Program in Electronic and Automation Systems Engineering

Brasília, December 11th, 2020

In the analog-to-digital conversion process, linear quantization is the most used. This quantization technique can, however, generate large quantization errors. These errors can be so large that the signal-to-noise ratio will not going to be sufficient to recover all the information contained in the signal. As an alternative to this problem, a non-linear quantization technique can be used. This work presents the modeling of non-linear quantizers for application in analog-digital converters with Σ - Δ multi-bit architecture. The designed quantizers use the Unscented Transform and the Logarithmic Companding technique and were developed with the aid of the Cadence and Matlab tools, following the Top-Down design methodology. The hardware description language, Verilog-A, was used in the modeling, which enables behavioral analysis and mixed simulation. The first part of this work presents the modeling of non-linear quantizers and also of a linear quantizer to be used as a reference. To validate the operation of the quantizers, the analysis was performed in time and frequency domains and the results obtained were compared. The second part presents the application of quantizers designed in Σ - Δ modulators of 1st and 2nd order in order to validate their operation. In this part, time and frequency analysis was also performed and the dynamic parameters of each topology were extracted. The results of the simulations showed that the non-linear quantizers performed better than the linear quantizer in parameters related to distortion, quantization noise and resolution.

Keywords: Nonlinear quantization, Companding, Unscented Transform, Analog-to-digital converters.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	QUANTIZAÇÃO NÃO LINEAR	1
1.2	OBJETIVOS E ESCOPO	2
1.3	METODOLOGIA DE PROJETO.....	3
1.4	ORGANIZAÇÃO DA DISSERTAÇÃO.....	4
2	REVISÃO BIBLIOGRÁFICA	5
2.1	LINGUAGENS DE DESCRIÇÃO DE <i>Hardware</i>	5
2.1.1	VERILOG	5
2.1.1.1	VERILOG-A.....	6
2.2	CONVERSORES ANALÓGICO-DIGITAIS.....	6
2.3	CARACTERÍSTICAS ESTÁTICAS EM CONVERSORES <i>A/D</i>	8
2.3.1	ERRO DE <i>Offset</i>	8
2.3.2	ERRO DE GANHO.....	8
2.3.3	NÃO LINEARIDADE DIFERENCIAL	9
2.3.4	NÃO LINEARIDADE INTEGRAL	10
2.4	CARACTERÍSTICAS DINÂMICAS EM CONVERSORES <i>A/D</i>	10
2.4.1	RELAÇÃO SINAL-RUÍDO (<i>SNR</i>)	10
2.4.2	DISTORÇÃO HARMÔNICA TOTAL (<i>THD</i>)	11
2.4.3	FAIXA DINÂMICA (<i>DR</i>)	11
2.4.4	RELAÇÃO SINAL-RUÍDO E DISTORÇÃO (<i>SNDR</i>).....	11
2.4.5	NÚMERO EFETIVO DE BITS (<i>ENOB</i>)	12
2.4.6	FAIXA DINÂMICA LIVRE DE ESPÚRIOS (<i>SFDR</i>)	12
2.5	TOPOLOGIAS DE CONVERSORES <i>A/D</i>	13
2.5.1	<i>Flash</i>	14
2.5.2	SIGMA-DELTA	15
2.5.2.1	FUNDAMENTOS DA MODULAÇÃO SIGMA-DELTA	16
2.5.2.2	TÉCNICA DE QUANTIZAÇÃO MULTI-BIT	19
3	TÉCNICAS DE QUANTIZAÇÃO NÃO LINEAR.....	21
3.1	A TRANSFORMADA DA INCERTEZA.....	21
3.1.1	QUANTIZADOR NÃO LINEAR UTILIZANDO A <i>UT</i>	22
3.1.1.1	PROPOSTA DE IMPLEMENTAÇÃO DO CIRCUITO.....	23
3.2	COMPANDING LOGARÍTMICO.....	25
3.2.1	μ -LAW	27
3.2.2	A-LAW	28

4 PROJETO, IMPLEMENTAÇÃO E RESULTADOS.....	30
4.1 QUANTIZADOR LINEAR	30
4.1.1 COMPARADOR	31
4.1.2 ENCODER	32
4.1.3 <i>ADC Flash</i>	34
4.1.4 DECODER	35
4.1.5 DIAGRAMA DE BLOCOS DO QUANTIZADOR LINEAR.....	38
4.2 QUANTIZADOR ARCO-SENO	39
4.3 QUANTIZADORES NÃO LINEARES UTILIZANDO A LEI μ	41
4.4 QUANTIZADORES NÃO LINEARES UTILIZANDO A LEI A	45
4.5 COMPARAÇÃO ENTRE QUANTIZADORES	49
4.6 MODULADOR Σ - Δ <i>Multi-bit</i>	55
4.6.1 INTEGRADOR	55
4.6.2 SUBTRATOR.....	57
4.7 MODULADOR Σ - Δ LINEAR	58
4.8 MODULADOR Σ - Δ ARCO-SENO	62
4.9 MODULADOR Σ - Δ μ -C	64
4.10 MODULADOR Σ - Δ A-C	66
4.11 COMPARAÇÃO ENTRE MODULADORES Σ - Δ <i>Multi-bit</i>	68
5 CONCLUSÃO.....	80
5.1 TRABALHOS FUTUROS	82
REFERÊNCIAS.....	82
A APÊNDICES.....	87
A.1 CÓDIGOS EM VERILOG A	87
A.1.1 COMPARADOR	87
A.1.2 DECODER	87
A.1.3 ENCODER	95
A.1.4 CHAVE	99
A.2 MODULADOR Σ - Δ <i>Multi-bit</i>	100
A.2.1 INTEGRADOR	100
A.2.2 SUBTRATOR.....	101
A.2.3 AMPLIFICADOR.....	101
B ANEXOS.....	102
B.1 QUANTIZADORES.....	102
B.2 MODULADORES Σ - Δ <i>Multi-bit</i>	103

LISTA DE FIGURAS

1.1	Comparação entre os processos de quantização uniforme e não uniforme para dois sinais distintos, um sinal forte e um sinal fraco (adaptada de [1]).	2
2.1	Relação entre Verilog-AMS, Verilog-A e Verilog-HDL.	6
2.2	Diagrama de blocos de um <i>ADC</i> genérico adaptada de [2]	7
2.3	Processo de conversão analógica-digital (Adaptada de [3]).	7
2.4	Erro de offset de a) $+1.5LSB$ b) $-2LSB$ em um <i>ADC</i> (Adaptada de [4]).	8
2.5	Erro de ganho de a) para $+1.5 LSB$, b) $-1.5 LSB$ em um <i>ADC</i> (Adaptada de [4]).	9
2.6	Erros de não linearidade a) <i>DNL</i> e b) <i>INL</i> para a curva de um <i>ADC</i> (Adaptada de [4]).	9
2.7	Exemplo de gráfico da <i>SNR</i> e <i>SNDR</i> e <i>DR</i> vs Potência do sinal entrada para um modulador Σ - Δ (Adaptada de [5]).	12
2.8	<i>SNR</i> e <i>SFDR</i> de um sinal aleatório [4].	13
2.9	Regiões de atuação das principais topologias de <i>ADCs</i> [6].	14
2.10	<i>ADC flash</i> de 3 bits [2].	15
2.11	Diagrama de blocos de um <i>ADC</i> Sigma-Delta (Adaptada de [7]).	16
2.12	Diagrama de blocos de um <i>ADC</i> Σ - Δ de 1ª ordem (Adaptada de [8]).	17
2.13	Modelo de um modulador Σ - Δ no domínio S.	17
2.14	Entrada e saída de um modulador Σ - Δ <i>single-bit</i> de 1ª ordem [8].	18
2.15	Modelo de um modulador Σ - Δ de 2ª ordem [2].	18
2.16	Diagrama de blocos de um modulador multi-bit de 1ª ordem.	19
2.17	Circuito de um modulador multi-bit de 1ª ordem [7].	20
2.18	Saída do modulador multi-bit no domínio do tempo a) e no domínio da frequência b) [7].	20
3.1	Princípio da UT (Adaptada de [9]).	21
3.2	Curva característica de um quantizador de 4 níveis com distribuição arco-seno e com um sinal de entrada senoidal [10].	23
3.3	Diagrama de blocos para implementação do quantizador arco-seno (Adaptada de [11]).	24
3.4	Proposta de implementação utilizando topologia <i>flash</i> : a) <i>ADC</i> ; b) <i>DAC</i> [11].	25
3.5	Diagrama de blocos de um quantizador não uniforme utilizando a técnica de <i>companding</i> .	26
3.6	Gráfico da função $F(x) = 1 + k^{-1} \ln(x)$ [12].	26

3.7	Curvas para o compressor utilizando a lei u [12].	27
3.8	Compressor baseado na lei A [12].	29
4.1	Testbench do comparador [13].	31
4.2	Simulação do comparador [13].	32
4.3	<i>Testbench</i> do encoder [13].	33
4.4	Simulação do encoder para o código de entrada “000000111111111”[13].	34
4.5	Parte do esquemático do <i>ADC flash</i> de 4 bits [13].	35
4.6	<i>Testbench</i> do decoder [13].	36
4.7	Simulação do decoder para o código de entrada “1010”.	37
4.8	Diagrama de blocos do quantizador linear.	38
4.9	Simulação do quantizador linear para um sinal senoidal [13].	38
4.10	Simulação do quantizador para um sinal do tipo rampa [13].	39
4.11	Simulação do quantizador Arco-seno para um sinal senoidal[13].	40
4.12	Simulação do quantizador Arco-seno para um sinal do tipo rampa [13].	41
4.13	Diagrama de blocos do quantizador não linear utilizando μ -law.	41
4.14	Simulação do quantizador μ para um sinal de entrada senoidal.	42
4.15	Simulação do quantizador μ para um sinal dente de serra.	42
4.16	Simulação do quantizador μ invertido para um sinal de entrada senoidal.	43
4.17	Simulação do quantizador μ invertido para um sinal dente de serra.	43
4.18	Diagrama de blocos do quantizador μ -C.	44
4.19	Simulação do quantizador μ -C para um sinal de entrada senoidal.	45
4.20	Simulação do quantizador μ -C para um sinal dente de serra.	45
4.21	Diagrama de blocos do quantizador não linear utilizando A -law.	46
4.22	Simulação do quantizador A para um sinal senoidal.	46
4.23	Simulação do quantizador A para um sinal dente de serra.	47
4.24	Simulação do quantizador A invertido para um sinal senoidal.	47
4.25	Simulação do quantizador A invertido para um sinal dente de serra.	48
4.26	Simulação do quantizador A -C para um sinal de entrada senoidal.	48
4.27	Simulação do quantizador A -C para um sinal dente de serra.	49
4.28	Análise transiente para um sinal senoidal processado pelo quantizador linear e seu respectivo erro de quantização.	50
4.29	Análise transiente para um sinal senoidal processado pelo quantizador arco-seno e seu respectivo erro de quantização.	50
4.30	Análise transiente para um sinal senoidal processado pelo quantizador μ e seu respectivo erro de quantização.	51
4.31	Análise transiente para um sinal senoidal processado pelo quantizador A e seu respectivo erro de quantização.	52

4.32	Diferença entre sinais processados pelo quantizador μ (verde) e pelo quantizador A (vermelho).	52
4.33	Análise transiente para um sinal senoidal processado pelo quantizador μ invertido e seu respectivo erro de quantização.	53
4.34	Análise transiente para um sinal senoidal processado pelo quantizador μ -C e seu respectivo erro de quantização.	53
4.35	DFTs dos quantizadores Linear (a) e Arco-seno (b).	54
4.36	DFTs dos quantizadores μ -C (a) e A-C (b).	55
4.37	Testbench do integrador [13].	56
4.38	Simulação do integrador [13].	57
4.39	<i>Testbench</i> do subtrator [13].	57
4.40	Simulação do subtrator [13].	58
4.41	Diagrama de blocos do modulador Σ - Δ <i>Multi-bit</i> de a) primeira e de b) segunda ordem (Adaptada de [10]).	59
4.42	Diagrama de blocos do quantizador proposto (Adaptada de [10]).	59
4.43	Simulação do modulador Σ - Δ <i>multi-bit</i> de 1ª ordem com quantizador linear. ...	60
4.44	Simulação do modulador Σ - Δ <i>multi-bit</i> de 1ª (a) e de 2ª (b) ordem com quantizador linear.	61
4.45	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1ª (a) e de 2ª (b) ordem com quantizador linear.	61
4.46	Simulação do modulador Σ - Δ <i>multi-bit</i> de 1ª ordem com quantizador Arco-seno.	62
4.47	Simulação do modulador Σ - Δ <i>multi-bit</i> de 1ª (a) e de 2ª (b) ordem com quantizador arco-seno.	63
4.48	Saída do modulador Σ - Δ <i>multi-bit</i> de 1ª ordem para diferentes amplitudes de entrada.	63
4.49	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1ª (a) e de 2ª (b) ordem com quantizador arco-seno.	64
4.50	Diagrama de blocos do modulador Σ - Δ <i>Multi-bit</i> de 1ª (a) e de 2ª (b) ordem utilizando o quantizador μ -C.	65
4.51	Simulação do modulador Σ - Δ <i>Multi-bit</i> de 1ª ordem com quantizador μ -C.	65
4.52	Simulação do modulador Σ - Δ <i>multi-bit</i> de 1ª (a) e de 2ª (b) ordem com quantizador μ -C.	66
4.53	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1ª (a) e de 2ª (b) ordem com quantizador μ -C.	66
4.54	Simulação do modulador Σ - Δ <i>Multi-bit</i> de 1ª ordem com quantizador A-C. ...	67
4.55	Simulação do modulador Σ - Δ <i>multi-bit</i> de 1ª (a) e de 2ª (b) ordem com quantizador A-C.	67

4.56	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizador <i>A-C</i>	68
4.57	Sinais de entrada, saída e erro de quantização para o modulador Σ - Δ <i>Multi-bit</i> de a) primeira e de b) segunda ordem com quantizador linear.	69
4.58	Sinais de entrada, saída e erro de quantização para o modulador Σ - Δ <i>Multi-bit</i> de a) primeira e de b) segunda ordem com quantizador arco-seno.	69
4.59	Análise transiente do modulador Σ - Δ <i>Multi-bit</i> de a) primeira e de b) segunda ordem com quantizador μ - <i>C</i>	70
4.60	Análise transiente do modulador Σ - Δ <i>Multi-bit</i> de a) primeira e de b) segunda ordem com quantizador <i>A-C</i>	70
4.61	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizadores linear e arco-seno.	71
4.62	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1 ^a e de 2 ^a ordem com quantizadores linear e arco-seno e μ - <i>C</i>	71
4.63	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizadores linear e arco-seno e <i>A-C</i>	72
4.64	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizadores linear, arco-seno e μ - <i>C</i> e marcação dos harmônicos principais.	72
4.65	<i>DFTs</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizadores linear, arco-seno e <i>A-C</i> e marcação dos harmônicos principais.	73
4.66	Gráfico da <i>SFDR</i> para os moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizador Linear, Arco-seno, μ - <i>C</i> e <i>A-C</i>	76
4.67	Gráfico da <i>SNHR</i> para os moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizador Linear, Arco-seno, μ - <i>C</i> e <i>A-C</i>	77
4.68	Gráfico da <i>SINAD</i> para os moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizador Linear, Arco-seno, μ - <i>C</i> e <i>A-C</i>	78
4.69	Gráfico do <i>ENOB</i> para os moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem com quantizador Linear, Arco-seno, μ - <i>C</i> e <i>A-C</i>	79
B.1	Respostas dos quantizadores.	102
B.2	Respostas dos moduladores Σ - Δ <i>multi-bit</i> de 1 ^a (a) e de 2 ^a (b) ordem.	103

LISTA DE TABELAS

4.1	Parâmetros de simulação comuns aos blocos.	31
4.2	Descrição dos pinos do comparador [13].	31
4.3	Parâmetros de simulação do comparador [13].	32
4.4	Códigos de entrada (termômetro) e saída (binário) do encoder projetado [13]. ..	33
4.5	Descrição dos pinos do Encoder [13].	34
4.6	Parâmetros de simulação do encoder [13].	34
4.7	Descrição dos pinos do <i>ADC</i> [13].	35
4.8	Códigos utilizados no projeto do decoder [13].	36
4.9	Descrição dos pinos do decoder [13].	37
4.10	Parâmetros de simulação do decoder [13].	37
4.11	Parâmetros de simulação do quantizador com sinal de entrada senoidal [13]. ..	38
4.12	Parâmetros de simulação do quantizador com sinal de entrada do tipo rampa [13].	39
4.13	Resistências do <i>ADC</i> e do <i>DAC</i> para o quantizador Arco-seno [13].	40
4.14	Parâmetros de simulação do quantizador com sinal de entrada senoidal.	42
4.15	Parâmetros de simulação do quantizador com sinal dente de serra.	42
4.16	Descrição dos pinos do integrador [13].	56
4.17	Parâmetros de simulação do integrador [13].	56
4.18	Descrição dos pinos do subtrator [13].	57
4.19	Parâmetros de simulação para o subtrator [13].	58
4.20	Especificações do modulador Σ - Δ de 1ª e de 2ª ordem.	60
4.21	<i>THD</i> dos moduladores Σ - Δ <i>multi-bit</i> de 1ª e de 2ª ordem projetados com os quantizadores Linear, Arco-seno, μ -C e A-C.	74
4.22	Parâmetros dinâmicos dos moduladores Σ - Δ <i>multi-bit</i> de 1ª e de 2ª ordem projetados com os quantizadores Linear e Arco-seno.	74
4.23	Parâmetros dinâmicos dos moduladores Σ - Δ <i>multi-bit</i> de 1ª e de 2ª ordem projetados com os quantizadores μ -C e A-C.	75

LISTA DE SÍMBOLOS

Σ	Sigma
Δ	Delta
v_{dd}	Tensão de referência positiva
v_{ss}	Tensão de referência negativa
A_0	Amplitude do sinal de entrada
dB	Decibéis
$dBFS$	Decibéis Full Scale
clk	Clock
f_0	Frequência do sinal
f_s	Frequência de amostragem
fb	Largura de banda do sinal
f_{in}	Frequência do sinal de entrada
H_n	Amplitude das n harmônicas
Hz	Hertz
K	Quilo
m	mili
n	Nano
N	Número de bits
P_e	Soma da potência do ruído de quantização
P_n	Soma da potência dos ruídos analógicos
P_h	Potência dos harmônicos
P_s	Potência do sinal de entrada
q	Número de níveis de quantização
td	Tempo de atraso
tf	Tempo de transição
μ	Mu
V_{in}	Tensão de entrada
V_{out}	Tensão de saída

LISTA DE ACRÔNIMOS E ABREVIACÕES

A/D	Analog-to-Digital
ADC	Analog-to-Digital Converter
AMS	Analog Mixed-Signal
BER	Bit Error Rate
D/A	Digital-to-Analog
DAC	Digital-to-Analog Converter
DNL	Differential Non-Linearity
DR	Dynamic Range
ENOB	Effective Number of Bits
HDL	Hardware Description Language
HDTV	High Definition Television
INL	Integral Non-Linearity
LSB	Least Significant Bit
PAM	Pulse Amplitude modulation
PDF	Probability Density Function
PSRR	Power Supply Rejection Ratio
SINAD	Signal-to-Noise and Distortion Ratio
SNDR	Signal-to-Noise Plus Distortion Ratio
SNHR	Signal Non-harmonic Ratio
SNR	Signal-to-Noise Ratio
S/H	Sample and Hold
SFDR	Spurious-Free Dynamic Range
THD	Total Harmonic Distortion
UT	Unscented Transform
Verilog	Verifying Logic
VHDL	VHSIC Hardware Description Language
OSR	Oversampling Rate

1 INTRODUÇÃO

1.1 QUANTIZAÇÃO NÃO LINEAR

Na cadeia de conversão analógico-digital, a quantização é o processo de substituição de amostras analógicas por valores aproximados que podem ser especificados por um sinal digital. Geralmente os conversores empregados nessas aplicações utilizam quantizadores lineares, onde todos os níveis de quantização são iguais [14]. No entanto, para algumas aplicações, um quantizador não linear pode ser mais apropriado.

Analisando-se um sinal de fala, por exemplo, para a maioria dos canais de comunicação de voz, volumes de voz muito baixos predominam em mais da metade das vezes e volumes de grande amplitude são raros [15]. Nesse contexto, a utilização de um quantizador linear iria gerar grandes erros de quantização para sinais de valores pequenos. Estes erros podem ser da mesma grandeza que o próprio sinal fazendo com que a relação sinal-ruído (*Signal-to-Noise Ratio - SNR*) não seja suficientemente grande para recuperar toda a informação do sinal.

Um dos métodos para resolver esse problema é a utilização de um quantizador não linear. Um quantizador não linear gera passos de quantização não uniformes que podem ser menores para as amplitudes de sinal mais baixas e maiores para as amplitudes de sinal mais altas, tornando o processo de quantização mais eficiente ao utilizar níveis de quantização que seriam desperdiçados com um quantizador linear.

A Figura 1.1 mostra uma comparação entre os processos de quantização linear e não linear para dois sinais distintos, um sinal de baixa amplitude e um sinal de alta amplitude. Como pode-se observar, a quantização linear reproduz de forma ineficiente o sinal fraco e a maior parte dos níveis de quantização não foram utilizados neste caso. Já no caso na quantização não linear, mais níveis de quantização foram utilizados e o sinal fraco foi melhor reproduzido.

A proposta deste trabalho é a implementação de quantizadores não lineares utilizando-se diferentes técnicas e a aplicação em conversores analógico-digitais afim de comparar seus desempenhos com relação a um quantizador linear.

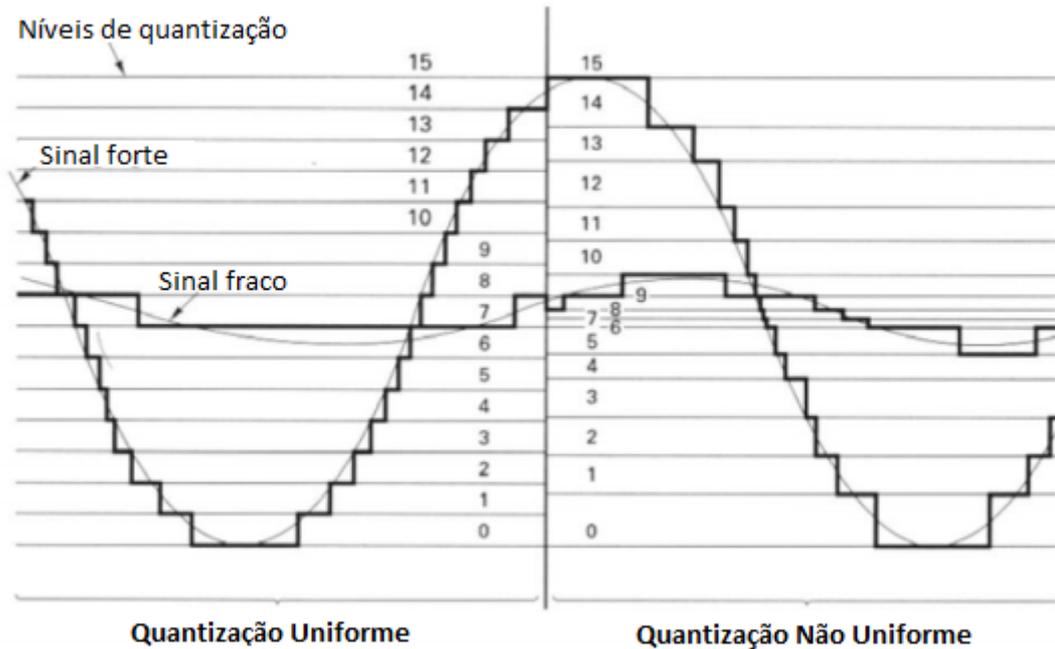


Figura 1.1 – Comparação entre os processos de quantização uniforme e não uniforme para dois sinais distintos, um sinal forte e um sinal fraco (adaptada de [1]).

1.2 OBJETIVOS E ESCOPO

O principal objetivo desta dissertação é a implementação e a análise de quantizadores não lineares utilizando-se diferentes técnicas para a aplicação em conversores analógico-digitais. Os quantizadores não lineares serão projetados utilizando-se os mesmos parâmetros de simulação e também será projetado um quantizador linear afim de comparar o desempenho dos sistemas.

Para alcançar o objetivo, são propostos os seguintes objetivos específicos:

- Desenvolver uma topologia de quantizador não linear utilizando-se a técnica de *companding* baseada da lei- μ ;
- Desenvolver uma topologia de quantizador não linear utilizando-se a técnica de *companding* baseada da lei- A ;
- Desenvolver uma topologia de quantizador não linear utilizando-se a Transformada da Incerteza;
- Desenvolver uma topologia de quantizador linear para ser usado como referência;
- Desenvolver uma topologia de modulador Sigma-Delta *Multi-bit* de 1ª e de 2ª ordem para a aplicação dos quantizadores;

- Implementar, simular e analisar os quatro quantizadores propostos;
- Implementar, simular e analisar os moduladores Sigma-Delta *Multi-bit* de 1ª e de 2ª ordem utilizando-se os quatro quantizadores propostos;
- Comparar o desempenho dos moduladores projetados;

1.3 METODOLOGIA DE PROJETO

No contexto de projetos na área de Eletrônica, as abordagens mais utilizadas no fluxo de projeto são as metodologias *Bottom-up* e *Top-down*. A metodologia *bottom-up* é do tipo ascendente, ou seja, ela parte do mais baixo para o mais alto nível de abstração. Nesse tipo de metodologia, o projeto começa com os blocos individuais de cada circuito, que depois são combinados para a formação de subsistemas. Esses subsistemas são combinados para formarem subsistemas maiores, e assim sucessivamente até chegar no sistema completo. O risco associado a esse método é que o desempenho exigido do sistema pode não ser atingido com os blocos individuais projetados e conectados entre si, o que implicaria no reprojeto de blocos do sistema [16].

A metodologia *Top-down* é do tipo descendente, ou seja, ela parte do mais alto para o mais baixo nível de abstração. Nesse tipo de metodologia, o desempenho de cada bloco individual é cuidadosamente analisado e compreendido antes dos blocos começarem a ser desenvolvidos. Essa prática reduz a necessidade de reprojeto dos blocos, mas com o risco de que o desempenho esperado por um ou mais blocos seja inalcançável, o que exigiria a revisão do projeto do sistema [17].

Um processo de projeto *Top-down* parte de uma arquitetura para um sistema a nível de transistores. Cada nível é totalmente projetado antes de se projetar o próximo e cada nível é aproveitado no projeto do próximo. Sendo assim, pode-se dividir o projeto em blocos menores e bem definidos, permitindo que mais projetistas possam trabalhar de forma mais organizada. Essa prática tende a reduzir o tempo de projeto, melhorar a comunicação entre projetistas e permitir que estes trabalhem juntos em diferentes locais. Seguir uma metodologia *Top-down* também reduz o impacto nas mudanças que podem ocorrer ao longo do projeto. Se por alguma razão o sistema tiver que ser parcialmente reprojetado, a alteração poderá ser feita rapidamente, os modelos poderão ser atualizados e o impacto sobre o resto do sistema será avaliado de forma rápida [17].

A metodologia *Top-down* foi a escolhida para este trabalho com a finalidade de testar os modelos de quantizadores não lineares e realizar a análise e comparação entre o desempenho de cada sistema antes de iniciar a sua implementação.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

Essa dissertação está dividida de 5 capítulos. No capítulo 1 é apresentada a problemática, os objetivos, o escopo e a organização do trabalho. O Capítulo 2 é composto pela revisão bibliográfica que aborda os aspectos necessários para a implementação deste trabalho. Neste capítulo estão inclusos os aspectos fundamentais da metodologia de projeto escolhida, os conceitos básicos para o funcionamento de um conversor analógico-digital, parâmetros importantes para a análise dos conversores e as topologias de interesse para a aplicação nessa dissertação. O Capítulo 3 apresenta as ferramentas necessárias e as técnicas para implementação dos quantizadores não lineares propostos. Essas ferramentas incluem a Transformada da Incerteza e a técnica de *Compadding* Logarítmico. O Capítulo 4 descreve o projeto e a implementação dos quantizadores e dos moduladores propostos. Esse capítulo mostra o processo de modelagem e simulação de cada bloco dos sistemas e mostra os resultados obtidos assim como a comparação entre os quantizadores e entre os moduladores projetados. O Capítulo 5 apresenta a conclusão, onde são expostas as considerações finais do projeto, e apresenta uma breve descrição de trabalhos futuros que possam dar continuidade a essa dissertação.

2 REVISÃO BIBLIOGRÁFICA

Esse capítulo apresenta alguns conceitos e técnicas necessárias para a implementação, simulação e análise dos sistemas propostos e está organizado da seguinte forma: A Seção 2.1 aborda conceitos sobre metodologias de projeto e apresenta a linguagem utilizada no desenvolvimento dessa dissertação. Na Seção 2.2 são introduzidos conceitos fundamentais sobre conversores *A/D* como o seu funcionamento. A Seção 2.3 apresenta parâmetros importantes para análise estática de conversores *A/D* e a Seção 2.4 para a análise dinâmica. Por fim, a Seção 2.5 apresenta algumas topologias de conversores *A/D*.

2.1 LINGUAGENS DE DESCRIÇÃO DE *HARDWARE*

As Linguagens de descrição de *Hardware* (Hardware Description Language - *HDL*) são utilizadas no projeto de *Hardware* por meio da descrição comportamental dos mesmos. Diferente das linguagens de programação que descrevem algoritmos e sequências de operações para dados ou periféricos, nos sistemas de *Hardware* existem diversos componentes diferentes que operam simultaneamente. Para modelá-los de forma adequada é necessário descrever tanto o comportamento dos componentes quanto a ligação que eles têm com os outros componentes do circuito.

As *HDLs* são utilizadas principalmente na simulação e síntese de projetos. A síntese é o processo de projeto do *hardware*, nessa etapa utiliza-se a *HDL* para descrevê-lo em um nível abstrato usando modelos de componentes que ainda não possuem implementação física. A simulação é o processo onde aplica-se estímulos à uma modelo executável, criado na síntese, para prever como ele irá reagir. Isso permite analisar detalhadamente a complexidade dos sistemas e diminuir o tempo e custo de implementação [17].

As *HDLs* podem descrever sistemas digitais, analógicos ou mistos. Atualmente as duas *HDLs* mais utilizadas para descrever sistemas mistos são o Verilog-AMS e VHDL-AMS, que são extensões das *HDLs* Verilog e VHDL.

2.1.1 Verilog

O Verilog-AMS é uma linguagem de modelagem para sistemas de sinais mistos, ou seja, sistemas que operam com sinais digitais e sinais analógicos. O Verilog-AMS trabalha com a incorporação e ampliação de duas línguas, Verilog-HDL e Verilog-A. Estas três linguagens formam a família Verilog. O Verilog-HDL permite a modelagem de componentes digitais e o

Verilog-A permite a modelagem de componentes analógicos. O Verilog-AMS combina estas duas *HDLs* e adiciona também a capacidade de descrição de componentes de sinal misto [17]. A relação entre essas três *HDLs* pode ser vista na Figura 2.1.

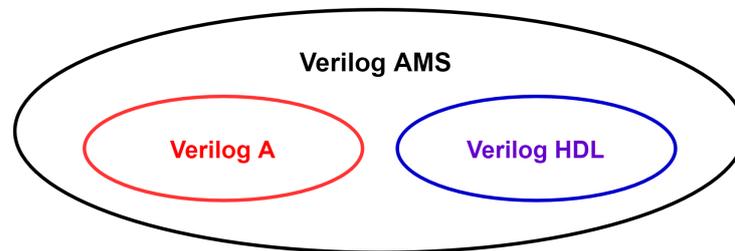


Figura 2.1 – Relação entre Verilog-AMS, Verilog-A e Verilog-HDL.

Neste trabalho será utilizada a *HDL* Verilog-A. A seção a seguir é dedicada à exposição desta linguagem.

2.1.1.1 Verilog-A

O Verilog-A é um subconjunto do Verilog-AMS que processa sinais analógicos. Ele pode ser utilizado em sistemas elétricos e mecânicos. Na eletrônica, o Verilog-A é muito utilizado no projeto e validação de sistemas analógicos e circuitos integrados através da criação de módulos que descrevem o comportamento de alto nível dos componentes ou do sistema [17].

Neste trabalho utilizou-se a *HDL* Verilog-A para a modelagem dos principais blocos dos quantizadores Linear e Arco-seno e dos moduladores Σ - Δ *multi-bit*.

2.2 CONVERSORES ANALÓGICO-DIGITAIS

Conversores analógico-digitais (*Analog-to-Digital Converters - ADCs*) são sistemas capazes de gerar uma representação digital a partir de uma grandeza analógica, geralmente um sinal de tensão ou de corrente elétrica. Os *ADCs* são muito utilizados na interface entre dispositivos digitais e dispositivos analógicos e são vastamente utilizados em aplicações para produtos de consumo, como eletrônicos no geral, e em sistemas especializados, como imagiologia médica, processamento de voz, instrumentação, leitura de sensores e chão de fábrica [18].

A Figura 2.2 mostra o diagrama de blocos de um *ADC* genérico e a figura 2.3 como ocorre esse processo de conversão. Como pode-se observar, o primeiro bloco do *ADC* é um pré-filtro ou filtro *anti-aliasing*, geralmente representado por um filtro passa-baixa ou passa-banda, utilizado para atenuar as componentes de alta frequência do sinal. Pelo teorema

de *Nyquist*, a frequência de amostragem (f_s) deve ser pelo menos duas vezes maior ou igual a componente de maior frequência do sinal, sendo assim, essa pré-filtragem descarta as frequências que não são essenciais para a informação contida no sinal.

O filtro *anti-aliasing* é seguido por um circuito de *sample-and-hold* (*S/H*) que executa a etapa de amostragem do sinal. Nessa etapa, O *S/H* realiza aquisição do sinal (*sample*), em que a tensão é registrada, e a de retenção (*hold*), em que a tensão não é alterada por um determinado período tempo igual a $1/f_s$.

Depois de amostrado o sinal passa pelo processo de quantização, foco do presente trabalho, onde são atribuídos valores de amplitude a cada sinal amostrado. Esse valor depende da resolução do *ADC* que define o número de níveis de quantização.

Por último, o sinal é codificado por meio de um *encoder*, e o sinal já amostrado e quantizado passa a ser representando em códigos digitais. Dentre os mais utilizados estão o código binário, gray e o termômetro [2].

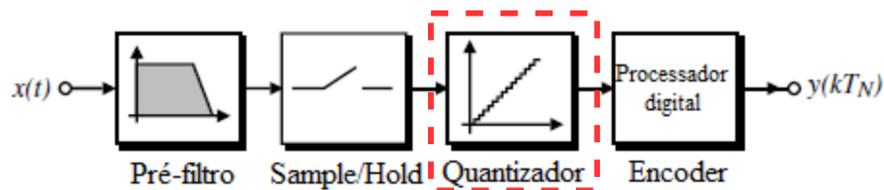


Figura 2.2 – Diagrama de blocos de um *ADC* genérico adaptada de [2]

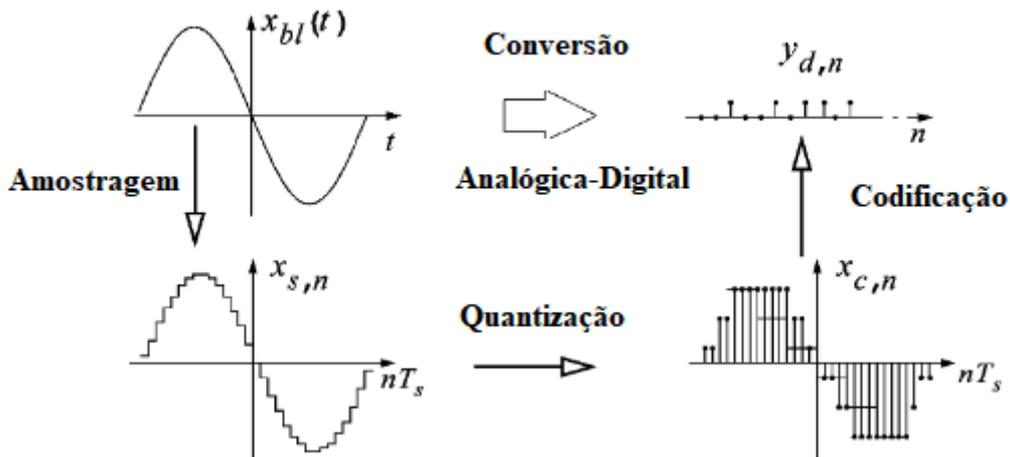


Figura 2.3 – Processo de conversão analógica-digital (Adaptada de [3]).

2.3 CARACTERÍSTICAS ESTÁTICAS EM CONVERSORES A/D

A caracterização de um *ADC* envolve a análise de seu comportamento estático e dinâmico. As características estáticas são baseadas no sinal de entrada e de saída do *ADC* analisados no domínio do tempo e podem ser verificadas por meio da diferença entre esses dois sinais. A presença de espaçamento não ideal ou uniforme nos níveis de transição entre os códigos pode ser indicativo de erros estáticos no componente [19]. As principais características que definem o desempenho estático dos conversores são o erro de *offset*, erro de ganho, não linearidade integral (*Integral Non-Linearity - INL*) e não linearidade diferencial (*Differential Non-Linearity - DNL*) [2].

2.3.1 Erro de *Offset*

O erro de *Offset* é o desvio da função de transferência do *ADC* com relação a reta de referência quando o sinal de entrada for 0 *LSB* (*Least Significant Bit*). Quando a transição de 0 para 1 não ocorre com de 0.5 *LSB* na entrada, ocorre esse erro. A Figura 2.4 ilustra o efeito do erro de *offset* para dois casos.

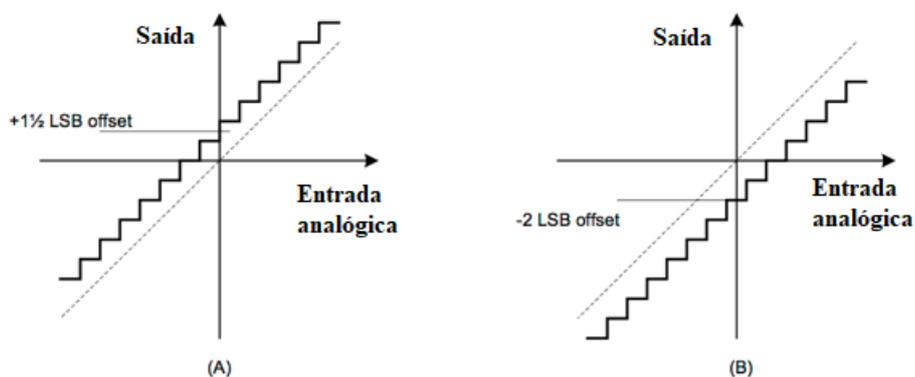


Figura 2.4 – Erro de offset de a) $+1.5LSB$ b) $-2LSB$ em um *ADC* (Adaptada de [4]).

2.3.2 Erro de Ganho

O erro de ganho acontece quando há um desvio no último passo da saída do *ADC* com relação à reta de referência de referência após a compensação do erro de *offset*. Esse erro altera a inclinação da curva de transferência e pode ser compensado escalonando-se os valores das amostras digitais. A figura 2.5 ilustra o efeito do erro de ganho para dois casos distintos.

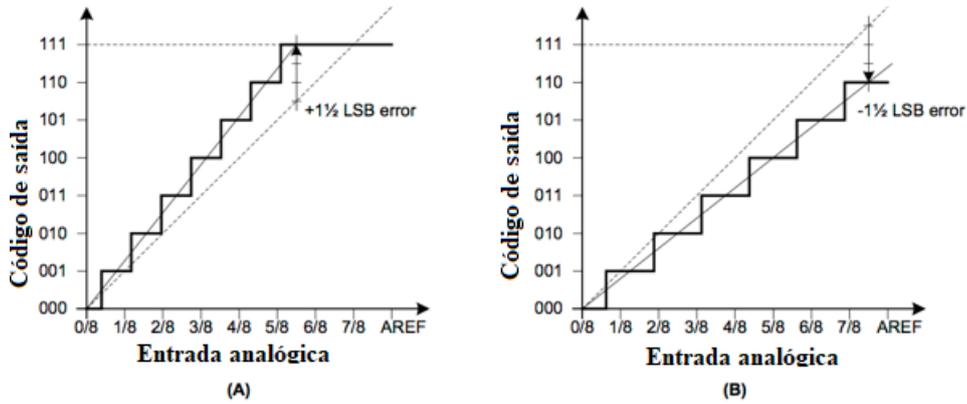


Figura 2.5 – Erro de ganho de a) para +1.5 *LSB*, b) -1.5 *LSB* em um *ADC* (Adaptada de [4]).

2.3.3 Não Linearidade Diferencial

A Figura 2.6 mostra um exemplo de erros de não linearidade em conversores *A/D*. Como pode-se observar, esses erros representam uma deformação no sinal de saída do *ADC*.

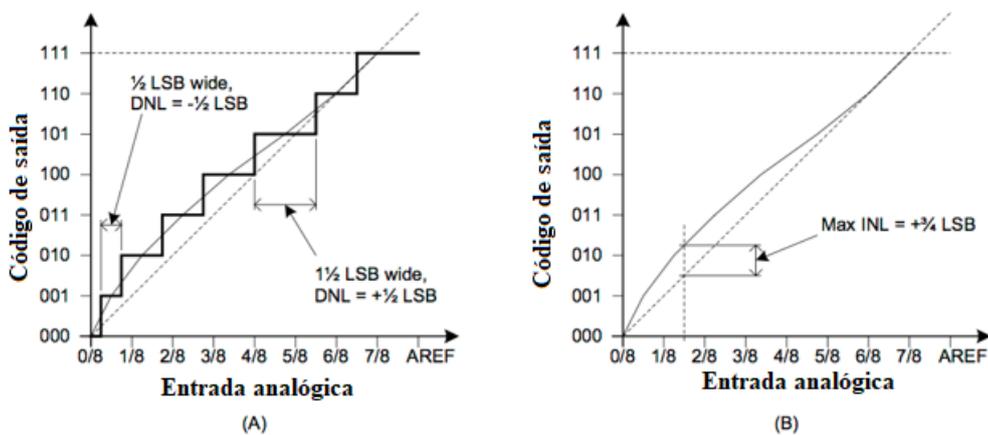


Figura 2.6 – Erros de não linearidade a) *DNL* e b) *INL* para a curva de um *ADC* (Adaptada de [4]).

O erro de *DNL* de um código x é igual ao desvio do valor ideal e pode ser expresso pela equação 2.1.

$$DNL = \frac{H(x) - H(x)_{ideal}}{H(x)_{ideal}} = \frac{H(x)}{H(x)_{ideal}} - 1 \quad (2.1)$$

Sendo $H(x)$ o comprimento do código x do *ADC* e $H(x)_{ideal}$ o comprimento do código x para um *ADC* com a mesma resolução ideal.

2.3.4 Não Linearidade Integral

Os erros de DNL para um código x podem ser acumulados por uma série finita de códigos, o que causará uma variação na curva ideal e resultará no erro INL expresso pela equação 2.2 [19].

$$INL(x) = \sum_{k=1}^x DNL(k) \quad (2.2)$$

2.4 CARACTERÍSTICAS DINÂMICAS EM CONVERSORES A/D

A análise estática de um ADC é feita no domínio do tempo. Ela caracteriza os conversores em DC ou em baixa frequência. Já a análise dinâmica é feita no domínio da frequência e geralmente é utilizada em aplicações onde o sinal de entrada varia rapidamente. Para esses casos a caracterização DC não é suficiente, sendo necessário também a análise AC [4].

Algumas métricas extraídas da análise dinâmica são frequentemente utilizadas na análise de performance dos moduladores Σ - Δ , principalmente as relacionadas diretamente com a análise espectral. Geralmente o sinal usado para extração dessas métricas no modulador é um sinal senoidal [5].

2.4.1 Relação Sinal-Ruído (SNR)

A relação sinal-ruído (*Signal to Noise Ratio* - SNR) é um dos parâmetros dinâmicos mais importantes em um conversor. Ela pode ser definida como a razão entre a potência do sinal e a potência total do ruído. O cálculo da SNR depende da resolução do conversor, da sua linearidade, distorção, incertezas na amostragem, ruídos eletrônicos e tempo de estabilização. A equação 2.3 mostra como é feito o seu cálculo teórico [20].

$$SNR_{dB} = 10 \log_{10} \frac{P_S}{P_e + P_n} \quad (2.3)$$

Onde P_S é a potência do sinal de entrada, P_e é a potência do ruído de quantização e P_n é a potência do ruído gerado pelos blocos analógicos.

Para um sinal que ocupa todo o *range* entrada, P_S pode ser definido pela equação:

$$P_S = \frac{(q\Delta)^2}{8} \quad (2.4)$$

Onde q representa o número de níveis de quantização, e Δ representa a diferença entre

dois níveis de quantização.

2.4.2 Distorção Harmônica Total (*THD*)

A distorção total harmônica (*Total Harmonic Distortion - THD*) pode ser definida como a razão entre a soma das harmônicas presentes no sinal de saída do *ADC* e a amplitude do sinal de entrada (A_0) [19]. Teoricamente, a distorção harmônica deveria ser calculada com base em todas as componentes harmônicas presentes no sinal de saída do *ADC*, no entanto, para o cálculo desse parâmetro, como pode ser visto na equação 2.5, são utilizadas apenas as cinco primeiras harmônicas. Isso é o suficiente porque as harmônicas de ordem superior têm efeito desprezível no cálculo da *THD* [21].

$$THD = \frac{\sqrt{H1^2 + H2^2 + H3^2 + H4^2 + H5^2}}{A_0} \quad (2.5)$$

2.4.3 Faixa Dinâmica (*DR*)

A faixa dinâmica (*Dynamic Range - DR*) é a razão entre a potência de saída do modulador, para um sinal senoidal de entrada com máxima amplitude pico a pico, e a potência total dos ruídos. Ela também pode ser definida como a potência do sinal de entrada onde a *SNR* é 0 *dB* como pode ser visto na figura 2.7.

2.4.4 Relação Sinal-Ruído e Distorção (*SNDR*)

A Relação Sinal Ruído e Distorção (*Signal to Noise and Distortion Ratio - SNDR*) também conhecida como *SINAD* pode ser definida como a razão entre a potência do sinal (P_S) e a soma da potência do ruído de quantização (P_e), dos ruídos analógicos (P_n) e da potência dos harmônicos (P_h). Sua fórmula pode ser vista na equação 2.6.

$$SNDR_{dB} = 10 \log_{10} \frac{P_S}{P_e + P_n + P_h} \quad (2.6)$$

A *SNDR* é um parâmetro muito utilizado na medição de performance de moduladores Σ - Δ pois inclui todos os efeitos de ruído, distorção e harmônicas introduzidas pelo *ADC* [22]. A figura 2.7 mostra o gráfico das medidas reais e ideais de *SNR*, *SNDR* e *DR* para um modulador Σ - Δ . Como pode-se observar na figura, esses parâmetros dependem da potência do sinal de entrada e geralmente a *SNDR* máxima é menor que a *SNR* máxima.

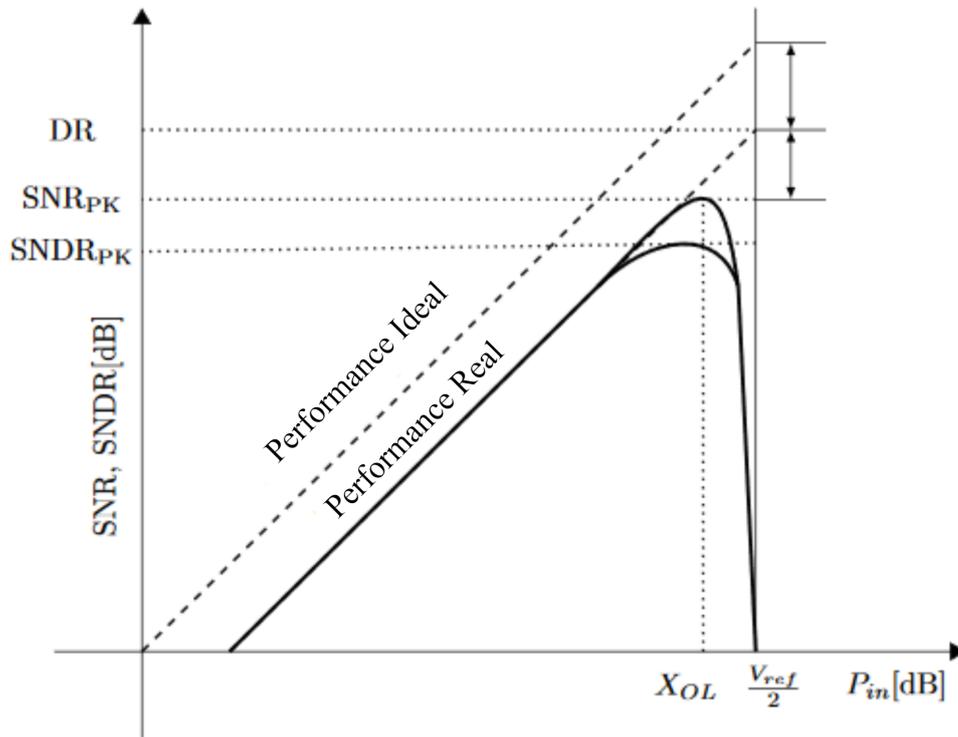


Figura 2.7 – Exemplo de gráfico da SNR e $SNDR$ e DR vs Potência do sinal entrada para um modulador Σ - Δ (Adaptada de [5]).

2.4.5 Número Efetivo de Bits ($ENOB$)

O número efetivo de bits (*Effective Number of Bits* - $ENOB$) é a medida que avalia a resolução real do ADC , ou seja, é a quantidade de bits úteis que o ADC entrega. Esse parâmetro é muito importante pois um conversor de N bits, por exemplo, pode ter alguns bits corrompidos por ruído e distorção entregando assim, uma resolução menor [4]. A equação 2.7 mostra como é feito o cálculo do $ENOB$.

$$ENOB = \frac{SNDR_{measured} - 1.76}{6.02} \quad (2.7)$$

2.4.6 Faixa Dinâmica Livre de Espúrios ($SFDR$)

A Faixa Dinâmica Livre de Espúrios (*Spurious-Free Dynamic Range* - $SFDR$) pode ser definida como a medida de pureza espectral do conversor. Ela está diretamente relacionada a quantidade de distorção dinâmica causada pelo circuito e pode ser extraída numa análise direta do espectro em frequência, como pode ser observado na figura 2.8.

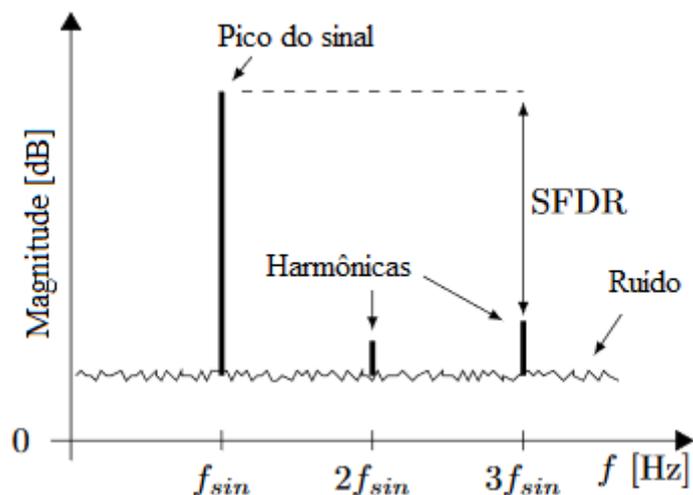


Figura 2.8 – SNR e $SFDR$ de um sinal aleatório [4].

Como mostra a figura 2.8 a $SFDR$ é a diferença entre a componente de sinal e a maior componente de distorção do espectro. Sua fórmula é dada pela equação 2.8.

$$SFDR_{dB} = 20 \times \log \frac{Ff}{HS} \quad (2.8)$$

Onde Ff é a frequência fundamental do sinal e HS é o maior espúrio do espectro.

2.5 TOPOLOGIAS DE CONVERSORES A/D

Para maximizar a largura de banda de entrada de um conversor, deseja-se fazer a largura de banda do sinal de entrada (fb) o mais próximo possível da metade da frequência de amostragem (fs). No entanto, isso requer um corte muito preciso para o filtro *anti-aliasing*, o que torna esse filtro complexo e de difícil implementação[2]. Os conversores analógico-digitais podem ser classificados de acordo com a sua taxa de amostragem. Os conversores de *Nyquist* convertem o sinal de entrada com taxa de amostragem próxima a frequência de Nyquist do sinal, ou seja, possuem fb muito próximo ou igual a $0.5fs$. Já os conversores com sobreamostragem (*Oversampling*), apresentam taxas muito maiores que a frequência de Nyquist e aumentam a SNR filtrando o ruído na banda do sinal e também por meio da técnica de formatação do ruído (*noise-shaping*) que joga parte do ruído de quantização para fora da banda do sinal de entrada.

A figura 2.9 mostra as diferentes topologias de conversores A/D e suas regiões de atuação de acordo com a sua resolução e taxa de amostragem.

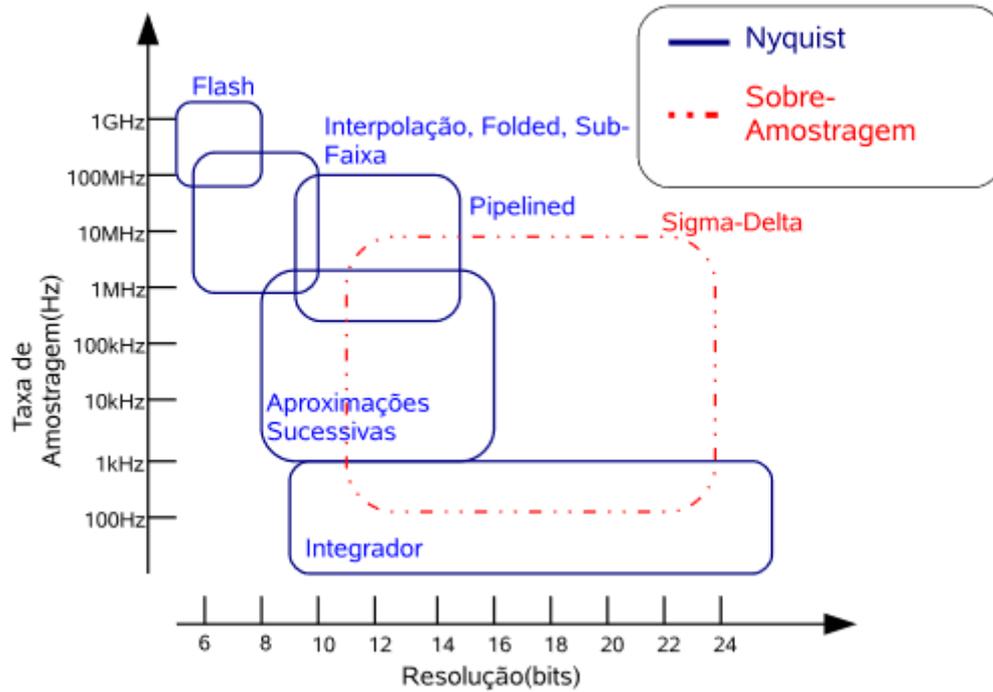


Figura 2.9 – Regiões de atuação das principais topologias de ADCs [6].

As próximas seções apresentarão as diferentes topologias de ADCs relacionadas ao desenvolvimento dessa dissertação.

2.5.1 Flash

O ADC paralelo, também conhecido como *flash* é umas das topologias de conversores que apresenta a maior velocidade de operação e é recomendado para aplicações que envolvem sinais com grande largura de banda [23]. Essa topologia, que pode ser observada na Figura 2.10, é baseada na comparação de tensão. Para um conversor de N bits, serão necessários $2^N - 1$ comparadores.

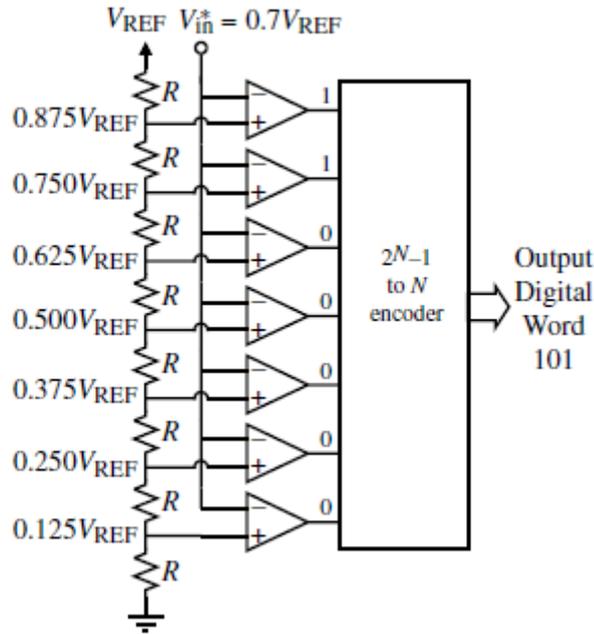


Figura 2.10 – *ADC flash* de 3 bits [2].

As entradas não-inversoras dos comparadores são conectadas entre si no ponto onde é aplicado o sinal analógico que será convertido. As tensões aplicadas nas entradas inversoras de cada um dos comparadores, são de um divisor resistivo, onde a tensão de referência para cada um é um bit menos significativo, maior que o do comparador abaixo. Cada comparador apresentará a saída com nível lógico alto quando a entrada analógica for maior que a sua tensão de referência. Na saída dos comparadores é necessário um circuito de codificação que irá receber os sinais dos comparadores e codificar o sinal de saída em código binário ou GRAY [2].

A desvantagem da utilização do *ADC flash* consiste no aumento do número de comparadores e da complexidade do codificador à medida que se aumenta a resolução, isso resulta no aumento na área de silício utilizada e no consumo de potência devido ao número elevado de componentes [2].

2.5.2 Sigma-Delta

Os *ADCs* baseados na modulação sigma-delta (Σ - Δ) são conversores sobreamostrados, ou seja, operam em frequências muito superiores à frequência de *Nyquist*. Sua topologia, apresentada na Figura 2.11, inclui um modulador Σ - Δ , um filtro digital, responsável pela remoção do ruído localizado fora da largura de banda de interesse e um decimador, cuja finalidade é reduzir a taxa de dados de saída de volta à taxa de *Nyquist*, funcionando assim como um filtro passa-baixas.

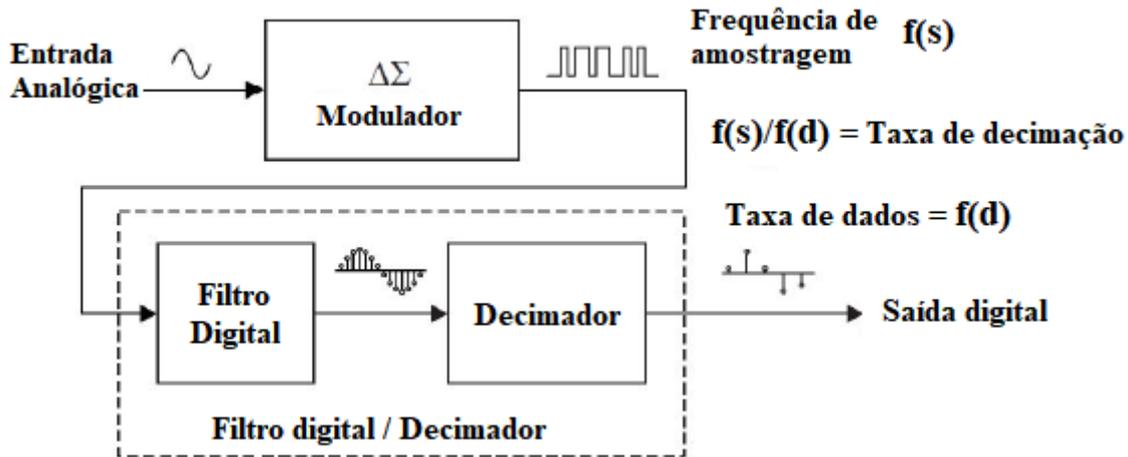


Figura 2.11 – Diagrama de blocos de um ADC Sigma-Delta (Adaptada de [7]).

A modulação (Σ - Δ) faz com que o ruído de quantização seja jogado em altas frequências no espectro e não espalhado por todo ele como ocorre em outros sistemas. Esse efeito é chamado de *noise-shaping* e serve para reduzir o ruído de quantização dentro da banda do sinal [7].

A sobreamostragem junto com o processo de decimação faz com que o conversor atinja uma alta resolução. O ADC Σ - Δ é amplamente utilizado em diversas aplicações pois apresenta como vantagens a alta precisão, alta resolução e uma menor sensibilidade às imperfeições dos circuitos analógicos. Por outro lado ele é aplicado apenas em situações onde não seja necessária uma alta velocidade, uma vez que o conversor é relativamente lento se comparado com outras topologias [24].

2.5.2.1 Fundamentos da Modulação Sigma-Delta

O modulador Σ - Δ é baseado na técnica de *oversampling*, essa técnica amostra o sinal analógico com frequência muito maior que a frequência de Nyquist como forma de evitar o fenômeno de *aliasing*. Dessa forma, é derivada a fórmula 2.9.

$$OSR = \frac{f_s}{2f_b} \quad (2.9)$$

Sendo *OSR* (*Oversampling Rate*) a taxa de *oversampling*, f_s a frequência de amostragem e f_b a frequência do sinal.

A sobreamostragem reduz a potência de ruído de quantização fixa, fazendo com que ela se propague sobre uma largura de banda muito maior do que a banda de sinal [25].

A Figura 2.12 mostra o diagrama de blocos de um modulador Σ - Δ de primeira ordem. Ele é constituído por um nó de diferença analógico, um integrador, um ADC de 1-bit, que

atua como quantizador, e um DAC de 1-bit num sistema realimentado. A saída do modulador tem apenas 1 bit de informação. A entrada para o integrador no modulador é a diferença entre o sinal de entrada $x(t)$ e o valor de saída quantizado $y(n)$ convertido de volta para o sinal analógico previsto. Essa diferença entre o sinal de entrada $x(t)$ e o sinal de retorno $y(t)$ na entrada do integrador é igual ao erro de quantização. Este erro é somado no integrador e depois quantizado pelo ADC de 1 bit [8].

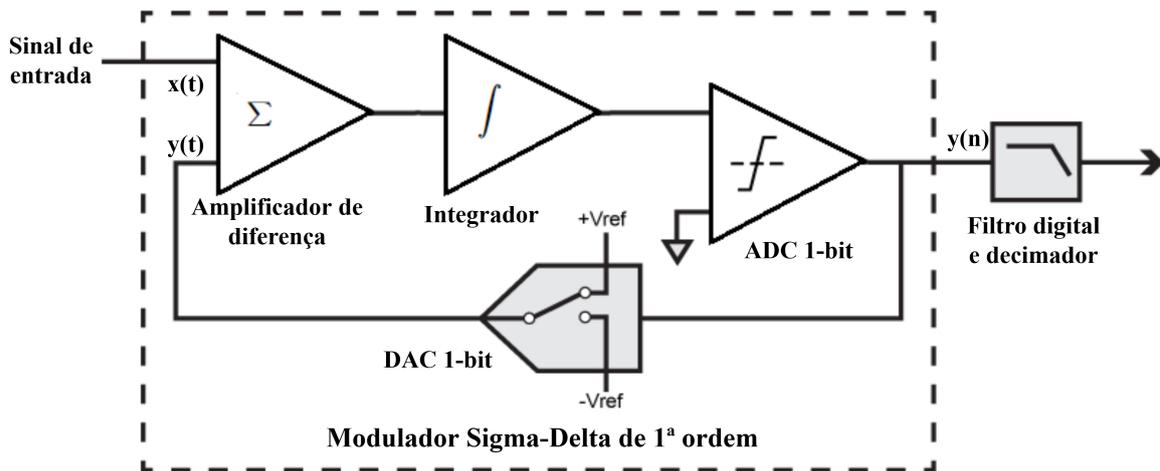


Figura 2.12 – Diagrama de blocos de um ADC Σ - Δ de 1ª ordem (Adaptada de [8]).

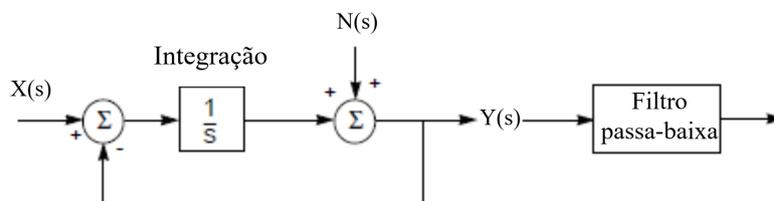


Figura 2.13 – Modelo de um modulador Σ - Δ no domínio S .

A figura 2.13 mostra um modelo simplificado no domínio S de um modulador Σ - Δ de 1ª ordem. O nó do lado direito do integrador representa o comparador, ou o quantizador, portanto, é nesse ponto que o ruído de quantização $N(s)$ é adicionado. Nota-se que quando o ruído gerado pelo quantizador é nulo, $N(S) = 0$, a função de transferência será:

$$\frac{Y(S)}{X(S)} = \frac{\frac{1}{S}}{1 + \frac{1}{S}} = \frac{1}{1 + S} \quad (2.10)$$

O que representa um filtro passa-baixas. O sinal então permanece inalterado desde que a sua frequência não exceda a frequência de corte do filtro. De forma parecida, quando o sinal de entrada é nulo, $X(S) = 0$, a função de transferência do ruído será representada pela

equação:

$$\frac{Y(S)}{N(S)} = \frac{1}{1 + \frac{1}{S}} = \frac{S}{1 + S} \quad (2.11)$$

O que representa um filtro passa-altas, indicando que o ruído foi empurrado para uma banda de frequência mais alta. Ou seja, neste processo de formação de ruído pelo modulador Σ - Δ , a potência do ruído de quantização da banda do sinal é empurrada para frequências mais elevadas.

A Figura 2.14 mostra as formas de onda de entrada (senoide) de saída para um modulador Σ - Δ de 1ª ordem. Quando a entrada senoidal está mais próxima da escala maior, a saída é positiva durante a maior parte dos ciclos de *clock*. O mesmo é válido quando a senoide está mais próxima da escala máxima negativa. Quando a entrada é próxima de zero, o valor da saída do modulador varia rapidamente entre uma escala máxima e mínima, com média de aproximadamente zero [8].

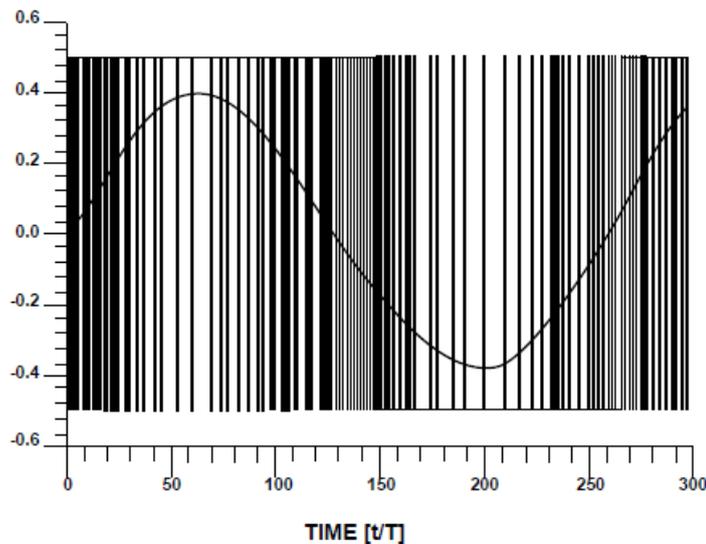


Figura 2.14 – Entrada e saída de um modulador Σ - Δ *single-bit* de 1ª ordem [8].

Um modulador de primeira ordem pode ser estendido a um modulador de segunda ordem inserindo-se um integrador adicional no *loop* do modulador, conforme mostrado na figura 2.15.

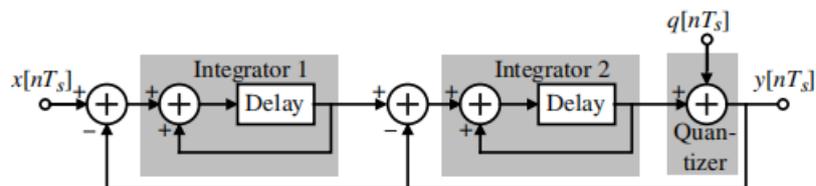


Figura 2.15 – Modelo de um modulador Σ - Δ de 2ª ordem [2].

2.5.2.2 Técnica de Quantização Multi-bit

Os moduladores Σ - Δ podem ser classificados de acordo com o número de bits do seu quantizador em *single-bit* ou *multi-bit*. O modulador apresentado na seção anterior trata-se de um modulador *single-bit*, já que utiliza um quantizador de 1 bit. A topologia *multi-bit* será apresentada nessa seção.

Como dito anteriormente, a topologia Σ - Δ é uma topologia de *ADC* amplamente utilizada em sistemas de aquisição de sinal pois é bastante robusta contra imperfeições de circuito e pode fornecer uma alta resolução e alta linearidade. Muitos conversores empregam um quantizador de bit único e um *DAC* de bit único no loop de feedback. A principal vantagem dessa técnica *single-bit* é a linearidade intrínseca do *DAC*. Essa linearidade no *feedback* é importante porque a saída do *DAC* é subtraída diretamente do sinal de entrada. Qualquer distorção, não-linearidade ou ruído na saída do *DAC* afetará diretamente o desempenho do modulador [7].

Por outro lado, a técnica *multibit* também é usada em *ADCs*. O aumento no número de níveis de quantização diminui o ruído de quantização aumentando a faixa dinâmica do *ADC* [26]. O modulador *multi-bit* também apresenta uma maior precisão na conversão, a redução do ruído de quantização e uma melhor estabilidade do sistema. Suas desvantagens implicam no aumento da complexidade do *ADC* e na necessidade de um *DAC* preciso [7]. A Figura 2.16 mostra o diagrama de blocos de um modulador *multi-bit* de N bits. Como pode-se observar, o quantizador de 1 bit, que poderia ser representado por um comparador na arquitetura *single-bit*, agora foi substituído por um *ADC* de N bits e um *DAC* com a mesma quantidade de bits.

A Figura 2.17 mostra uma implementação em nível de circuito de um modulador Σ - Δ multi-bit de 1ª ordem, usando um *ADC* e *DAC* de 4 bits. A Figura 2.18 mostra as saídas de simulação deste modulador nos domínios de tempo e frequência.

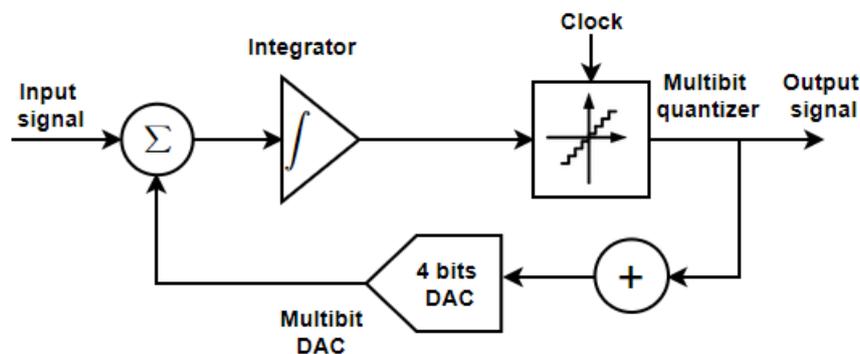


Figura 2.16 – Diagrama de blocos de um modulador multi-bit de 1ª ordem.

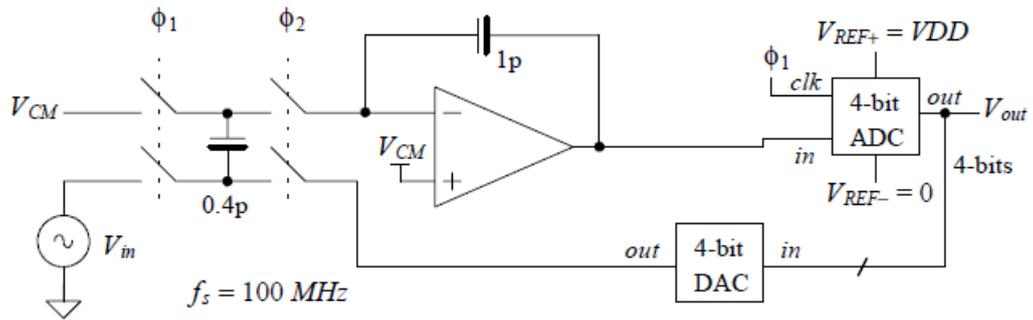


Figura 2.17 – Circuito de um modulador multi-bit de 1ª ordem [7].

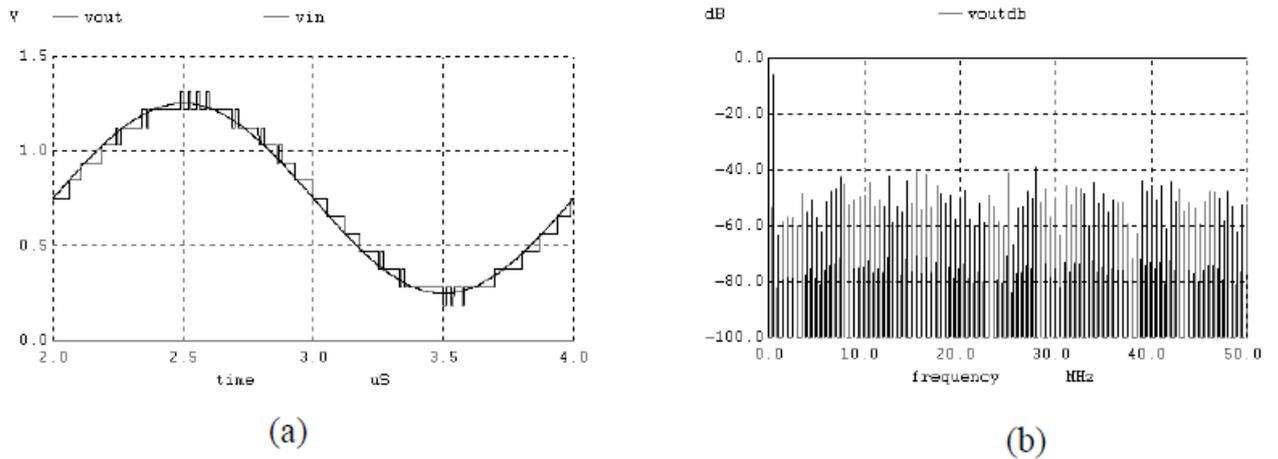


Figura 2.18 – Saída do modulador multi-bit no domínio do tempo a) e no domínio da frequência b) [7].

3 TÉCNICAS DE QUANTIZAÇÃO NÃO LINEAR

Esse capítulo apresenta as ferramentas e técnicas utilizadas para o projeto dos quantizadores não lineares apresentados neste trabalho e esta organizado da seguinte forma: A Seção 3.1 apresenta a Transformada da Incerteza (*Unscented Transform - UT*) e como ela pode ser aplicada como uma ferramenta para o projeto de quantizadores não lineares em conversores *A/D*. A Seção 3.2 explica os conceitos e o funcionamento da técnica de *Companding* e das leis logarítmicas, lei- μ e da lei- A , e como elas podem ser aplicadas no projeto de quantizadores não lineares.

3.1 A TRANSFORMADA DA INCERTEZA

A Transformada da Incerteza (*Unscented Transform - UT*) é uma estrutura matemática que modela uma função densidade de probabilidade (*Probability Density Function - PDF*) contínua em uma discreta [27]. A *UT* foi criada para resolver problemas de linearização e para realizar o cálculo estatístico de uma variável aleatória que sofreu uma transformação não linear [28]. Sua ideia geral consiste no fato de que é mais fácil aproximar uma distribuição de probabilidade do que aproximar uma função ou transformação de uma função não linear [29]. A figura 3.1 ilustra esse conceito.

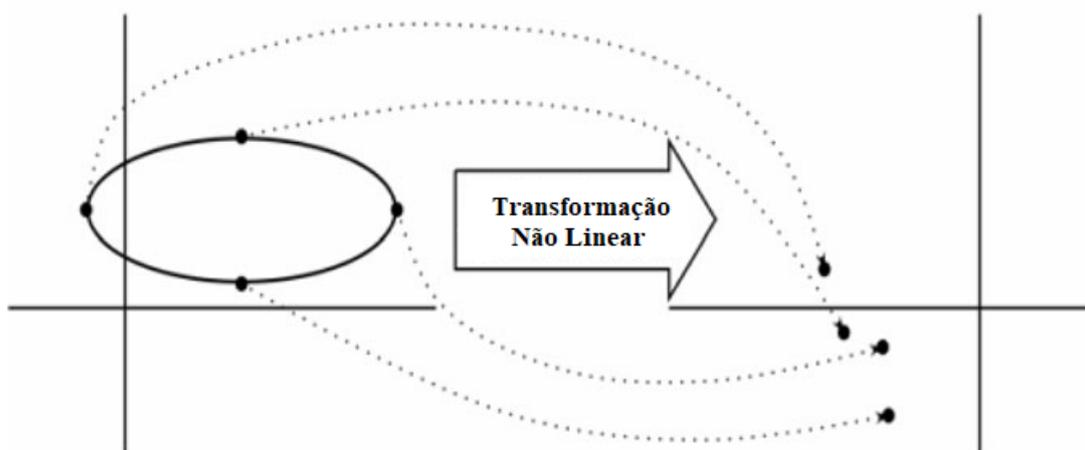


Figura 3.1 – Princípio da UT (Adaptada de [9]).

Na figura 3.1, um conjunto de pontos, deterministicamente escolhidos, é selecionado de forma que sua média e covariância sejam definidas. Uma função não linear é aplicada em

cada ponto, obtendo-se uma nuvem de pontos transformados. A estatística desses pontos transformados pode ser calculada para a formação de uma estimativa da média e covariância não lineares.

3.1.1 Quantizador Não Linear utilizando a *UT*

Dada uma variável aleatória x com média \bar{x} e covariância \sum_x , a *UT* é o conjunto de pontos sigma, S_i , cada um com probabilidade w_i onde a média e covariância dos pontos S_i são as mesmas que as de x . Esses pontos podem ser usados para capturar as influências do mapeamento não linear nas propriedades estatísticas de uma variável aleatória. [11] mostrou em seu trabalho que com os momentos da distribuição contínua de entrada x pode-se mapear o comportamento e derivar as condições necessárias relacionadas a distribuição de probabilidade discreta com os momentos de ordem superior, isto é

$$E\{X^k\} = \sum_{i=1}^n S_i^k w_i = \int_{-\infty}^{\infty} x^k p(x) dx \quad (3.1)$$

para $k = 0, 1, \dots, 2n + 1$ onde n é o número de pontos sigma, S_i , e $p(x)$ é PDF de x .

Em seu trabalho, [11] mostrou que a *UT* pode ser utilizada como um modelo para o processo de quantização em conversores *A/D*. Para isso, ele utilizou uma distribuição arco-seno e propôs uma implementação de circuito baseada em uma topologia *flash*, por esse motivo o quantizador por ele proposto será referenciado como Quantizador Arco-seno. Os resultados obtidos em sua tese mostraram que para o caso específico que foi analisado, o quantizador arco-seno obteve menor distorção comparado com o quantizador linear para sinais de amplitude alta, muito próximos a *DR* máxima.

A metodologia utilizada por [11], para a modelagem de um quantizador de N -níveis de quantização, começa pela procura de um conjunto de N valores de saída e de $N - 1$ níveis de *threshold*, th_n . Para isso ele pegou como entrada uma *PDF* e como saída os pontos sigma da *UT*, S_i . Para o cálculo dos níveis de *threshold* ele utilizou a seguinte equação:

$$P(th_{n-1} < x < th_n) = \int_{th_n}^{th_{n-1}} p_x(x) dx = w_n, \quad (3.2)$$

para garantir que a probabilidade do sinal de entrada estar no intervalo $[th_{n-1}, th_n]$, seja a mesma probabilidade, w_n , associada aos pontos sigma S_n .

Resolvendo a equação 3.2 para um caso geral, [11] encontrou os níveis de *threshold*

localizados em:

$$th_n = P_x^{-1} \left(\sum_n^{i=1} w_i \right), \quad (3.3)$$

onde $P_x^{-1}(P_x(x)) = x$ é conhecida como função quantil [11].

A figura 3.2 mostra a curva característica para um quantizador de 4 níveis utilizando a metodologia descrita por [11]. Na figura, pode-se observar que os níveis de saída do quantizador são extraídos diretamente dos pontos sigma da UT .

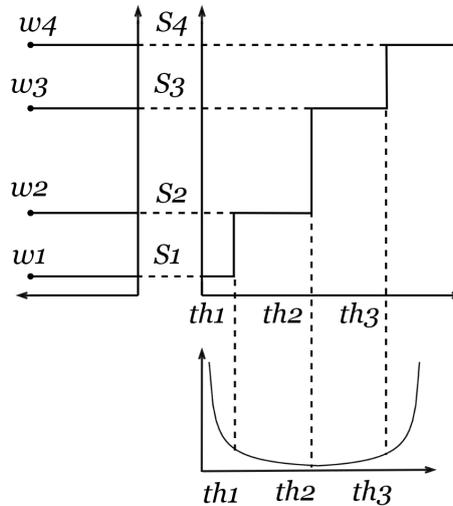


Figura 3.2 – Curva característica de um quantizador de 4 níveis com distribuição arco-seno e com um sinal de entrada senoidal [10].

3.1.1.1 Proposta de Implementação do Circuito

Após encontrar os *thresholds* e os níveis de saída, descritos na seção anterior, [11] propôs uma implementação de circuito baseado na topologia *flash*. Essa topologia é formada por um *ADC* seguido de um *DAC*. Para um quantizador de N -níveis, a rede resistiva do *ADC* será formada por resistores com valores dados por:

$$R_n = th_n \times R - \left(\sum_{k=0}^{n-1} R_k \right), \quad (3.4)$$

para todo $n=1,2,3... N$, onde R é um resistor com valor arbitrário, $R_0 = 0$ e $th_n = 1$.

E para a rede resistiva do *DAC*, os valores dos resistores são dados por:

$$R_n = S_n \times R - \left(\sum_{k=0}^{n-1} R_k \right), \quad (3.5)$$

para todo $n=1,2,3\dots N+1$, onde R é um resistor com valor arbitrário, $R_0 = 0$ e $S_{(n+1)} = 1$.

A Figura 3.3 mostra o diagrama de blocos do quantizador proposto por [11]. O quantizador é formado por um *ADC*, representado na figura 3.3-a, que pode ser implementado por uma rede escalada de tensão/corrente, resultando em um conjunto de $N - 1$ níveis de tensão que serão comparados com o sinal de entrada. Este processo resulta em um código termômetro que pode ser codificado. O *DAC*, figura 3.3-b, pode ser implementado também por uma rede escalada de tensão seguida de um decoder que converte o sinal binário em código *one-hot* [13].

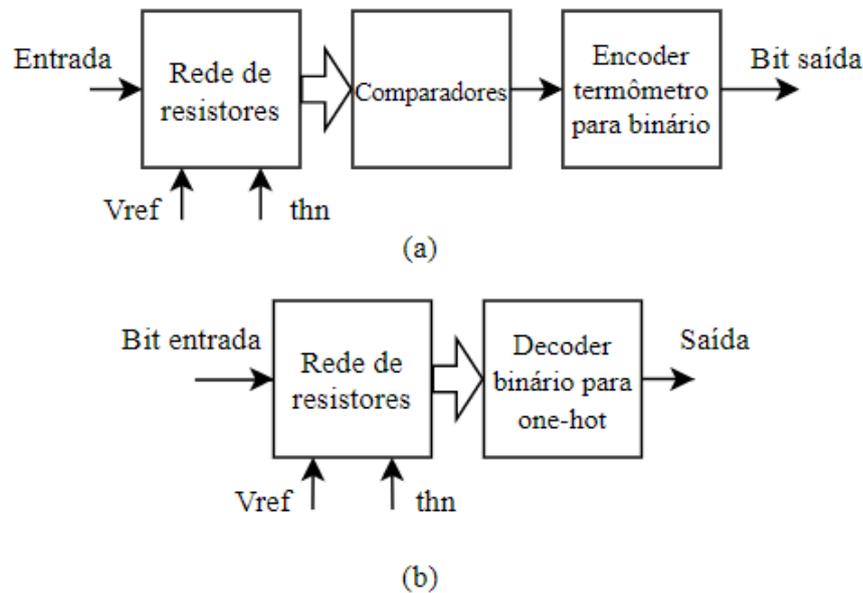


Figura 3.3 – Diagrama de blocos para implementação do quantizador arco-seno (Adaptada de [11]).

A Figura 3.4-a mostra a proposta do circuito elétrico do *ADC flash*. Neste caso, a rede resistiva implementa o dimensionamento da tensão. Os valores dos resistores são definidos pela equação 3.4.

A Figura 3.4-b mostra a proposta do circuito elétrico do *DAC*. Neste caso, a rede resistiva implementa a função de escala considerando os níveis de saída desejados. Para o circuito, os valores de resistores são definidos pela equação 3.5.

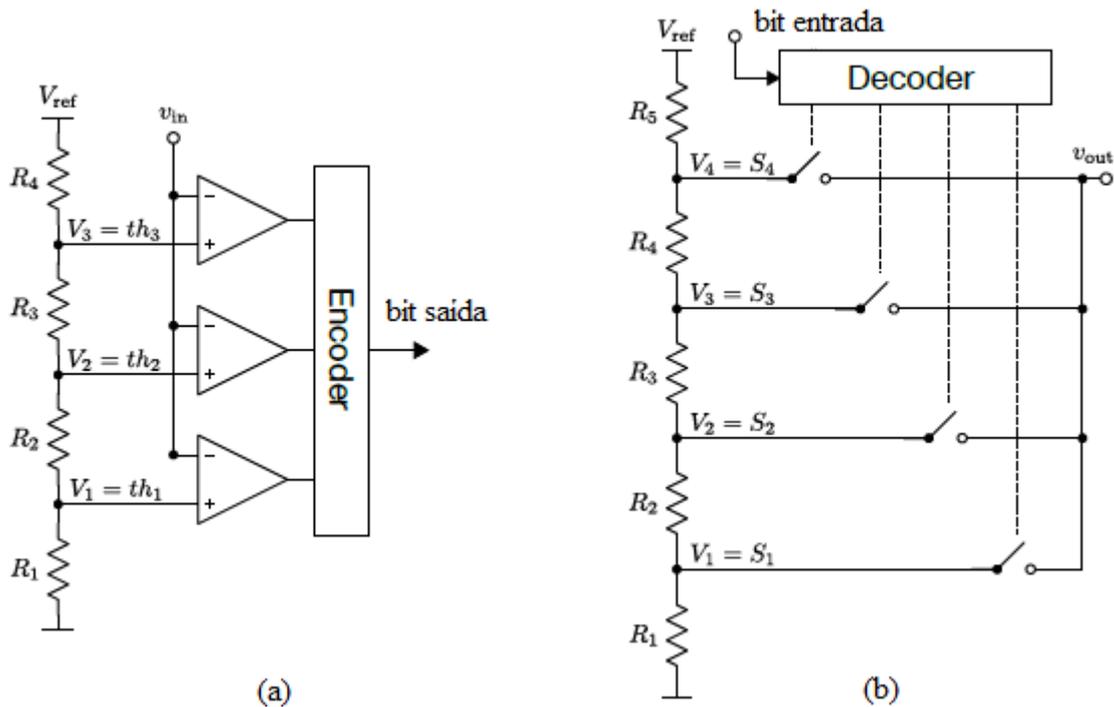


Figura 3.4 – Proposta de implementação utilizando topologia *flash*: a) ADC; b) DAC [11].

3.2 COMPANDING LOGARÍTMICO

Como mencionado na seção 1.1 do capítulo 1, existem alguns casos em que um ADC recebe uma determinada faixa de entrada analógica com mais frequência do que outras faixas. Sinais de áudio e voz, por exemplo, apresentam uma densidade mais alta de valores menores. Para esse tipo de sinal, muitas vezes é utilizado um processo de quantização adaptivo ou baseado na técnica de compressão e expansão ou companding [30]. A técnica de companding realiza compressão e depois a expansão do sinal e normalmente é implementada utilizando-se a lei A (A-Law) ou lei μ (μ -Law) para tornar sua distribuição mais adequada para um processo de quantização. A lei A é mais utilizada na Europa enquanto a lei μ é usada na América do Norte e no Japão e ambas usam tamanhos de etapas de quantização não uniformes que aumentam logaritmicamente com o nível do sinal [31]. A quantização adaptativa, por outro lado, requer informações sobre a distribuição de entrada e conseqüentemente adapta seu comportamento de quantização [30].

A Figura 3.5 mostra o princípio da técnica de companding. O diagrama de blocos da figura 3.5 é formado por um compressor, projetado para amplificar as amplitudes mais baixas do sinal às custas de atenuar as mais altas, seguido de um quantizador uniforme que irá realizar a quantização do sinal comprimido e por um expansor que irá executar as operações inversas a do compressor [12].

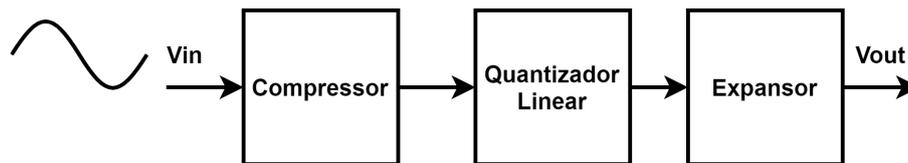


Figura 3.5 – Diagrama de blocos de um quantizador não uniforme utilizando a técnica de companding.

Em sistemas de telecomunicação, deseja-se uma SNR constante, independentemente da distribuição do sinal e, portanto, deseja-se encontrar a curva do compressor que pode atingir esse estado [12].

A lei de compressão logarítmica, expressa pela equação,

$$F(x) = 1 + k^{-1} \ln(x) \quad (3.6)$$

teoricamente pode atingir uma SNR constante independente da distribuição do sinal. Nesse tipo de quantização, o tamanho do passo em uma dada amplitude é proporcional a própria amplitude. Essa proporcionalidade, entretanto, não pode ser alcançada próxima a origem uma vez que a função logarítmica $F(x)$ é zero em $x = e^{-k}$, e não é definida no intervalo $0 \leq x < e^{-k}$, como mostra a figura 3.6. Uma lei de compressão/expansão prática não pode ser descontínua na origem, pois também deve especificar a função de transferência para sinais de nível baixo. Esse problema, no entanto, pode ser contornado realizando-se algumas alterações na lei logarítmica [12].

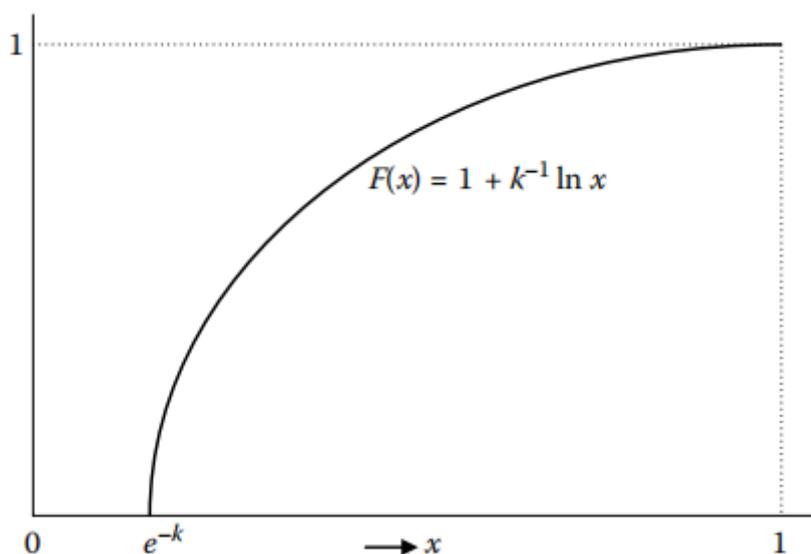


Figura 3.6 – Gráfico da função $F(x) = 1 + k^{-1} \ln(x)$ [12].

3.2.1 μ -Law

Tendo em vista o problema de descontinuidade na origem da lei de compressão logarítmica, pode-se utilizar a lei μ como uma aproximação. Essa aproximação desloca o cruzamento por zero que ocorre em $x = e^{-k}$ para a origem, relacionando a função $F(x)$ a $\ln(1 + \mu x)$ em vez de $\ln(x)$, de acordo com a equação,

$$F(x) = \frac{\log(1 + \mu x)}{\log(1 + \mu)}, \quad 0 \leq x \leq 1 \quad (3.7)$$

A Figura 3.7 mostra como o parâmetro μ controla a não linearidade da curva. A lei de compressão / expansão prática usada no método de modulação por código de pulso (Pulse Code Modulation - PCM) é uma aproximação segmentada da lei contínua para $\mu = 255$. Para $\mu = 0$, a conversão se torna uniforme (quantização linear). Quando $\mu = 1$, esta lei se aproxima da curva logarítmica para os níveis mais altos do sinal. Normalmente os sistemas de fala utilizam $\mu = 255$ pois esta característica é compatível com a audição humana [31]. Curvas de desempenho típicas para a lei μ , com parâmetro $\mu > 100$, mostram que a SNR é praticamente constante em uma faixa dinâmica de entrada razoavelmente grande. Além disso, essas curvas são bastante insensíveis à distribuição do sinal desde que os pontos adequados sejam escolhidos [12].

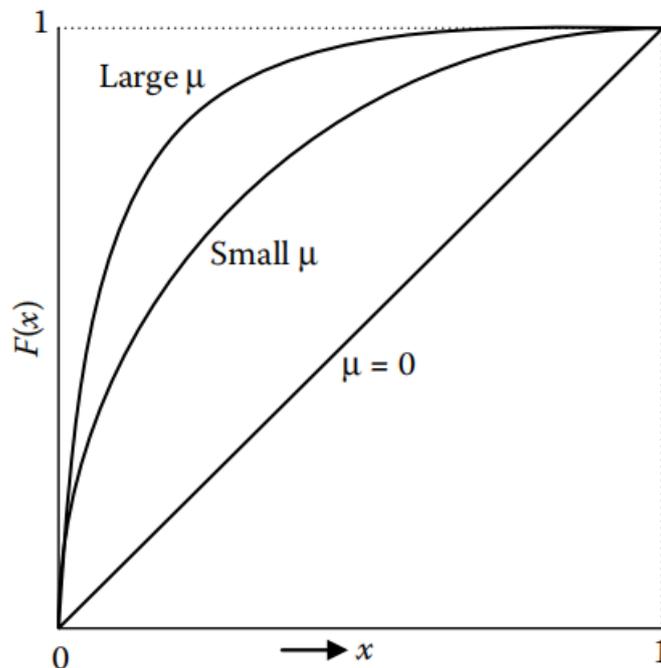


Figura 3.7 – Curvas para o compressor utilizando a lei u [12].

3.2.2 A-Law

Outra forma de se realizar a compressão logarítmica é utilizando-se a lei A (A -law). A curva do compressor, neste caso, é especificada para ser estritamente linear em níveis de sinal baixos e estritamente logarítmica em níveis de sinal altos. A extensão do segmento linear até a origem é obtida desenhando-se uma tangente à curva logarítmica $F(x) = 1 + k^{-1} \ln(x)$ como é mostrado na figura 3.8. Sendo m a inclinação da tangente e $x = x_t$ o ponto tangencial. Então, uma vez que $y = mx_t$ é um ponto na curva logarítmica, tem-se que,

$$mx_t = 1 + K^{-1} \ln x_t \quad (3.8)$$

Além disso, a relação da tangente implica que

$$m = \left. \frac{dF(x)}{dx} \right|_{x=x_t} = \frac{k^{-1}}{x_t} \quad (3.9)$$

E resolvendo-se as equações acima para m e x_t ,

$$x_t = e^{1-k} \quad (3.10)$$

$$m = k^{-1} e^{k-1} \quad (3.11)$$

A curva do compressor A-law em termos do parâmetro k pode, portanto, ser escrita como:

$$F(x) = \begin{cases} k^{-1} e^{1-k} x & 0 \leq x \leq e^{1-k} \\ 1 + k^{-1} \ln x & e^{k-1} \leq x \leq 1 \end{cases} \quad (3.12)$$

No entanto, esta lei é normalmente especificada em termos do parâmetro $A = e^{k-1}$, o que implica,

$$k = 1 + \ln A \quad (3.13)$$

Então, a metade positiva da curva do compressor é definida por

$$F(x) = \begin{cases} \frac{Ax}{1+\ln A} & 0 \leq x \leq 1/A \\ \frac{1+\ln Ax}{1+\ln A} & 1/A \leq x \leq 1 \end{cases} \quad (3.14)$$

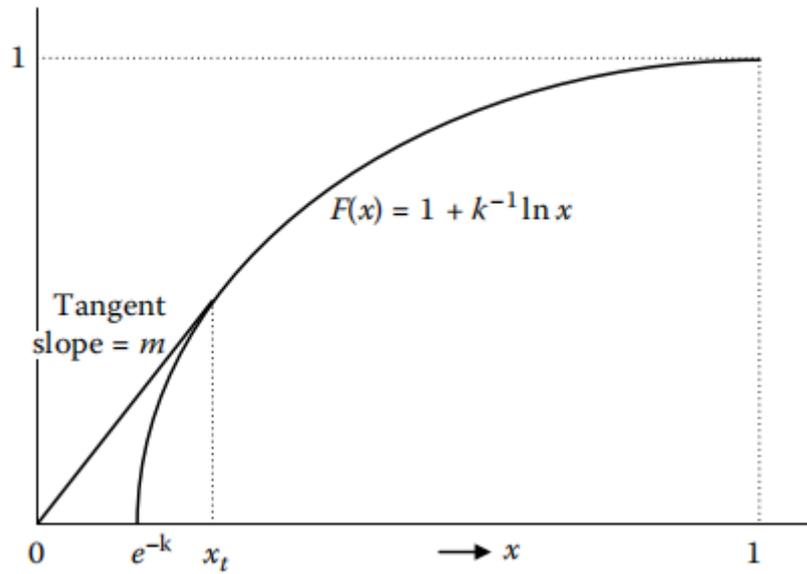


Figura 3.8 – Compressor baseado na lei A [12].

O parâmetro A controla a extensão do segmento linear na característica de compressão/expansão. Valores grandes de A implicam em um pequeno segmento linear e, portanto, em uma faixa de sinal maior na qual a SNR é constante. Para níveis de sinal pequenos, a lei A é linear e, portanto, se comporta como um quantizador uniforme, sendo assim, a SNR diminui linearmente com a potência do sinal [12].

4

PROJETO, IMPLEMENTAÇÃO E RESULTADOS

Essa capítulo apresenta a implementação, simulação e resultados dos quantizadores utilizando as técnicas descritas no Capítulo 3, incluindo o diagrama de blocos, a lógica e os testes de cada bloco implementado.

4.1 QUANTIZADOR LINEAR

Para o projeto dos quantizadores utilizou-se a topologia *flash*. Essa topologia foi escolhida pela facilidade de implementação, em um sistema de baixa resolução, e por ser uma arquitetura tipicamente utilizada no projeto dos quantizadores em moduladores Σ - Δ *Multi-bit* [32].

Para se ter um parâmetro de comparação para a análise de desempenho dos quantizadores não lineares, projetou-se também um quantizador linear. Esse quantizador foi projetado utilizando-se a topologia *flash*, mostrada na figura 3.4 do capítulo anterior e a linguagem de descrição de hardware Verilog-A.

Utilizando-se a arquitetura *flash*, para um *ADC* de N bits são necessários $2^N - 1$ comparadores e a mesma quantidade de níveis de referência. Neste projeto, onde utilizou-se um *ADC* de 4 bits, foram necessários 15 comparadores e 15 níveis de referência. Na saída dos comparadores, como mostra a figura 3.4, utilizou-se um circuito de codificação para recebe os sinais dos comparadores e codificar o sinal de saída em código binário. Para o projeto do *DAC*, utilizou-se o esquema da figura 3.4-b que conta com a rede de resistores, as chaves e um circuito decodificador.

Na sequência, cada um dos blocos pertencentes ao sistema serão apresentados a partir de sua descrição funcional e simulação. Os códigos em Verilog-A utilizados para a modelagem dos blocos podem ser encontrados no Apêndice A.

A Tabela 4.1 mostra os parâmetros que são comuns à maioria dos blocos modelados.

Tabela 4.1 – Parâmetros de simulação comuns aos blocos.

Parâmetro	Descrição	Valor
avdd	Tensão de referência positiva	1 V
avss	Tensão de referência negativa	-1 V
tf	Tempo de transição de subida e descida do modelo	1 fs
td	Tempo de atraso do modelo	1 fs

4.1.1 Comparador

Um comparador tem a função de comparar dois sinais de entrada. No caso dessa aplicação, utilizou-se um comparador síncrono que compara o sinal de entrada V_{in} com uma tensão de referência V_{ref} na borda de subida do $clock$. Se a tensão de entrada for maior que a tensão de referência, a saída do dispositivo ficará saturada e terá o valor de $avdd$, caso ocorra a situação inversa, a saída será $avss$.

A figura 4.1 mostra o símbolo do comparador implementado e o circuito utilizado para a sua simulação e a tabela 4.2 mostra a descrição dos pinos do bloco.

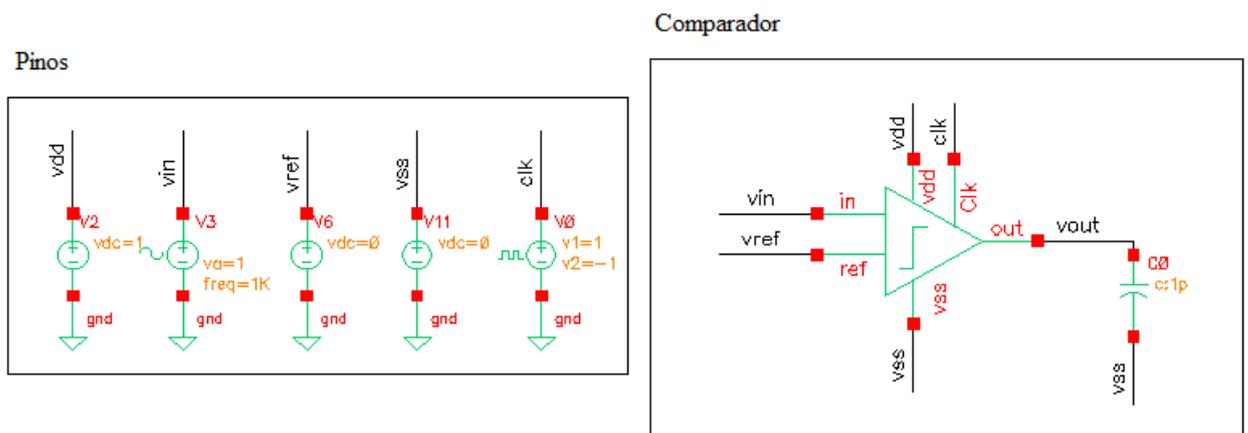


Figura 4.1 – Testbench do comparador [13].

Tabela 4.2 – Descrição dos pinos do comparador [13].

Pino	Descrição	Tipo
vin	Tensão do sinal de entrada	<i>input</i>
vref	Tensão de referência	<i>input</i>
vout	Tensão de saída	<i>output</i>
gnd	Ground (0V)	<i>inout</i>
vdd	Tensão de alimentação (1V)	<i>inout</i>
clk	Clock	<i>input</i>

A tabela 4.3 mostra os parâmetros utilizados na simulação e a figura 4.2 as formas de onda de entrada (azul) e saída (vermelho), resultantes da simulação do comparador.

Tabela 4.3 – Parâmetros de simulação do comparador [13].

Parâmetro	Descrição	Valor
frequência	Frequência de operação da fonte de teste	1KHz
ciclos	Quantidade de ciclos utilizados na simulação transiente	3

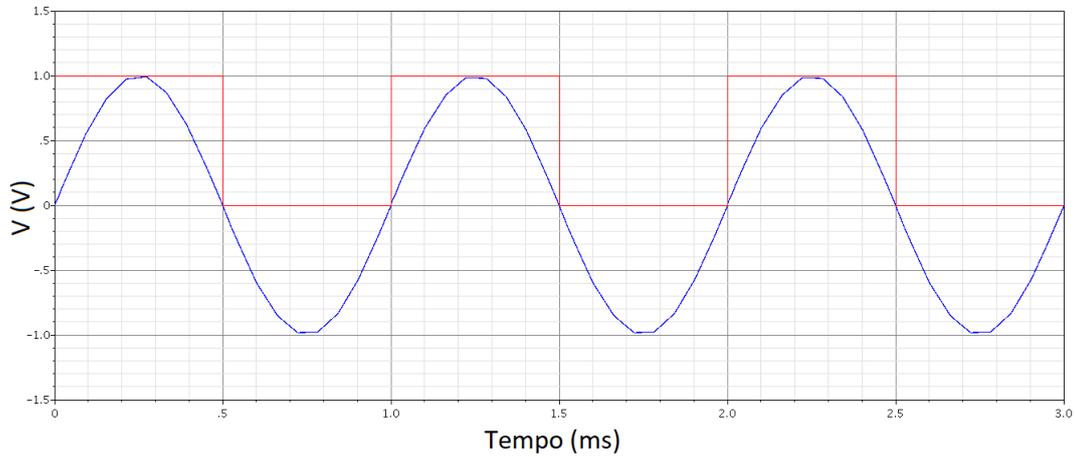


Figura 4.2 – Simulação do comparador [13].

4.1.2 Encoder

Um encoder ou codificador, é um circuito digital que converte um código ou número em código binário. As saídas dos comparadores, descritos na seção anterior, formam o chamado código termômetro. Neste tipo de código cada símbolo é uma sequência de 0s seguida por uma sequência de 1s e ocorre apenas uma transição por vez [33]. O encoder foi então projetado para converter o código termômetro em binário. A Tabela 4.4 mostra o exemplo de código termômetro e seu correspondente binário para um conversor de 4 bits.

Tabela 4.4 – Códigos de entrada (termômetro) e saída (binário) do encoder projetado [13].

Nível	Código Termômetro	Código Binário
0	000000000000000	0000
1	000000000000001	0001
2	000000000000011	0010
3	000000000000111	0011
4	000000000001111	0100
5	000000000011111	0101
6	000000000111111	0110
7	000000001111111	0111
8	000000011111111	1000
9	000000111111111	1001
10	000001111111111	1010
11	000011111111111	1011
12	000111111111111	1100
13	001111111111111	1101
14	011111111111111	1110
15	111111111111111	1111

A figura 4.3 mostra o símbolo do encoder implementado e o circuito utilizado na sua simulação e a tabela 4.5 mostra a descrição dos pinos do bloco.

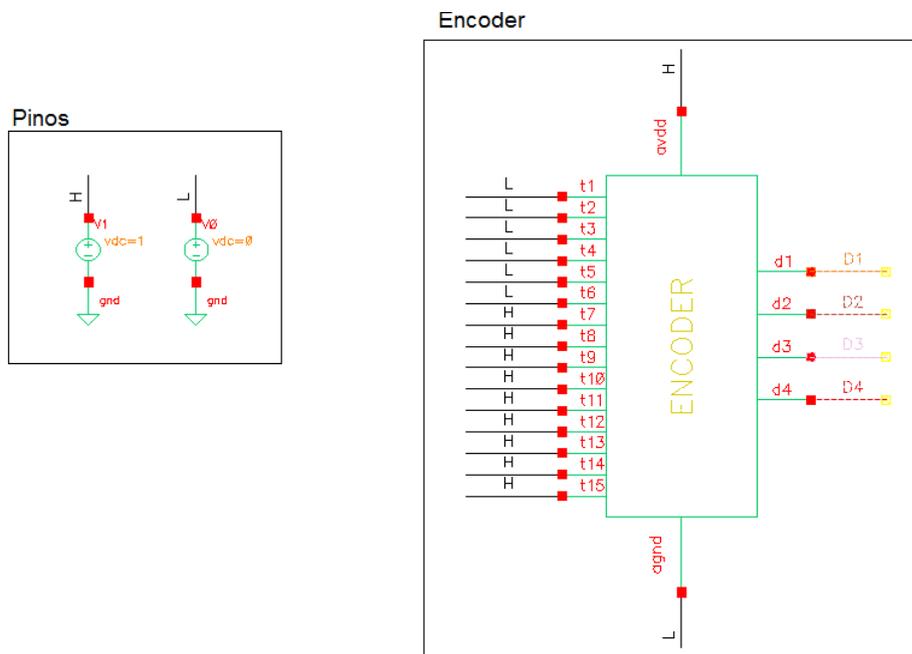


Figura 4.3 – Testbench do encoder [13].

Tabela 4.5 – Descrição dos pinos do Encoder [13].

Pino	Descrição	Tipo
$t1$ a $t15$	Bits de entrada que formam o código termômetro	<i>input</i>
$d1$ a $d4$	Bits de saída do código binário	<i>output</i>

Sendo $t1$ e $d1$ os bits mais significativos.

A tabela 4.6 mostra os parâmetros utilizados na simulação e a figura 4.4 mostra a saída do encoder para um sinal de entrada “000000111111111” que corresponde ao nível 9 da Tabela 4.4.

Tabela 4.6 – Parâmetros de simulação do encoder [13].

Parâmetro	Descrição	Valor
$t1$ a $t6$	Tensão aplicada a entrada do encoder	0 V
$t7$ a $t15$	Tensão aplicada a entrada do encoder	1V

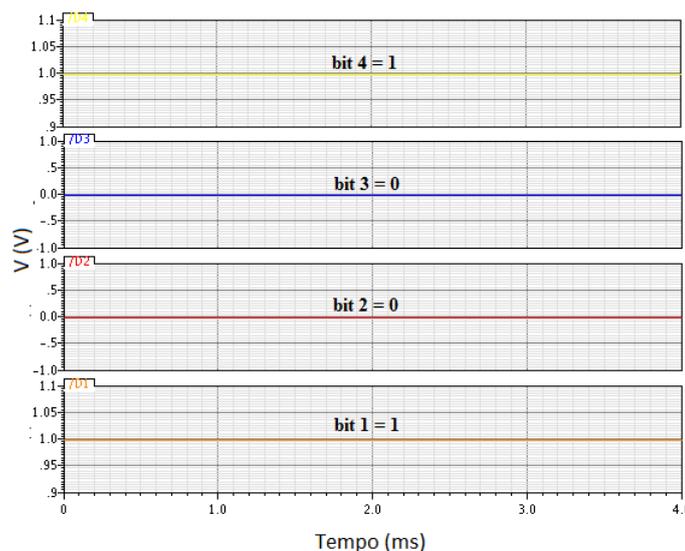


Figura 4.4 – Simulação do encoder para o código de entrada “000000111111111”[13].

4.1.3 ADC Flash

Para o projeto do *ADC*, conectou-se os comparadores modelados sendo que um dos pinos recebe o sinal de entrada e o outro a tensão de referência fornecida pela divisão da tensão nos resistores. O valor dos resistores escolhidos para essa topologia foi de $10K$ e por se tratar de um sistema linear, todos os valores de resistência são iguais. Na saída dos comparadores conectou-se o encoder descrito na seção anterior, completando-se assim o esquemático do *ADC Flash*.

A Figura 4.5 mostra parte do esquemático do *ADC Flash* de 4 bits projetado e a tabela 4.7 a descrição dos pinos do bloco.

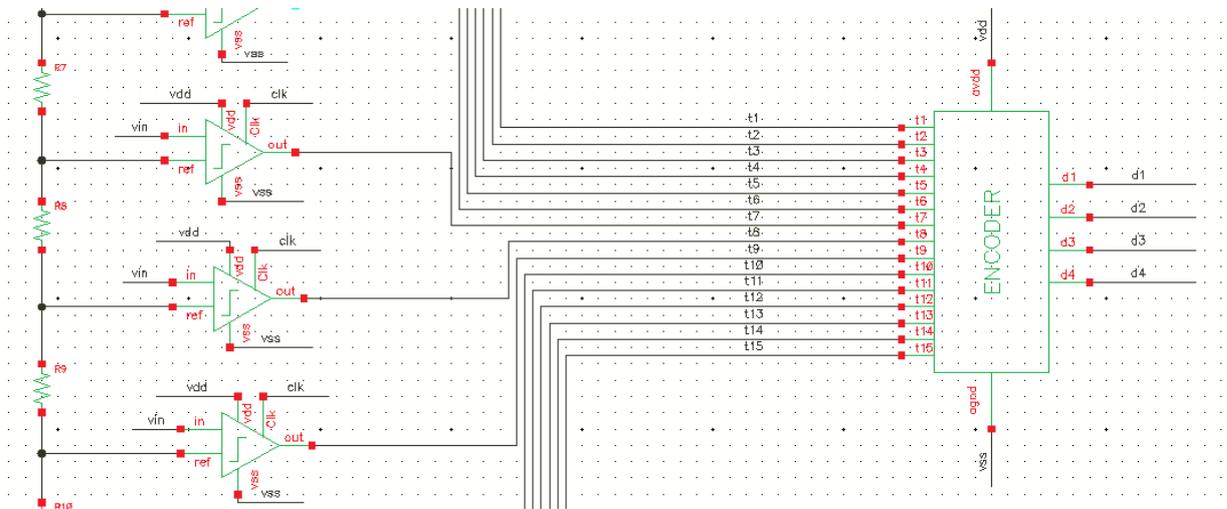


Figura 4.5 – Parte do esquemático do *ADC flash* de 4 bits [13].

Tabela 4.7 – Descrição dos pinos do *ADC* [13].

Pino	Descrição	Tipo
Vin	Sinal de entrada aplicado no conversor	<i>input</i>
d1 a d4	Bits de saída gerados pelo conversor	<i>output</i>

4.1.4 Decoder

Um decoder ou decodificador é um circuito digital que faz a conversão de um código binário para outro código. A saída do *ADC* projetado gera um código binário. Para converter esse código novamente para um sinal analógico é necessário um decodificador que converta o código binário em um código *One-hot*, liberando assim a tensão adequada para os níveis do quantizador. O código *One-hot* é formado por um grupo de bits entre os quais as combinações são com um único bit em nível alto (1) e todos os outros em nível baixo (0) [34]. Projetou-se então um decoder binário para *One-hot* de 4 bits cujos códigos de entrada e saída podem ser vistos na tabela 4.8.

Tabela 4.8 – Códigos utilizados no projeto do decoder [13].

Nível	Código Binário	Código One-hot
0	0000	0000000000000001
1	0001	0000000000000010
2	0010	0000000000000100
3	0011	0000000000001000
4	0100	0000000000010000
5	0101	0000000000100000
6	0110	0000000001000000
7	0111	0000000010000000
8	1000	0000000100000000
9	1001	0000001000000000
10	1010	0000010000000000
11	1011	0000100000000000
12	1100	0001000000000000
13	1101	0010000000000000
14	1110	0100000000000000
15	1111	1000000000000000

A figura 4.6 mostra o símbolo do decoder implementado e o circuito utilizado na sua simulação e a tabela 4.9 mostra a descrição dos pinos do bloco.

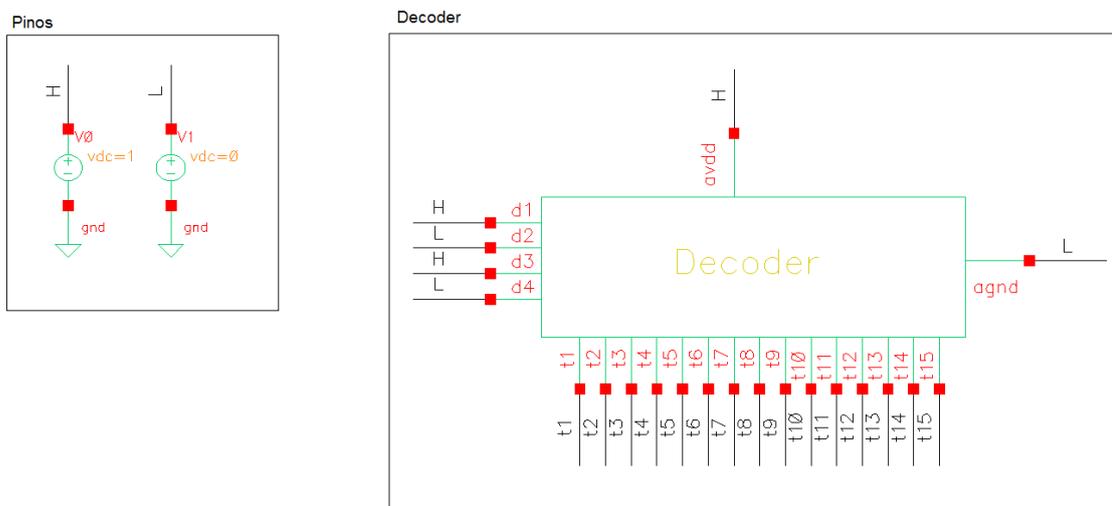


Figura 4.6 – Testbench do decoder [13].

Tabela 4.9 – Descrição dos pinos do decoder [13].

Pino	Descrição	Tipo
$d1$ a $d4$	Bits de entrada que formam o código binário	<i>input</i>
$t1$ a $t15$	Bits de saída do código One-hot	<i>output</i>

Para realizar a simulação do decoder aplicou-se um código “1010” na entrada, que corresponde ao nível 10 da tabela 4.8 e observou-se o código gerado na saída. A tabela 4.10 mostra os parâmetros utilizados na simulação e a figura 4.7 o código de saída.

Tabela 4.10 – Parâmetros de simulação do decoder [13].

Parâmetro	Descrição	Valor
$d2$ e $d4$	Tensão aplicada a entrada do decoder	0 V
$d1$ e $d3$	Tensão aplicada a entrada do decoder	1V

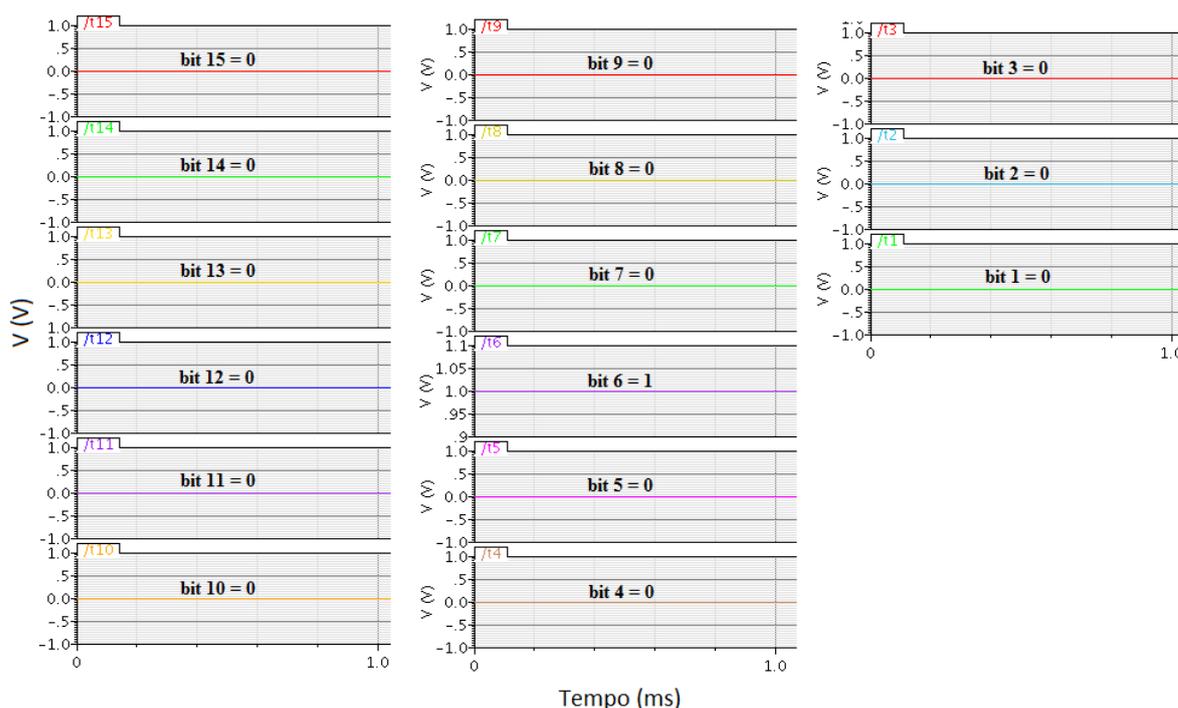


Figura 4.7 – Simulação do decoder para o código de entrada “1010”.

No projeto do *DAC* modelou-se também uma chave ideal para realizar a distribuição dos níveis de tensão vinda dos resistores. O teste do *DAC* foi feito conectando-se todos os blocos e fazendo a conversão do sinal de analógico para digital e novamente para analógico de forma a validar o processo de conversão.

4.1.5 Diagrama de Blocos do Quantizador Linear

Após a modelagem e teste de todos os blocos descritos nas seções anteriores, projetou-se então o quantizador linear, conectando-se o *ADC* ao *DAC* como mostra a figura 4.8.

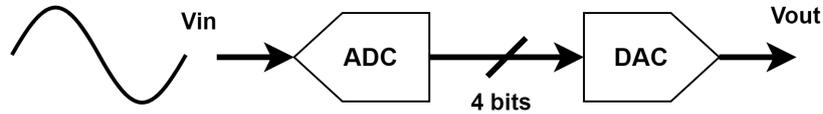


Figura 4.8 – Diagrama de blocos do quantizador linear.

Para o teste e simulação do quantizador aplicou-se dois sinais, uma senoide e uma rampa, na entrada do sistema com os parâmetros especificados nas Tabelas 4.11 e 4.12 e observou-se o sinal de saída.

Como pode-se observar na Figura 4.9 e na Figura 4.10, o quantizador linear gera passos de quantização constantes, o que dá a característica de linearidade ao sistema.

Tabela 4.11 – Parâmetros de simulação do quantizador com sinal de entrada senoidal [13].

Parâmetro	Descrição	Valor
frequência	Frequência de operação da fonte de teste senoidal	48.828 Hz
ciclos	Quantidade de ciclos utilizados na simulação	2

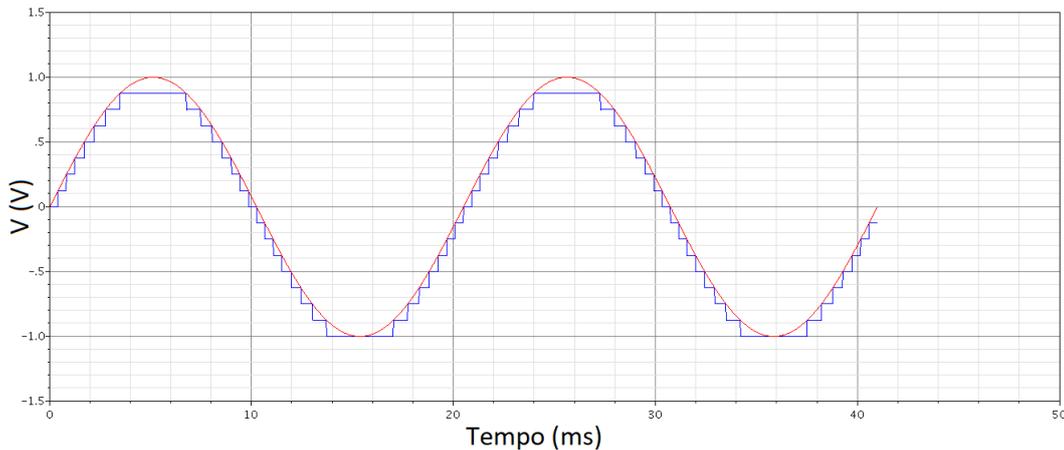


Figura 4.9 – Simulação do quantizador linear para um sinal senoidal [13].

Tabela 4.12 – Parâmetros de simulação do quantizador com sinal de entrada do tipo rampa [13].

Parâmetro	Descrição	Valor
Tensão 1	Primeira tensão do sinal	0
Tensão 2	Segunda tensão do sinal	1
t_r	tempo de subida do sinal	10 ms
t_f	tempo de descida do sinal	1 ms
ciclos	Quantidade de ciclos utilizados na simulação	1

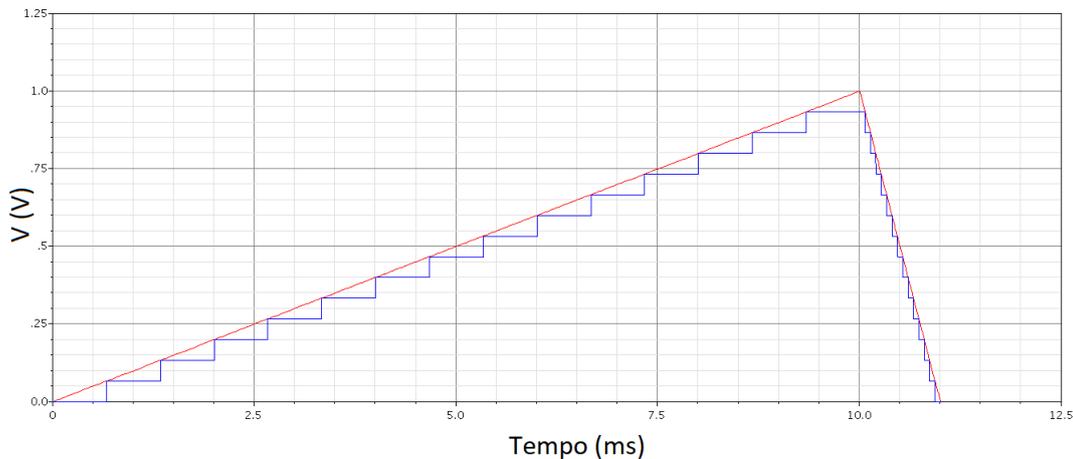


Figura 4.10 – Simulação do quantizador para um sinal do tipo rampa [13].

4.2 QUANTIZADOR ARCO-SENO

O desenvolvimento do modelo do quantizador Arco-seno foi o mesmo utilizado na aplicação do quantizador linear. Todos os blocos modelados na seções anteriores também foram utilizados nessa implementação e a única diferença entre os dois sistemas está nos valores das resistências utilizadas na rede de resistores do *ADC* e do *DAC*. No projeto do quantizador linear utilizou-se resistores iguais de $10K$ para o *ADC* e para o *DAC*. No caso do quantizador com distribuição arco-seno, esses valores foram calculados de acordo com as Equações 3.4 e 3.5 sugeridas por [11]. Tendo em vista essas equações e com o valor arbitrado de $R = 10K$ calculou-se os valores das resistências para o *ADC* e para o *DAC* mostrados na Tabela 4.13.

Tabela 4.13 – Resistências do *ADC* e do *DAC* para o quantizador Arco-seno [13].

RN	RADC(Ω)	RDAC(Ω)	RN	RADC(Ω)	RDAC(Ω)
R16	96	24	R8	975	980
R15	285	191	R7	938	961
R14	462	375	R6	865	906
R13	621	545	R5	758	815
R12	758	693	R4	621	693
R11	865	815	R3	462	545
R10	938	906	R2	285	375
R9	975	961	R1	96	191

Sendo R16 o resistor mais próximo a VCC e R1 o resistor mais próximo a 0V.

Os testes e simulações do quantizador não linear também foram feitos da mesma forma que a do quantizador linear, conectando-se o *ADC* ao *DAC*, aplicando-se sinais de entrada e observando-se as saídas. Os parâmetros utilizados na simulação também foram os mesmos da Tabela 4.11, para o sinal senoidal, e da Tabela 4.12 para o sinal rampa.

Nesse caso, pode-se observar nas figuras 4.12 e 4.11 que para ambas as simulações o quantizador Arco-seno tem passos de quantização menores e conseqüentemente, melhor resolução nas extremidades dos sinais.

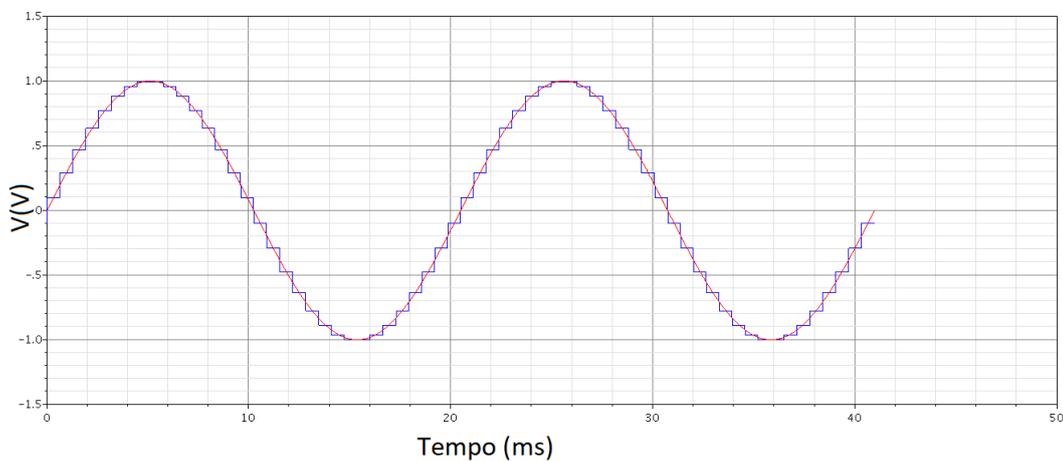


Figura 4.11 – Simulação do quantizador Arco-seno para um sinal senoidal[13].

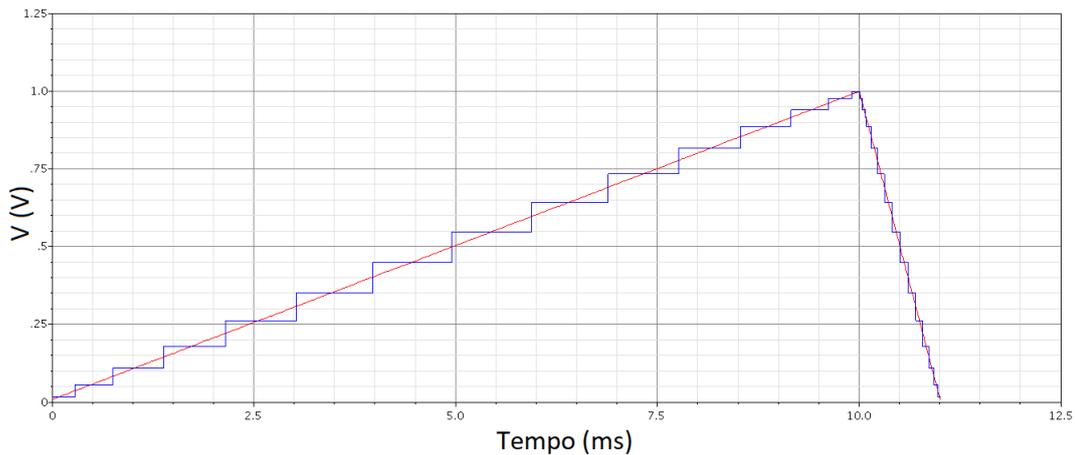


Figura 4.12 – Simulação do quantizador Arco-seno para um sinal do tipo rampa [13].

4.3 QUANTIZADORES NÃO LINEARES UTILIZANDO A LEI μ

Para o desenvolvimento do modelo do quantizador não linear utilizou-se a ferramenta Simulink do Matlab. No projeto do quantizador utilizou-se a mesma topologia *flash* dos quantizadores Linear e Arco-seno e além disso blocos adicionais de compressão e expansão. A Figura 4.13 mostra o diagrama de blocos desse quantizador, que será referenciado como quantizador μ . Como pode-se observar, foi utilizado o quantizador linear de 4-bits descrito anteriormente e além disso, um modelo de compressor e expansor baseados na lei μ com parâmetro $\mu = 255$.

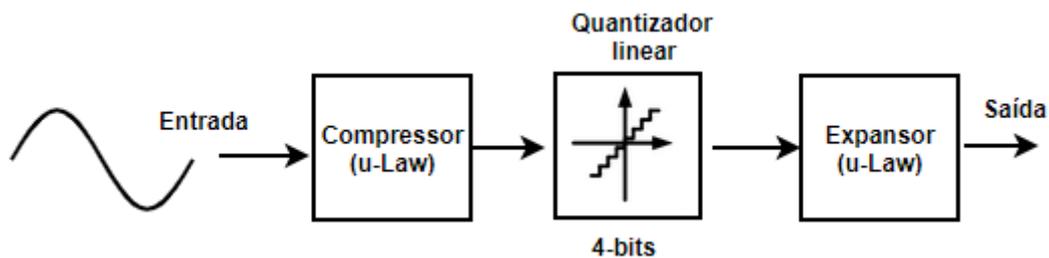


Figura 4.13 – Diagrama de blocos do quantizador não linear utilizando μ -law.

Para o teste e simulação do quantizador, aplicou-se dois sinais na entrada do sistema, uma senoide e um dente de serra, com parâmetros especificados nas Tabelas 4.14 e 4.15 e observou-se os sinais de saída.

Como pode-se observar na Figura 4.14 e na Figura 4.15, o quantizador μ gera passos de quantização maiores nas extremidades do sinal e gera passos de quantização menores, portanto com uma maior resolução, no centro da faixa dinâmica.

Tabela 4.14 – Parâmetros de simulação do quantizador com sinal de entrada senoidal.

Parâmetro	Descrição	Valor
frequência	Frequência de operação da fonte de teste senoidal	97.656 Hz
ciclos	Quantidade de ciclos utilizados na simulação	2

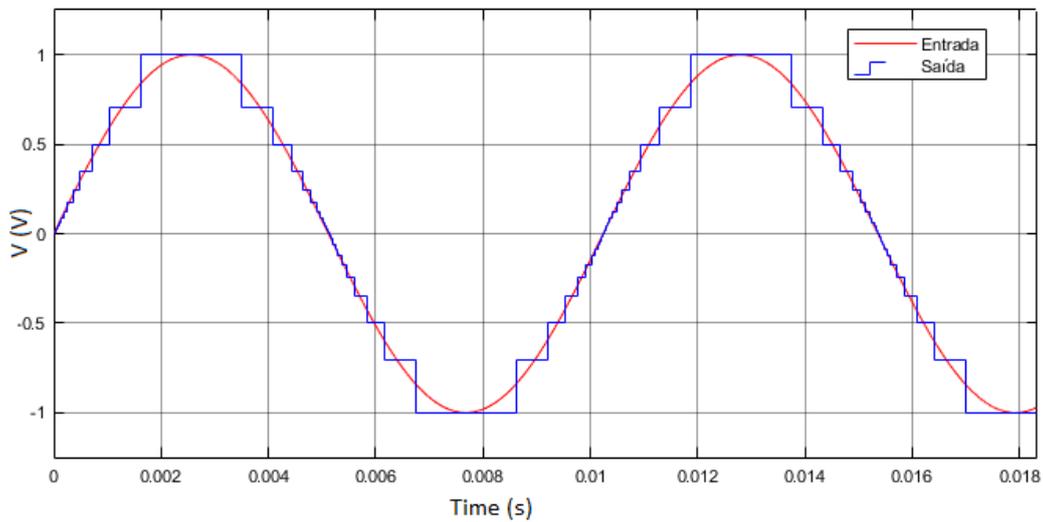


Figura 4.14 – Simulação do quantizador μ para um sinal de entrada senoidal.

Tabela 4.15 – Parâmetros de simulação do quantizador com sinal dente de serra.

Parâmetro	Descrição	Valor
Tensão 1	Primeiro valor de tensão do sinal	1 V
Tensão 2	Segundo valor de tensão do sinal	-1 V
Frequência	Frequência do sinal	97.656 Hz
Ciclos	Quantidade de ciclos da simulação	1

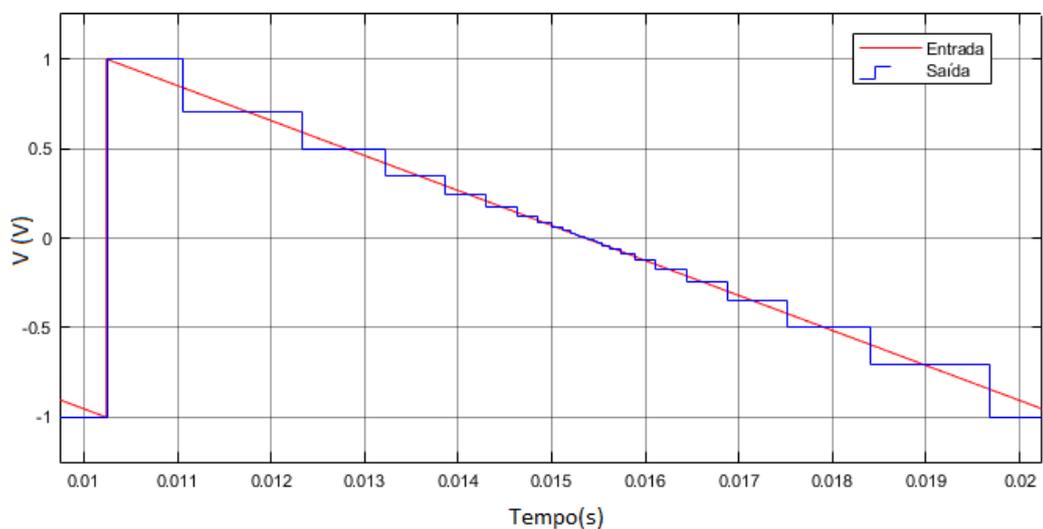


Figura 4.15 – Simulação do quantizador μ para um sinal dente de serra.

O efeito oposto pode ser obtido ao trocar a posição do compressor e do expansor na figura 4.13. As figuras 4.16 e 4.17 mostram os resultados de simulação utilizando os mesmos parâmetros apenas expandindo o sinal antes do quantizador e comprimindo depois, dessa forma, o sinal gera passos de quantização menores nas extremidades do sinal e portanto uma melhor resolução nesses pontos.

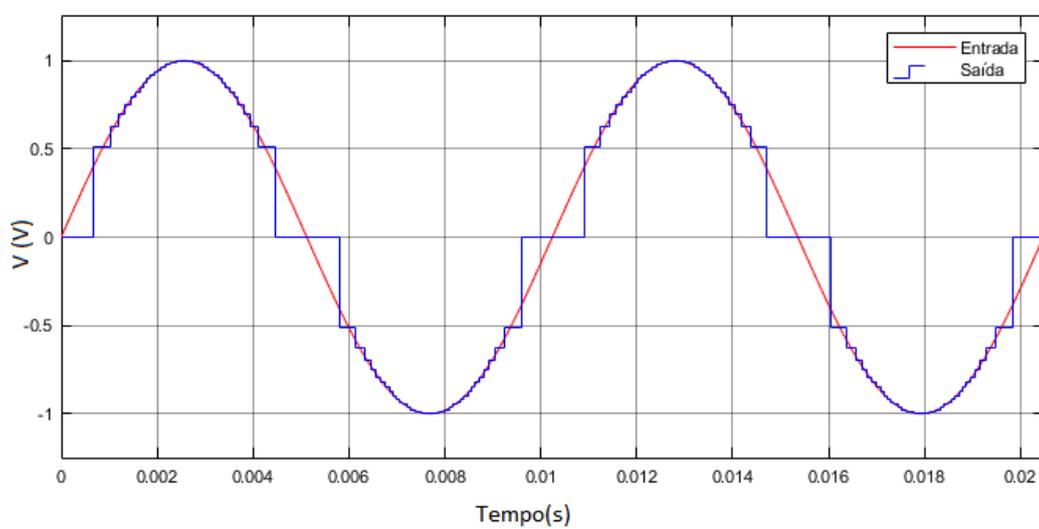


Figura 4.16 – Simulação do quantizador μ invertido para um sinal de entrada senoidal.

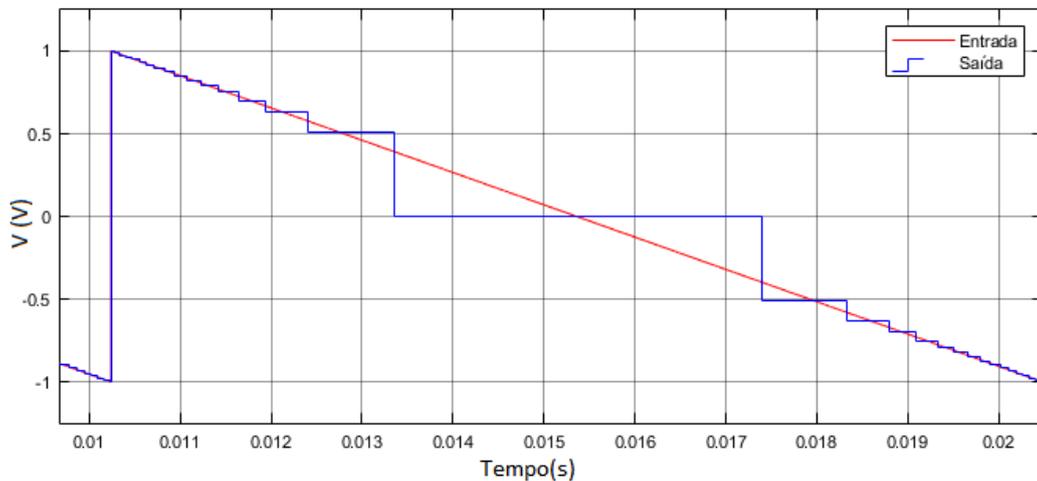


Figura 4.17 – Simulação do quantizador μ invertido para um sinal dente de serra.

Observando-se os dois quantizadores, que apresentam características opostas e maior vantagem para uma determinada faixa dinâmica, decidiu-se então juntar os dois sistemas, usando um ou outro, de acordo com a faixa dinâmica em que o sinal de entrada se encontra, e otimizando o processo de quantização.

A figura 4.18 mostra o diagrama de blocos do quantizador μ combinado, que será referenciado como quantizador μ -C.

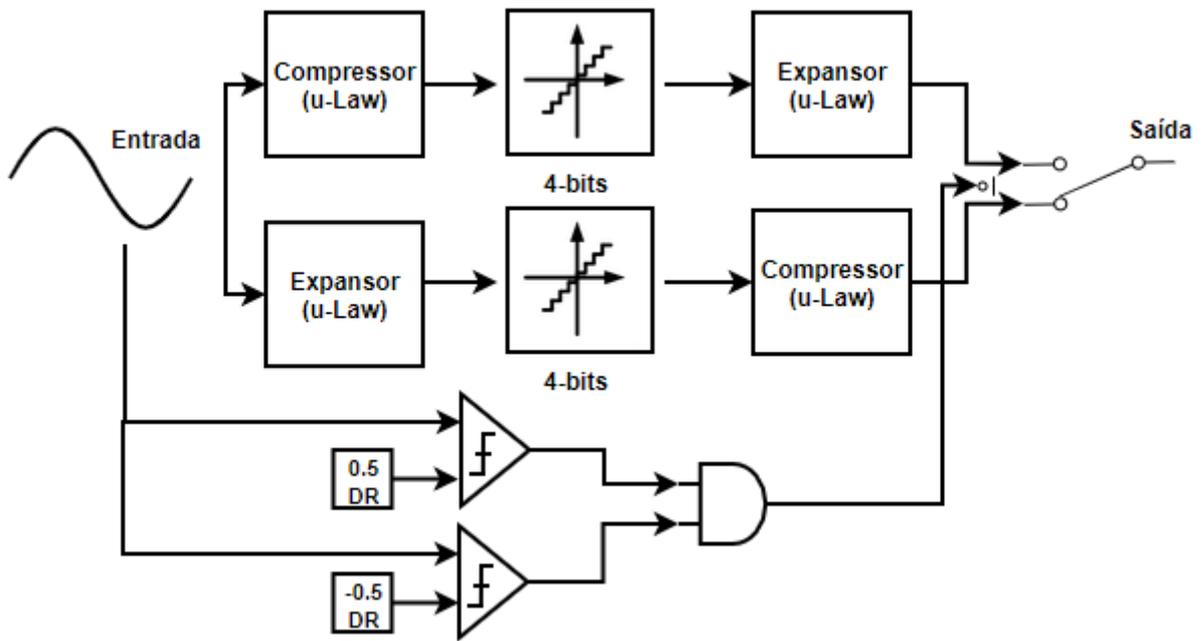


Figura 4.18 – Diagrama de blocos do quantizador μ -C.

No diagrama da figura 4.18 os quantizadores baseados na lei μ são conectados em paralelo e suas saídas são conectadas a uma chave que possui entrada de controle. Essa entrada é um sinal gerado por dois comparadores, que verificam se o sinal de entrada está localizado no meio da faixa dinâmica (entre 0.5 DR e -0.5 DR), e por uma porta lógica AND. Se o sinal estiver localizado no meio da faixa dinâmica (entre 0.5 DR e -0.5 DR), o quantizador μ será utilizado, caso contrário, o quantizador μ com expansor e compressor invertidos será utilizado. Dessa forma, ao associar os dois quantizadores, o sinal de saída sempre terá uma melhor resolução independente da sua faixa dinâmica.

As figuras 4.19 e 4.20 mostram os sinais de entrada e saída utilizando o quantizador μ -C.

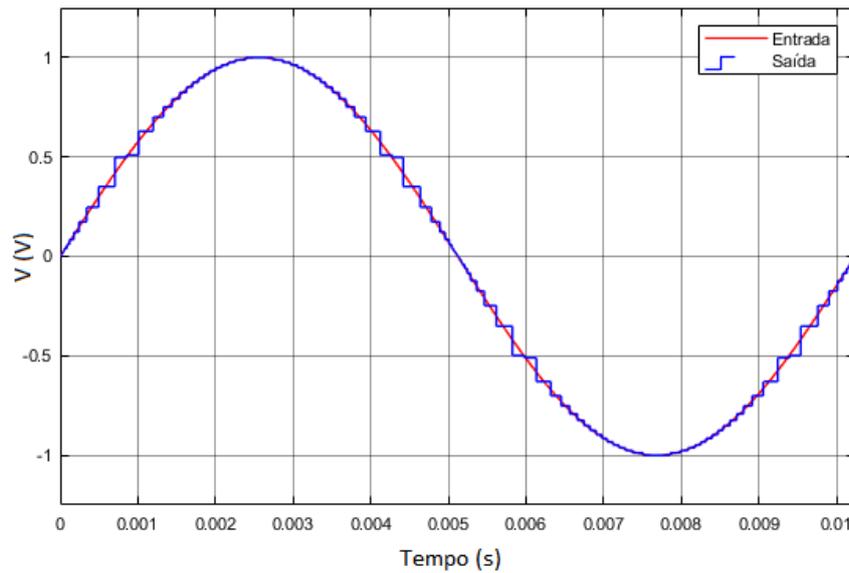


Figura 4.19 – Simulação do quantizador μ -C para um sinal de entrada senoidal.

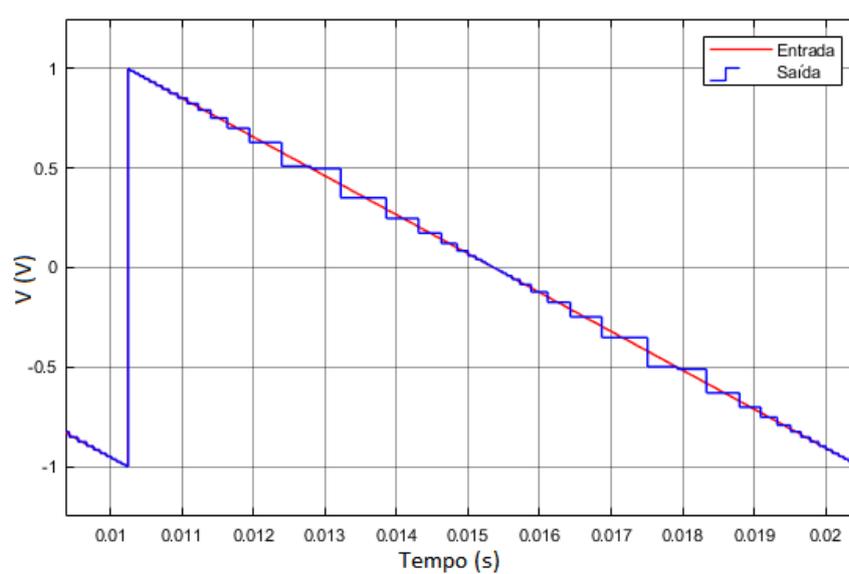


Figura 4.20 – Simulação do quantizador μ -C para um sinal dente de serra.

4.4 QUANTIZADORES NÃO LINEARES UTILIZANDO A LEI A

O desenvolvimento do modelo do quantizador não linear utilizando a lei A foi semelhante ao do quantizador utilizando lei μ . A Figura 4.21 mostra o diagrama de blocos desse quantizador, que será referenciado como quantizador A. A diferença entre os dois sistemas está na utilização do compressor e expansor baseados na Lei A que utilizam o parâmetro $A = 87.56$.

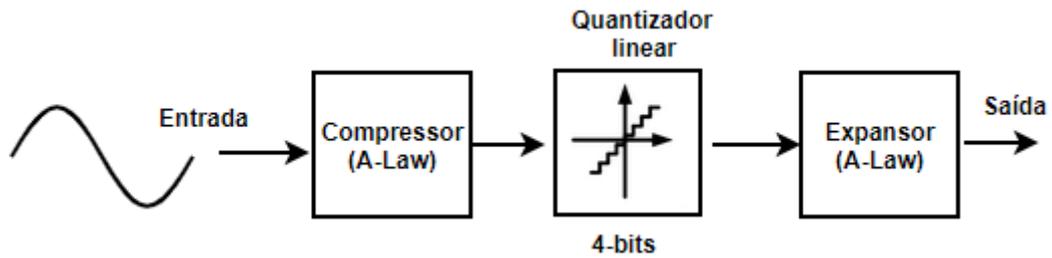


Figura 4.21 – Diagrama de blocos do quantizador não linear utilizando *A-law*.

Para o teste e simulação do quantizador também aplicou-se os sinais de entrada senoidal e dente de serra, com parâmetros especificados nas Tabelas 4.14 e 4.15 e observou-se os sinais de saída.

As Figuras 4.22 e 4.23 mostram os sinais de entrada e saída do quantizador A e assim como no quantizador μ , pode-se observar que o sistema gera passos de quantização maiores nas extremidades do sinal e passos de quantização menores, portanto com uma maior resolução, no centro da faixa dinâmica.

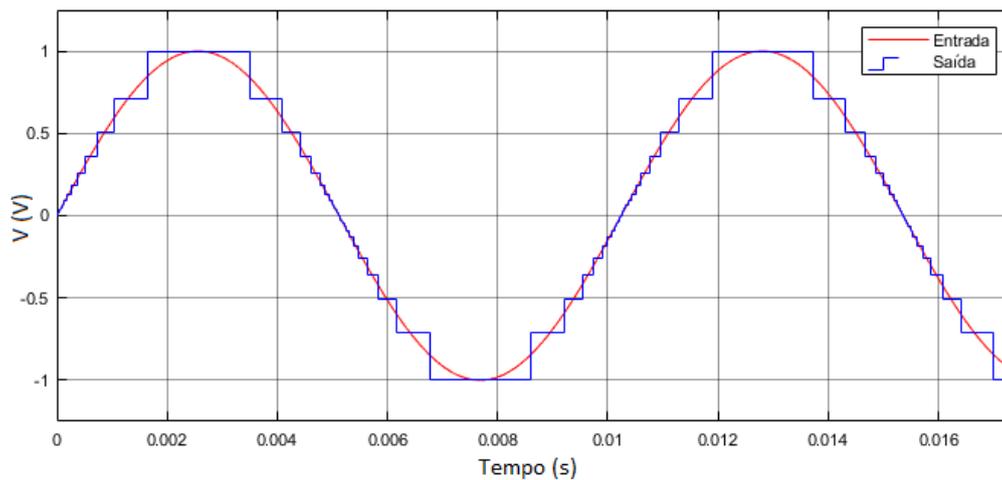


Figura 4.22 – Simulação do quantizador A para um sinal senoidal.

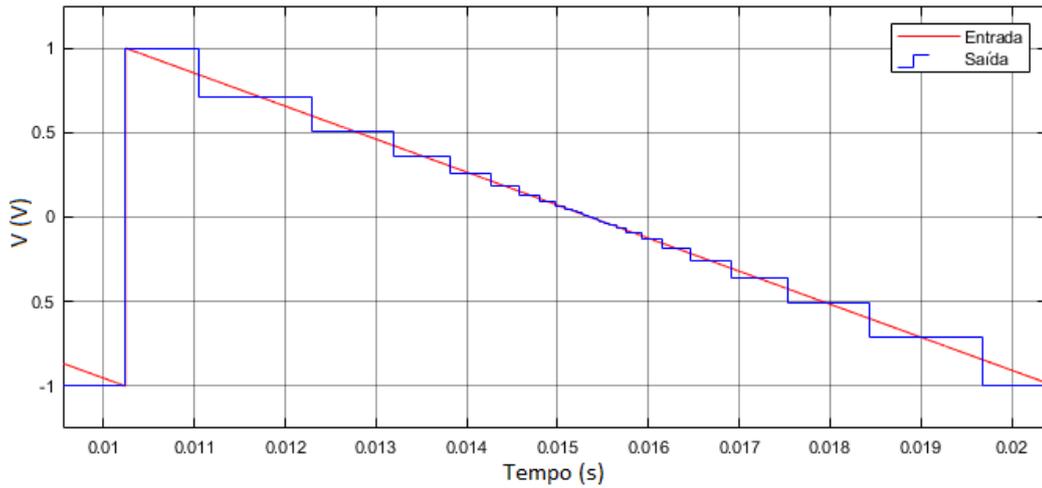


Figura 4.23 – Simulação do quantizador A para um sinal dente de serra.

Como no caso do quantizador μ , também pode-se obter o efeito oposto ao trocar a posição do compressor e expensor no diagrama da figura 4.21. As figuras 4.24 e 4.25 mostram os resultados de simulação para o quantizador A invertido.

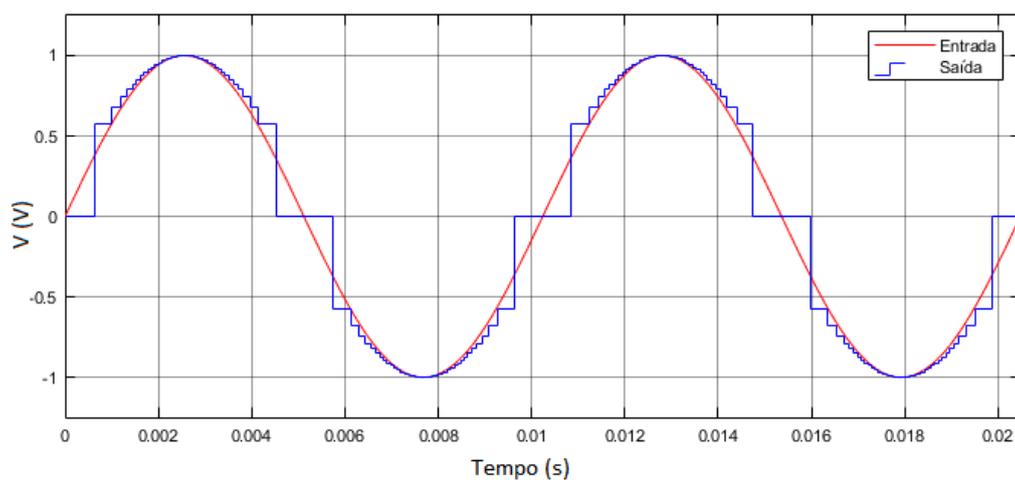


Figura 4.24 – Simulação do quantizador A invertido para um sinal senoidal.

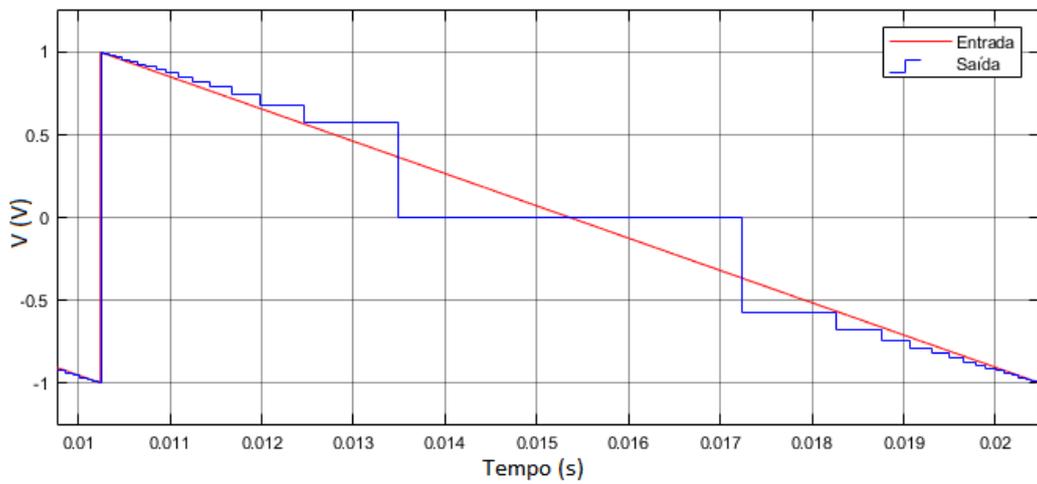


Figura 4.25 – Simulação do quantizador A invertido para um sinal dente de serra.

Assim como no caso da seção anterior, juntou-se os dois sistemas baseados na lei A de forma a se obter uma melhor resolução para toda a faixa dinâmica. Para isso utilizou-se o mesmo circuito da Figura 4.18 apenas trocando os expansores e compressores pelos que utilizam a Lei A ao invés da Lei μ .

As figuras 4.26 e 4.27 mostram os sinais de entrada e saída para esse sistema que será referenciado como quantizador A-C.

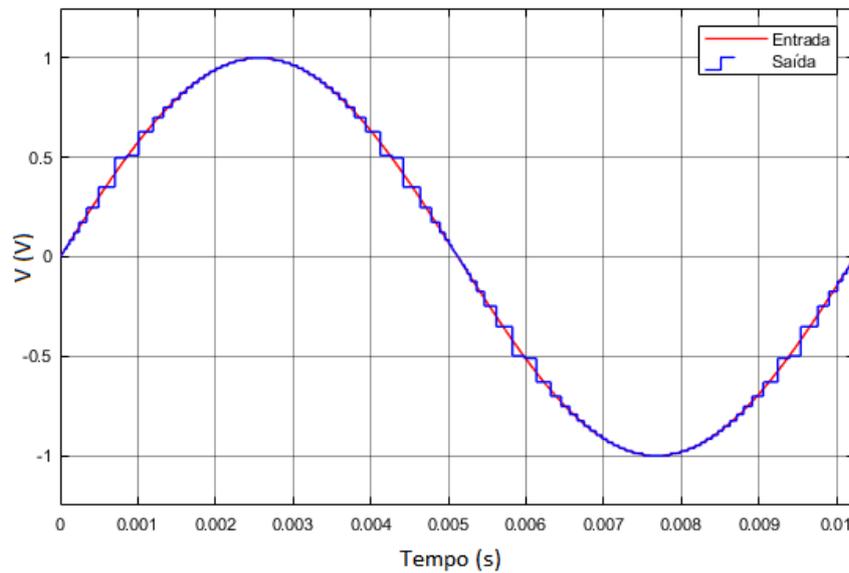


Figura 4.26 – Simulação do quantizador A-C para um sinal de entrada senoidal.

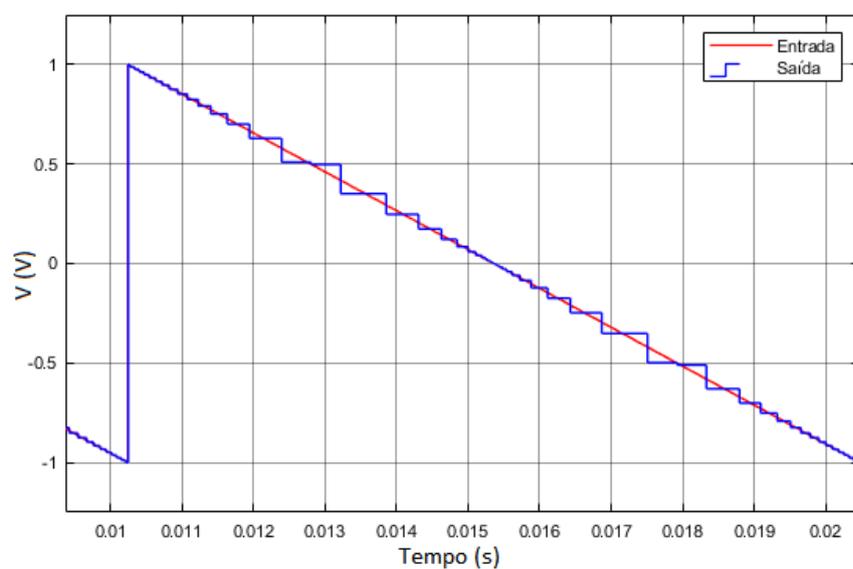


Figura 4.27 – Simulação do quantizador A-C para um sinal dente de serra.

4.5 COMPARAÇÃO ENTRE QUANTIZADORES

Essa Seção apresenta uma análise mais detalhada sobre os resultados obtidos na implementação dos quantizadores e uma comparação entre os sistemas. Para isso, foram realizadas análises no tempo e na frequência para cada um dos quantizadores.

A Figura 4.28 mostra os sinais de entrada e saída do quantizador linear e seu respectivo erro de quantização. Como esperado para o quantizador linear, pode-se observar em que os passos de quantização são constantes e o seu erro de quantização é dado pelo padrão clássico sino / dente de serra.

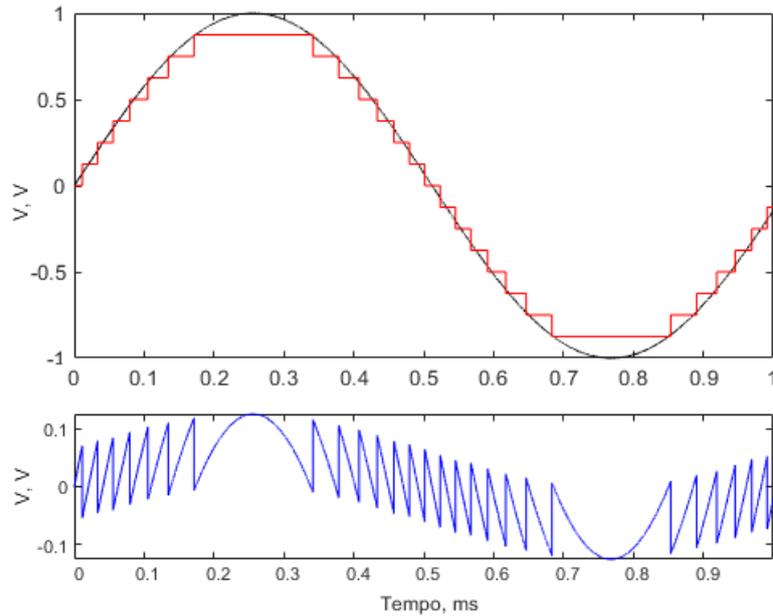


Figura 4.28 – Análise transiente para um sinal senoidal processado pelo quantizador linear e seu respectivo erro de quantização.

A figura 4.29 mostra os sinais de entrada, saída e o erro de quantização para o quantizador arco-seno. Nesse caso, pode-se observar que os passos de quantização não são constantes, eles são maiores no centro da faixa dinâmica e menores nas extremidades do sinal, sendo assim, o sinal apresenta uma melhor resolução nesses pontos. O erro de quantização apresenta um comportamento modular e também é menor nas extremidades do sinal.

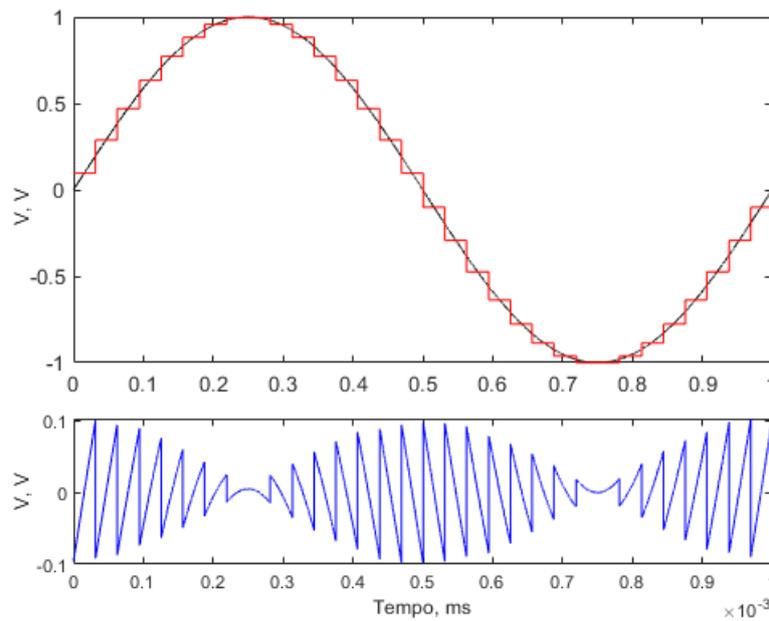


Figura 4.29 – Análise transiente para um sinal senoidal processado pelo quantizador arco-seno e seu respectivo erro de quantização.

A figura 4.30 mostra os sinais de entrada, saída e o erro de quantização para o quantizador μ . Assim como no caso do quantizador arco-seno, o quantizador μ é um quantizador não linear e por isso não apresenta passos de quantização constantes. No entanto, ao contrário do quantizador arco-seno, os passos de quantização são menores no centro da faixa dinâmica e maiores nas extremidades do sinal, sendo assim, esse quantizador apresenta uma maior resolução no centro da faixa dinâmica e conseqüentemente menor erro de quantização nessa faixa.

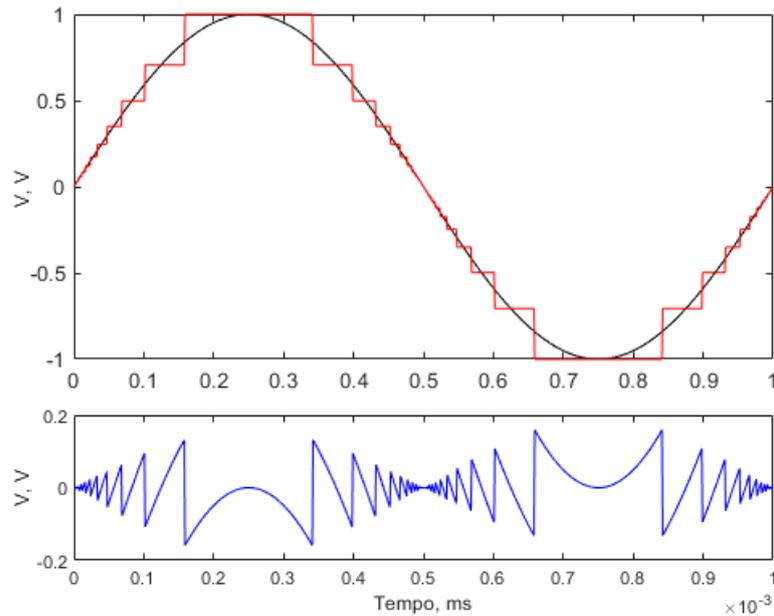


Figura 4.30 – Análise transiente para um sinal senoidal processado pelo quantizador μ e seu respectivo erro de quantização.

A figura 4.31 mostra os sinais de entrada, saída e o erro de quantização para o quantizador A. Assim como no caso do quantizador μ , o quantizador A também apresenta o comportamento não linear com maior resolução e menor erro de quantização do centro da faixa dinâmica. Comparando as Figuras 4.30 e 4.31 pode-se observar que o comportamento dos quantizadores μ e A é bastante similar. Fazendo a sobreposição dos sinais, no entanto, pode-se observar a diferença entre os dois sistemas como mostra a Figura 4.32.

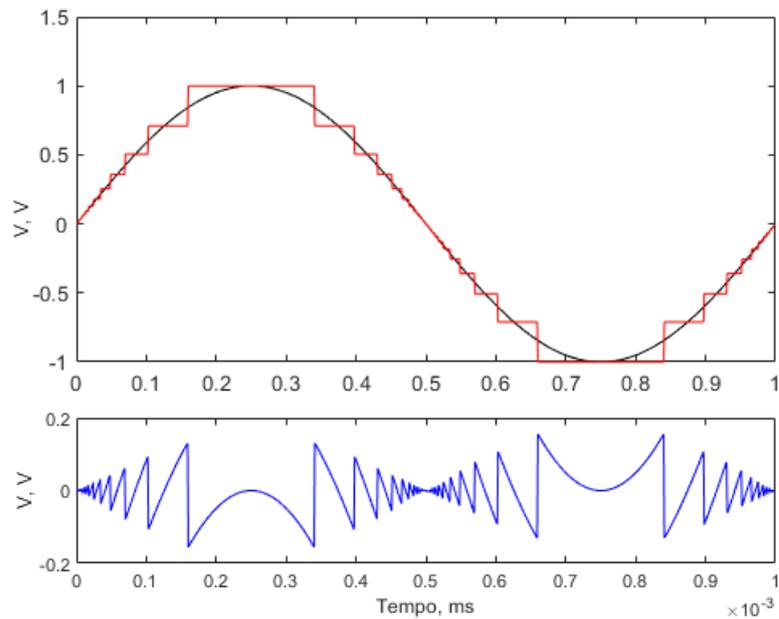


Figura 4.31 – Análise transiente para um sinal senoidal processado pelo quantizador A e seu respectivo erro de quantização.

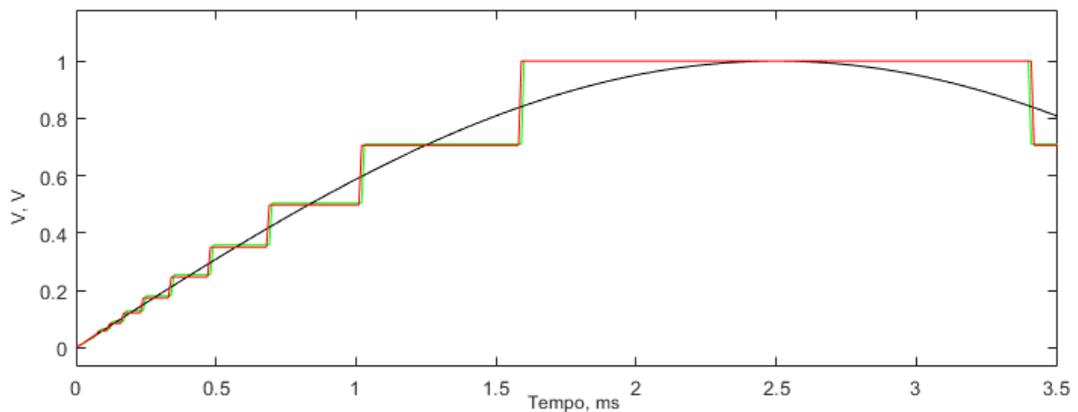


Figura 4.32 – Diferença entre sinais processados pelo quantizador μ (verde) e pelo quantizador A (vermelho).

A Figura 4.33 mostra os sinais de entrada, saída e o erro de quantização para o quantizador μ invertido. Ao contrário do quantizador μ , esse sistema apresenta uma resolução maior no centro da faixa dinâmica e portanto menor erro de quantização nessa faixa. O quantizador A invertido também apresentou o resultado semelhante apenas com um pequeno deslocamento do sinal no tempo.

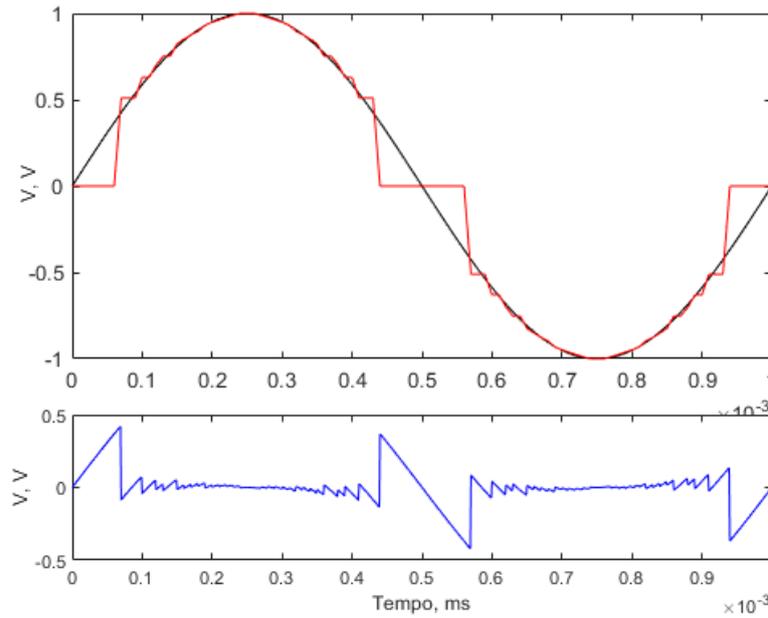


Figura 4.33 – Análise transiente para um sinal senoidal processado pelo quantizador μ invertido e seu respectivo erro de quantização.

A Figura 4.34 mostra os sinais para o quantizador μ -C. Como pode-se observar nos resultados dessa simulação, a combinação dos dois quantizadores logarítmicos aumenta a resolução do sinal de saída e diminui o erro de quantização de forma significativa em comparação com os outros sistemas apresentados para quase toda a faixa dinâmica do sinal.

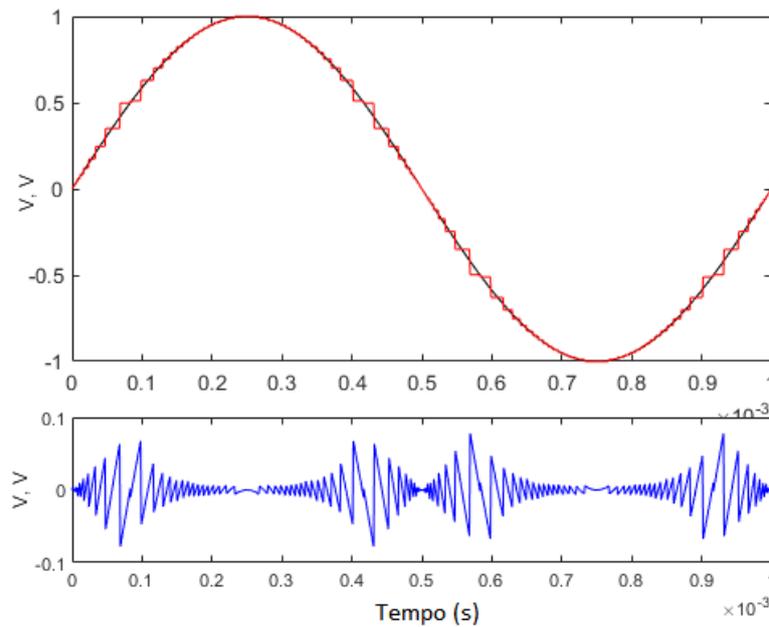


Figura 4.34 – Análise transiente para um sinal senoidal processado pelo quantizador μ -C e seu respectivo erro de quantização.

A comparação entre todos os quantizadores projetados pode ser melhor visualizada no Anexo B.1.

Também realizou-se a análise em frequência para os quantizadores Linear, Arco-seno, μ -C e A-C. A Figura 4.35 mostra as *DFTs* do quantizador Linear (4.35-a) e Arco-seno (4.35-b). A Figura 4.36 mostra as *DFTs* do quantizador μ -C (4.36-a) e do quantizador A-C (4.36-b).

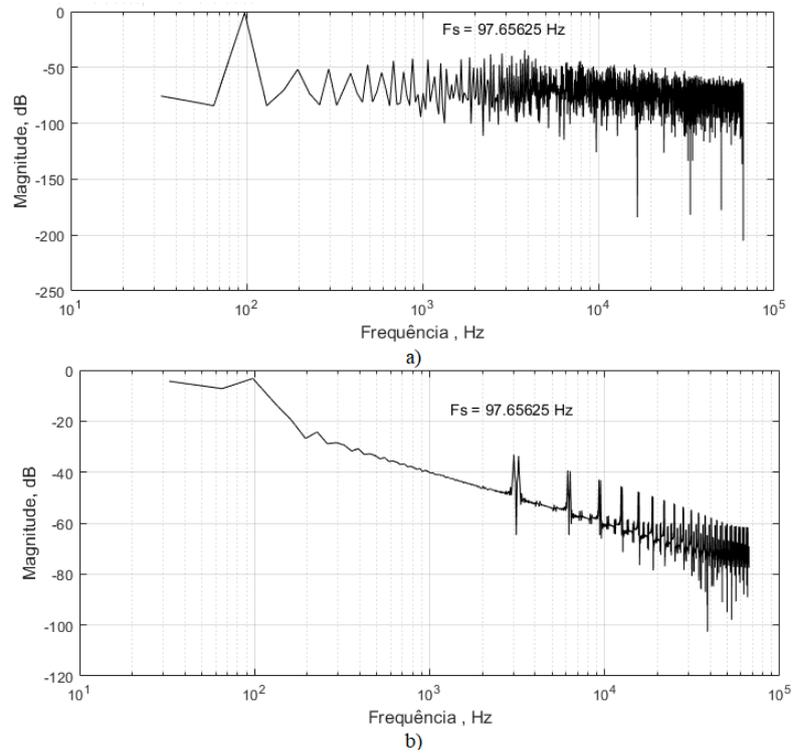


Figura 4.35 – DFTs dos quantizadores Linear (a) e Arco-seno (b).

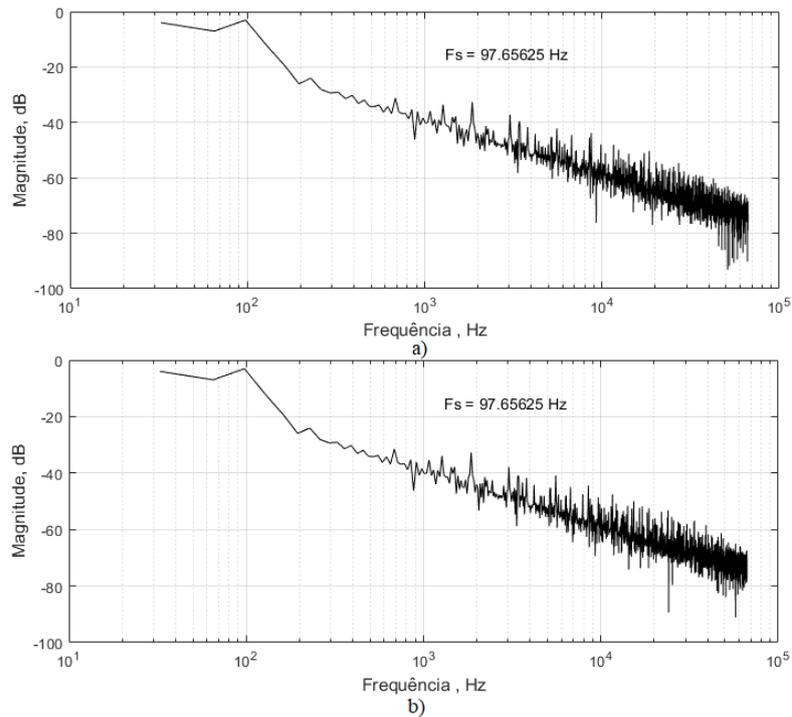


Figura 4.36 – DFTs dos quantizadores μ -C (a) e A-C (b).

4.6 MODULADOR Σ - Δ *MULTI-BIT*

Essa seção apresenta a modelagem e simulações em alto nível de um modulador Σ - Δ *Multi-bit* de primeira e de segunda ordem. O objetivo dessa implementação é a aplicação dos quantizadores projetados nos capítulos anteriores em um sistema real e conhecido. Dessa forma será possível obter uma estimativa sobre o desempenho alcançável dos mesmos e também sobre os requisitos de *design* das arquiteturas caso seja realizado o seu projeto a nível de transistor. Assim como no caso dos quantizadores, as simulações foram realizadas com as ferramentas da *Cadence Design Systems* e os blocos modelados utilizando-se a *HDL Verilog-A*. Na sequência, serão apresentados os blocos pertencentes ao sistema a partir de sua descrição funcional, tabela de pinos e simulações.

4.6.1 Integrador

Um integrador é um circuito que realiza o processo de integração em um determinado sinal de entrada. No sistema de um modulador Σ - Δ , o integrador integra o erro entre o sinal amostrado e sinal de entrada por meio da malha de realimentação.

A Figura 4.37 mostra o símbolo do integrador implementado e o circuito utilizado para a sua simulação e a tabela 4.16 mostra a descrição dos pinos do bloco.

Para a simulação do integrador aplicou-se um sinal quadrado na entrada, com os parâmetros especificados na Tabela 4.22, e analisou-se o sinal de saída representado na figura 4.38.

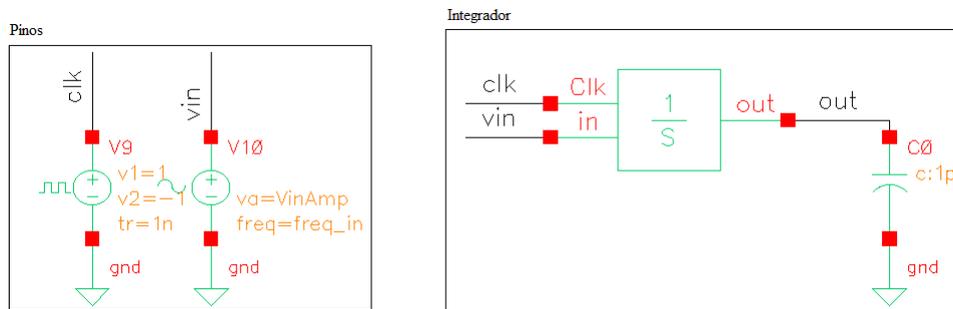


Figura 4.37 – Testbench do integrador [13].

Tabela 4.16 – Descrição dos pinos do integrador [13].

Pino	Descrição	Tipo
Clk	Clock	<i>input</i>
in	Tensão do sinal de entrada	<i>input</i>
out	Tensão do sinal de saída	<i>output</i>

A Tabela 4.22 mostra os parâmetros utilizados na simulação e a figura 4.38 as formas de onda de entrada (azul) e saída (vermelho), resultantes da simulação do integrador.

Tabela 4.17 – Parâmetros de simulação do integrador [13].

Parâmetro	Descrição	Valor
Frequência	frequência da fonte de teste	1K
Amplitude	Amplitude do sinal de teste	1V
ciclos	Quantidade de ciclos na análise transiente	3

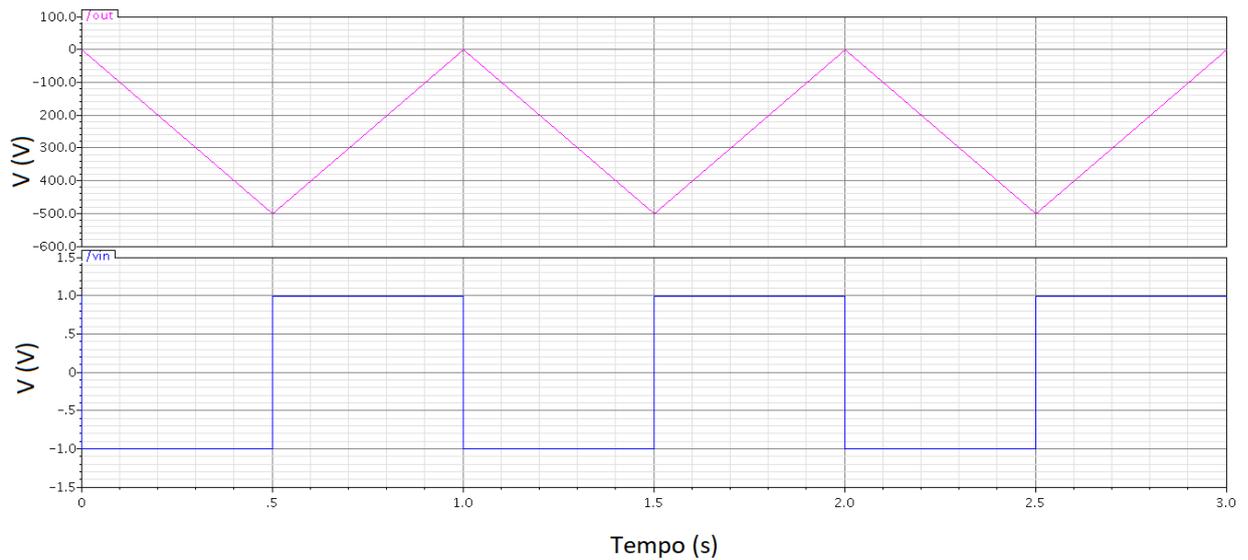


Figura 4.38 – Simulação do integrador [13].

4.6.2 Subtrator

O subtrator é um circuito que realiza a subtração entre dois sinais, nesse caso, o sinal de entrada e de saída do modulador Σ - Δ . A Figura 4.39 mostra o símbolo do integrador implementado e o circuito utilizado para a sua simulação e a tabela 4.18 mostra a descrição dos pinos do bloco.

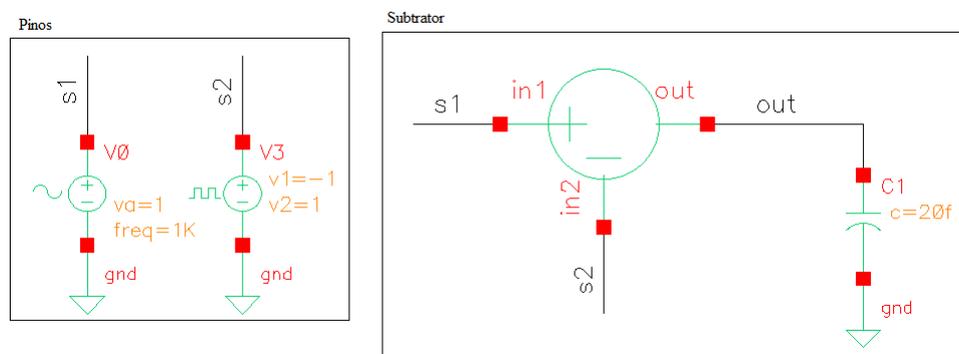


Figura 4.39 – *Testbench* do subtrator [13].

Tabela 4.18 – Descrição dos pinos do subtrator [13].

Pino	Descrição	Tipo
in1	Tensão do primeiro sinal	<i>input</i>
in2	Tensão do segundo sinal	<i>input</i>
out	Tensão do sinal de saída	<i>output</i>

Para a simulação do subtrator aplicou-se dois sinais na entrada do sistema, com os pa-

râmetros especificados na Tabela 4.19, e analisou-se o sinal de saída representado na figura 4.40.

Tabela 4.19 – Parâmetros de simulação para o subtrator [13].

Parâmetro	Descrição	Valor
Frequência	Frequências das fontes de teste	$in1 = 1K$, $in2 = 5K$
Amplitude	Amplitudes das fontes de teste	$in1 = 1$, $in2 = 1$
Ciclos	Quantidade de ciclos na análise transiente	5

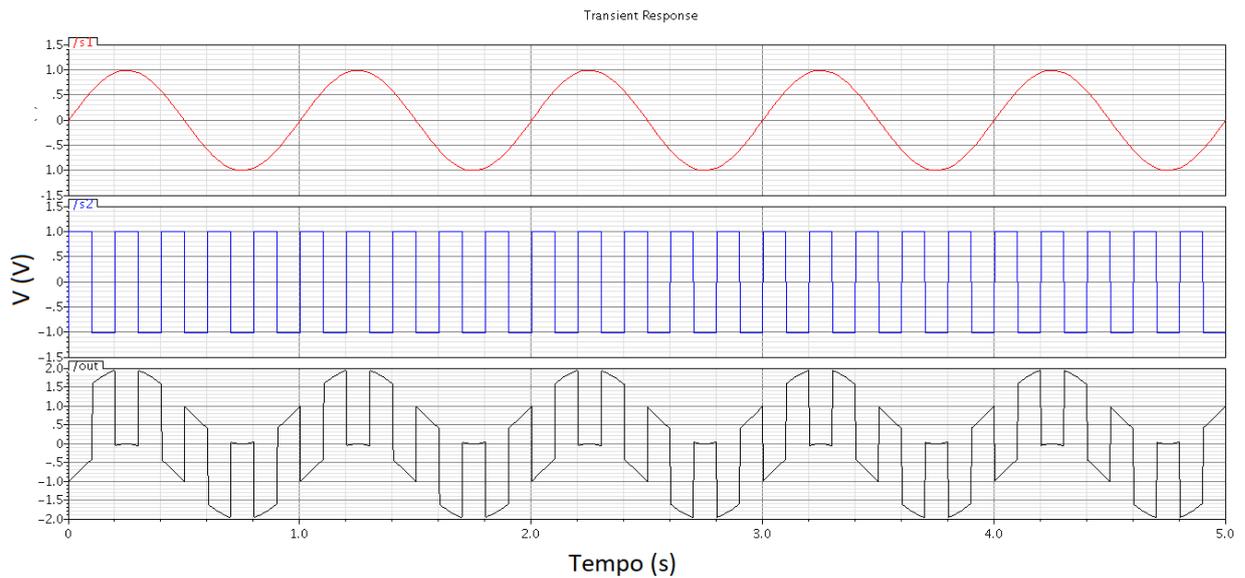


Figura 4.40 – Simulação do subtrator [13].

4.7 MODULADOR Σ - Δ LINEAR

Após a modelagem dos blocos descritos nas seções anteriores, projetou-se então os moduladores Σ - Δ *multi-bit*. A Figura 4.41-a mostra a arquitetura do modulador Σ - Δ multi-bit de 1ª ordem com quantizador linear. O modulador é formado por um filtro analógico e o quantizador de 4-bits , projetado na Seção 4.1, em um *loop de feedback*. Um modulador de 2ª ordem (4.41-b) também foi modelado para comparação e análise da consistência dos resultados obtidos nas simulações.

Além do quantizador já descrito na seção 4.1, foi adicionado ao sistema um circuito amplificador de ganho variável (*Variable Gain Amplifier - VGA*), sendo assim, o circuito do quantizador proposto passa a ser representado pela Figura 4.42.

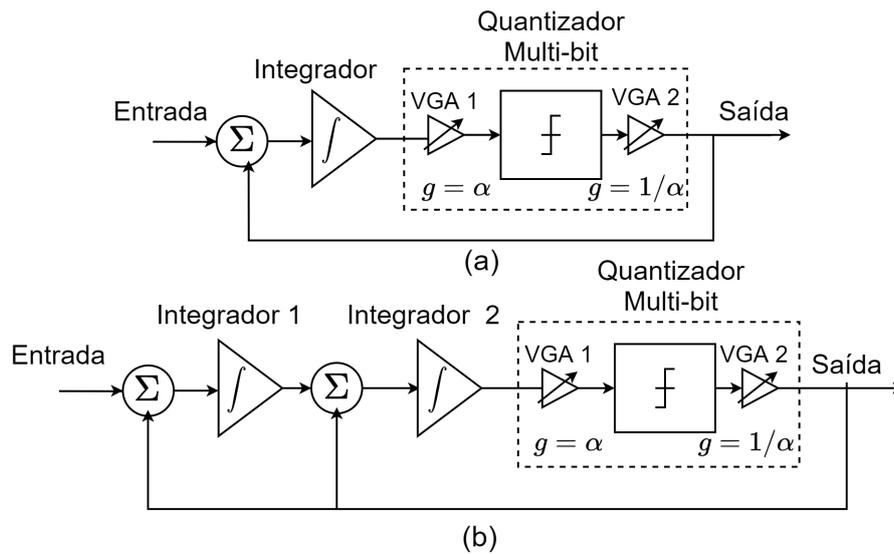


Figura 4.41 – Diagrama de blocos do modulador $\Sigma\text{-}\Delta$ *Multi-bit* de a) primeira e de b) segunda ordem (Adaptada de [10]).

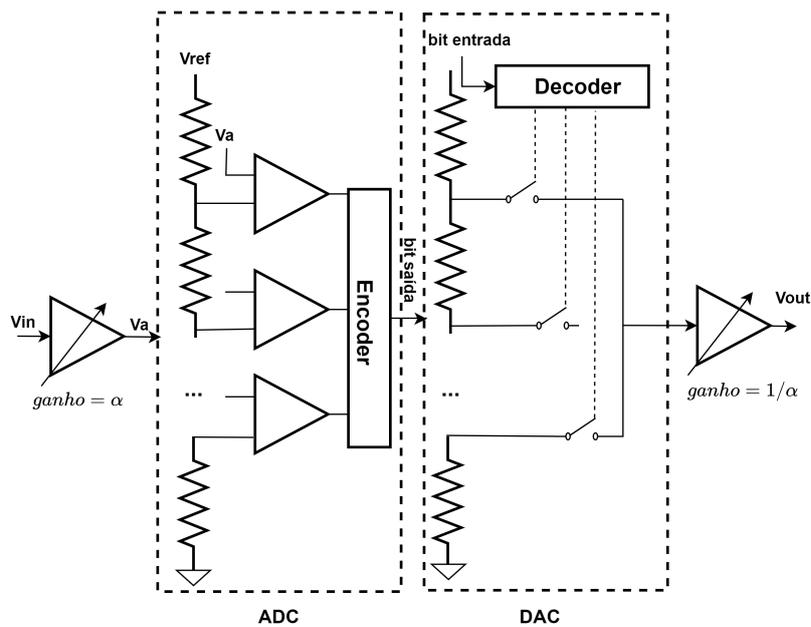


Figura 4.42 – Diagrama de blocos do quantizador proposto (Adaptada de [10]).

Um VGA é um circuito amplificador cujo ganho varia de acordo com uma tensão de controle. No sistema proposto, são utilizados dois VGAs, como ilustra a Figura 4.42, um antes do quantizador, com ganho α e um depois do quantizador, com ganho $1/\alpha$. Os ganhos dos VGAs dependem da amplitude de entrada do sinal, sendo α inversamente proporcional a essa amplitude. Esse esquema faz com que o sinal sofra uma expansão antes do processo de quantização e uma compressão para retornar a sua amplitude original depois de quantizado. Essa técnica foi introduzida para que o processo de quantização não linear funcionasse para toda a faixa dinâmica mantendo a sua característica de não linearidade.

A Tabela 4.20 apresenta as especificações do modulador projetado. Como os quantizadores não lineares apresentam vantagens em sistemas de processamento de sinais de áudio, escolheu-se uma frequência de entrada dentro da faixa de frequência audível (20 Hz a 20 KHz) [35].

Tabela 4.20 – Especificações do modulador Σ - Δ de 1ª e de 2ª ordem.

Especificações do Modulador Σ-Δ de 1ª e de 2ª Ordem	
Frequência de amostragem: f_s	1 MHz
Tensão de referência positiva ($avdd$)	1 V
Tensão de referência negativa ($avss$)	-1 V
Coeficientes de malha aberta	$A_{integrador} = 1000$
Coeficientes de realimentação	$A_{loop} = 1$
Frequência do sinal de entrada (f_{in})	976.5625
Oversampling (OSR)	512

Para a simulação dos moduladores de 1ª e de 2ª ordem aplicou-se um sinal senoidal na entrada do sistema, com parâmetros especificados na Tabela 4.20, e observou-se o comportamento dos nós internos do modulador.

A Figura 4.43 mostra os resultados de simulação do modulador de 1ª ordem para o sinal de entrada (4.43-a), o sinal de saída (4.43-b), o sinal de erro (4.43-c), representado pela diferença entre a entrada e a saída do modulador, e o sinal após o estágio de integração (4.43-d), que representa a integral do erro.

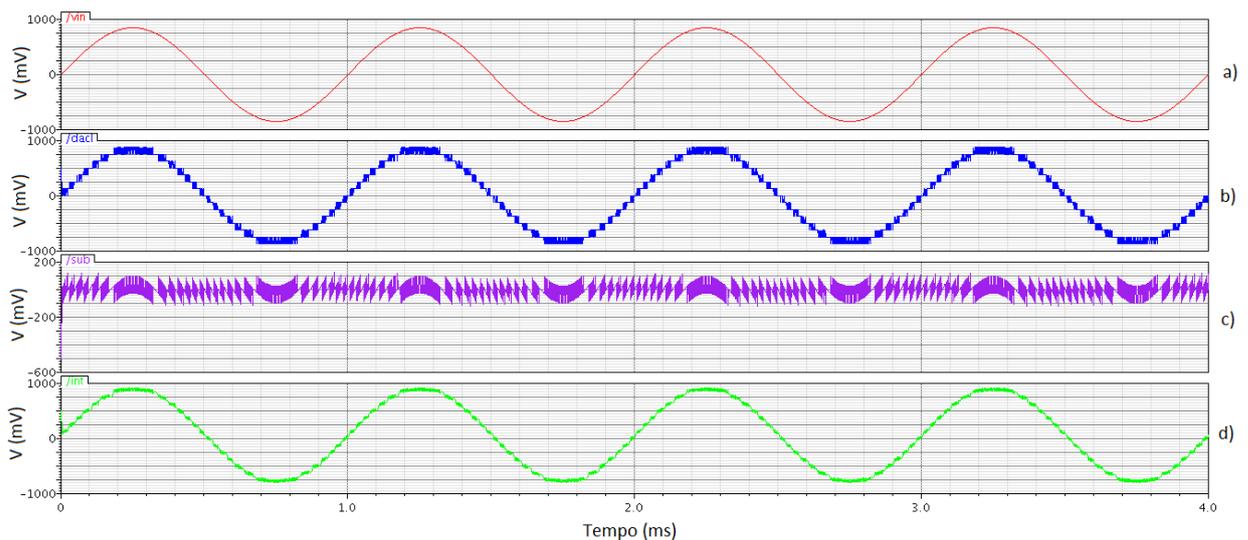


Figura 4.43 – Simulação do modulador Σ - Δ *multi-bit* de 1ª ordem com quantizador linear.

A Figura 4.44 mostra os sinais de entrada e saída do modulador de 1ª (4.44-a) e de 2ª (4.44-b) ordem sobrepostos com 1 ciclo de simulação. Observa-se que para os dois casos a

saída apresenta o comportamento esperado de um modulador $\Sigma\text{-}\Delta$ *multi-bit* descrito na seção 2.5.2.2 e que os passos de quantização são constantes devido ao uso do quantizador linear de 4 bits. Visualmente também é possível perceber que o modulador de 2ª ordem apresenta menos ruído no sinal de saída.

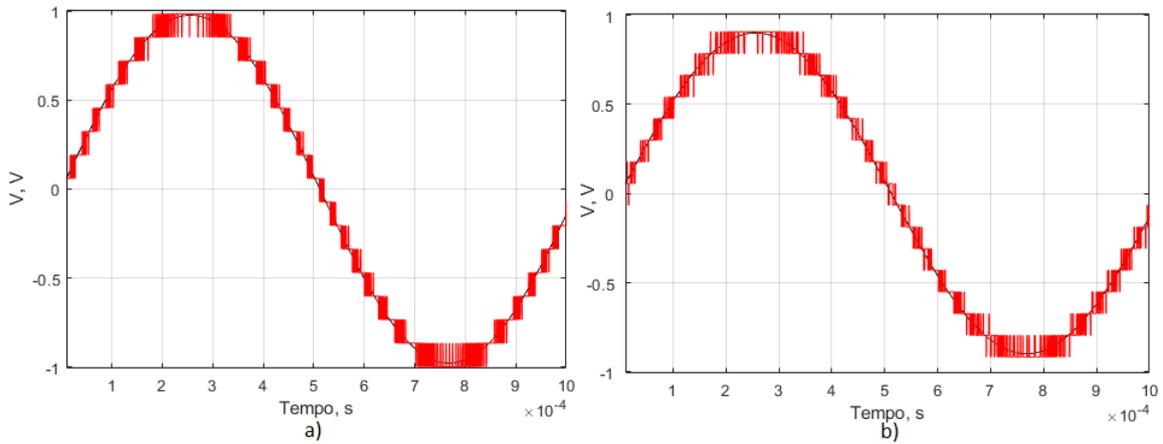


Figura 4.44 – Simulação do modulador $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador linear.

Realizou-se também uma análise em frequência dos moduladores afim de verificar o seu desempenho. A Figura 4.45 mostra as *DFTs* dos sinais de saída dos moduladores de 1ª (4.45-a) e de 2ª (4.45-b) ordem. No espectro pode-se observar o impulso em 0 dB que representa a frequência fundamental e o *noise shaping* característico do *ADC* $\Sigma\text{-}\Delta$ que, como esperado, é maior no modulador de 2ª ordem. Para o cálculo da *DFT* utilizou-se 3 ciclos de simulação e 3072 amostras para garantir uma amostragem coerente.

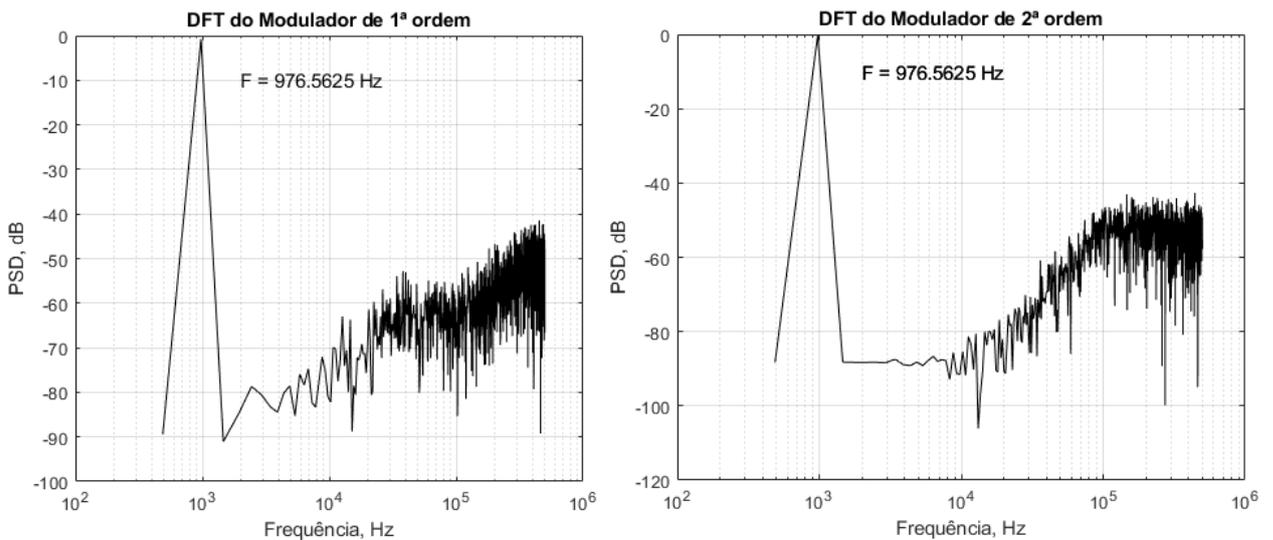


Figura 4.45 – *DFTs* dos moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador linear.

4.8 MODULADOR Σ - Δ ARCO-SENO

Para o quantizador Arco-seno, também implementou-se os moduladores Σ - Δ *multi-bit* de 1ª e de 2ª ordem. Para isso foram utilizados os mesmos blocos usados no modulador linear, com exceção do quantizador que foi substituído pelo quantizador arco-seno. O esquemático utilizado na simulação também foi o da Figura 4.41 mas utilizando o quantizador Arco-seno modelado na Seção 4.2. Também foram utilizados os VGAs na entrada e na saída do quantizador. As especificações do modulador projetado também são as mesmas mostradas na tabela 4.20.

Assim como no modulador linear, a simulação do modulador arco-seno foi feita aplicando-se um sinal senoidal na entrada e observando-se o comportamento dos nós internos do sistema.

A Figura 4.46 mostra os resultados de simulação do modulador de 1ª ordem para o sinal de entrada (4.46-a), o sinal de saída (4.46-b), o sinal de erro (4.46-c), representado pela diferença entre a entrada e a saída do modulador, e o sinal (4.46-d) que representa a integral do erro.

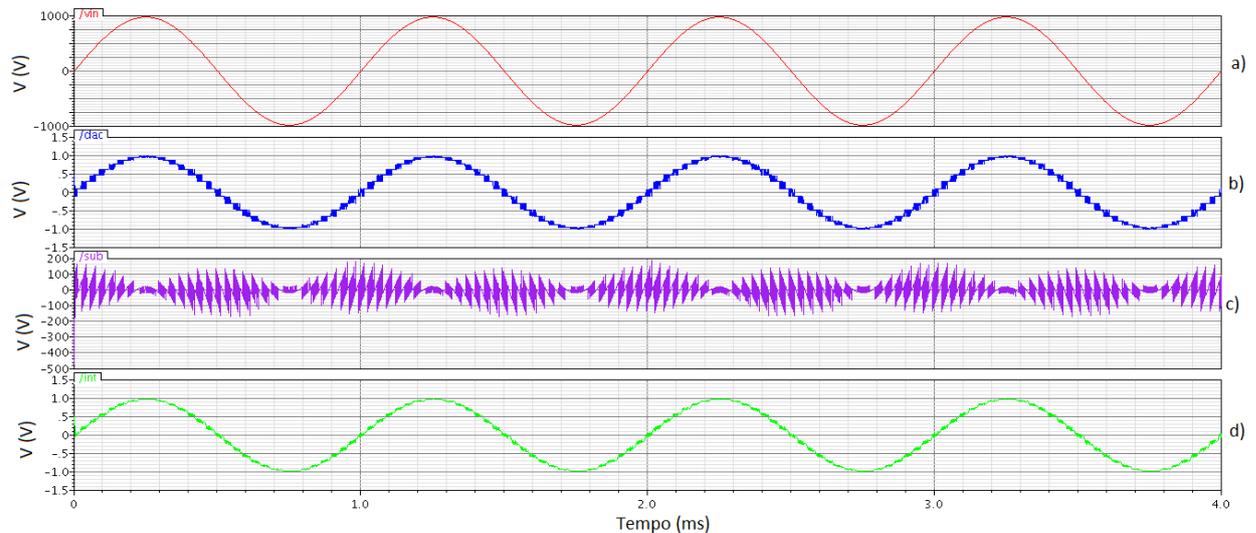


Figura 4.46 – Simulação do modulador Σ - Δ *multi-bit* de 1ª ordem com quantizador Arco-seno.

A Figura 4.47 mostra os sinais de entrada e saída do modulador de 1ª (4.47-a) e de 2ª (4.47-b) ordem, com quantizador arco-seno, sobrepostos com 1 ciclo de simulação. Observe-se que para os dois casos a saída apresenta o comportamento esperado de um modulador Σ - Δ *multi-bit* descrito na seção 2.5.2.2 e que os passos de quantização são diferentes devido a não linearidade do quantizador.

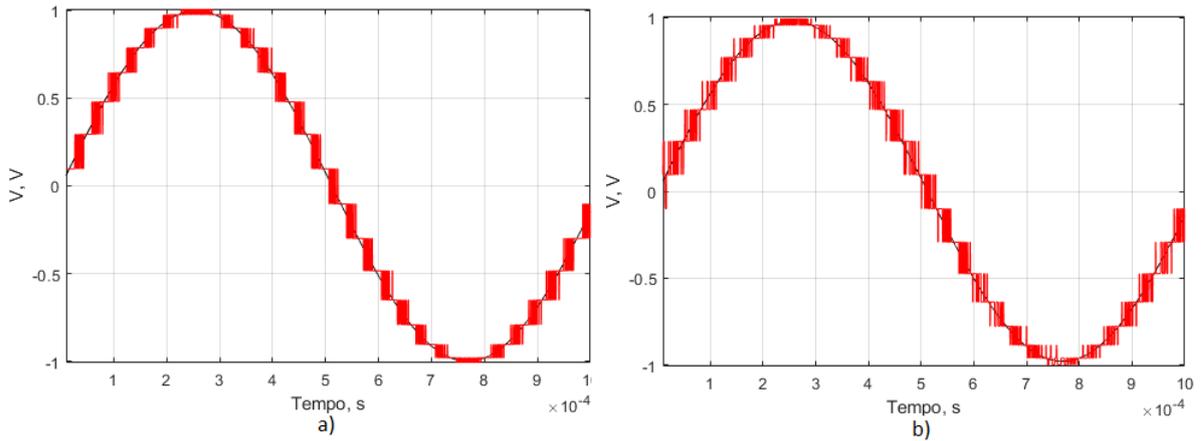


Figura 4.47 – Simulação do modulador Σ - Δ *multi-bit* de 1^a (a) e de 2^a (b) ordem com quantizador arco-seno.

A Figura 4.48 mostra o sinal de saída do modulador arco-seno para diferentes amplitudes de sinal de entrada. Como pode-se observar, o modulador mantém a sua característica de não linearidade, representada pela maior resolução nas extremidades do sinal, para todas a faixa dinâmica. Esse resultado é possível devido ao uso dos VGAs na entrada e saída do quantizador, que fazem com que o sinal sempre seja quantizado na sua escala total.

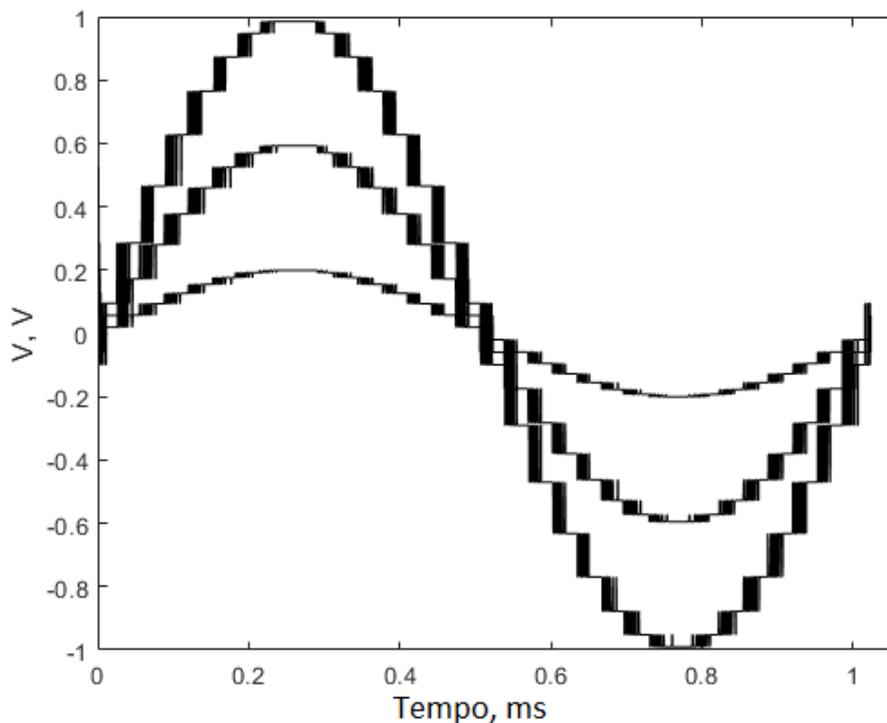


Figura 4.48 – Saída do modulador Σ - Δ *multi-bit* de 1^a ordem para diferentes amplitudes de entrada.

Assim como no modulador linear, nesse sistema também realizou-se a análise em frequência por meio da *DFT*. Para isso, utilizou-se os mesmos parâmetros e quantidade de amostras

do sistema linear. A Figura 4.49 mostra os espectros para os moduladores de 1ª (4.49-a) e de 2ª (4.49-b) ordem utilizando o quantizador arco-seno.

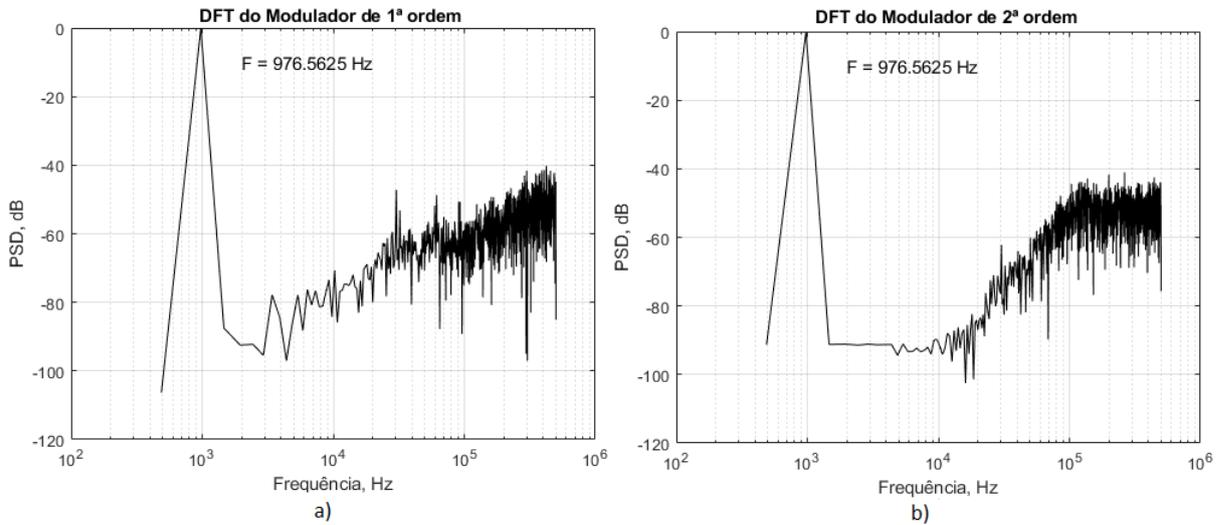


Figura 4.49 – *DFTs* dos moduladores Σ - Δ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador arco-seno.

4.9 MODULADOR Σ - Δ μ -C

Para o projeto dos moduladores com o quantizador μ -C, utilizou-se os blocos modelados para o modulador Σ - Δ e o quantizador μ -C de 4 bits modelado na seção 4.3. A Figura 4.50 mostra o esquemático utilizado para a simulação dos moduladores Σ - Δ de 1ª e de 2ª ordem utilizando o quantizador μ -C. As especificações dos moduladores projetado também são as mesmas mostradas na tabela 4.20. Além disso, nos compressores e expansores utilizou-se o parâmetro $\mu = 255$.

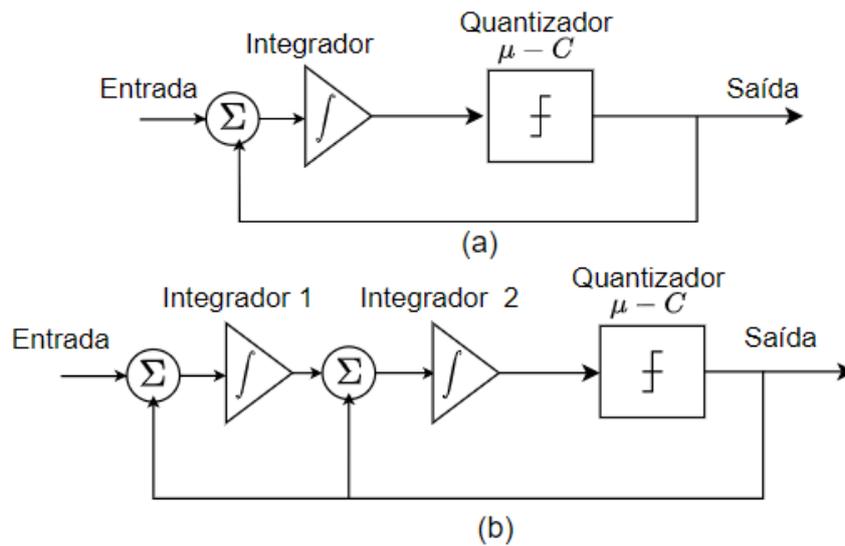


Figura 4.50 – Diagrama de blocos do modulador Σ - Δ *Multi-bit* de 1ª (a) e de 2ª (b) ordem utilizando o quantizador μ -C.

Como para os outros moduladores, a simulação do sistema foi feita aplicando-se um sinal senoidal na entrada e observando-se os sinais internos do modulador. A Figura 4.51 mostra os resultados de simulação do modulador de 1ª ordem para o sinal de entrada 4.51-a, de saída 4.51-b, de erro 4.51-c e de integral 4.51-d.

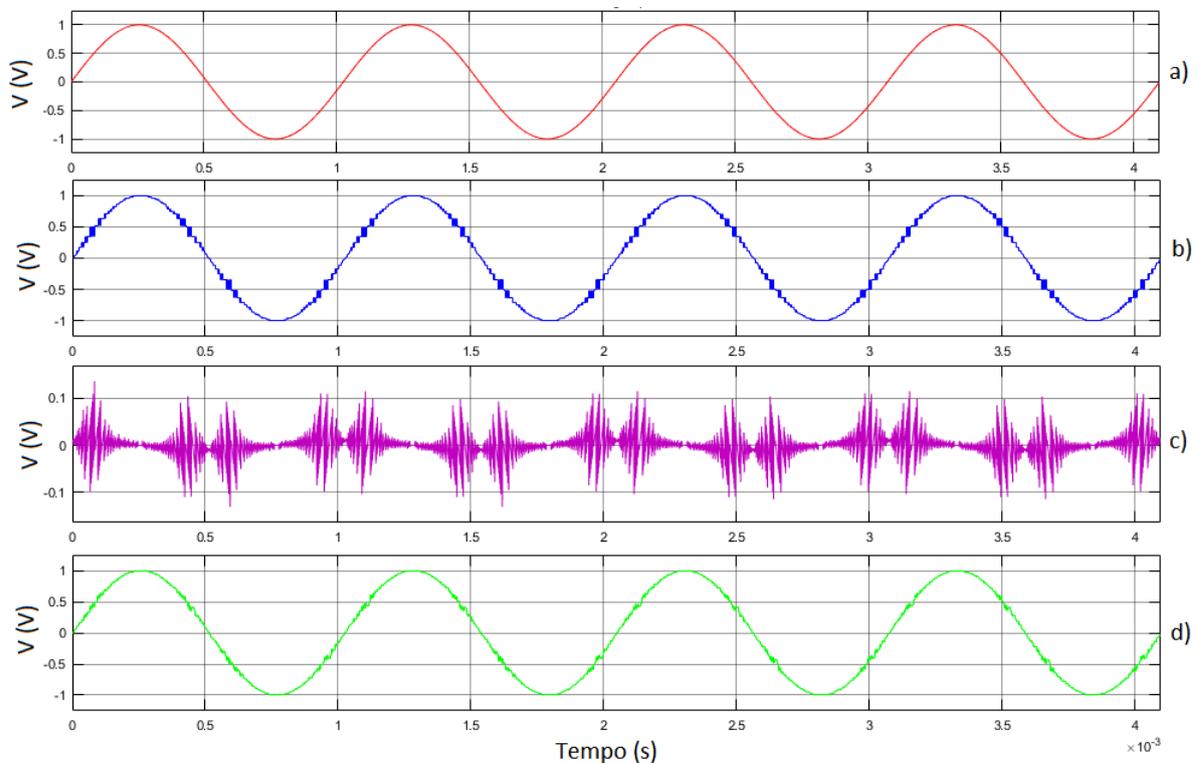


Figura 4.51 – Simulação do modulador Σ - Δ *Multi-bit* de 1ª ordem com quantizador μ -C.

A Figura 4.52 mostra os sinais de entrada e saída do modulador de 1ª e de 2ª ordem, com

quantizador μ -C com 1 ciclo de simulação.

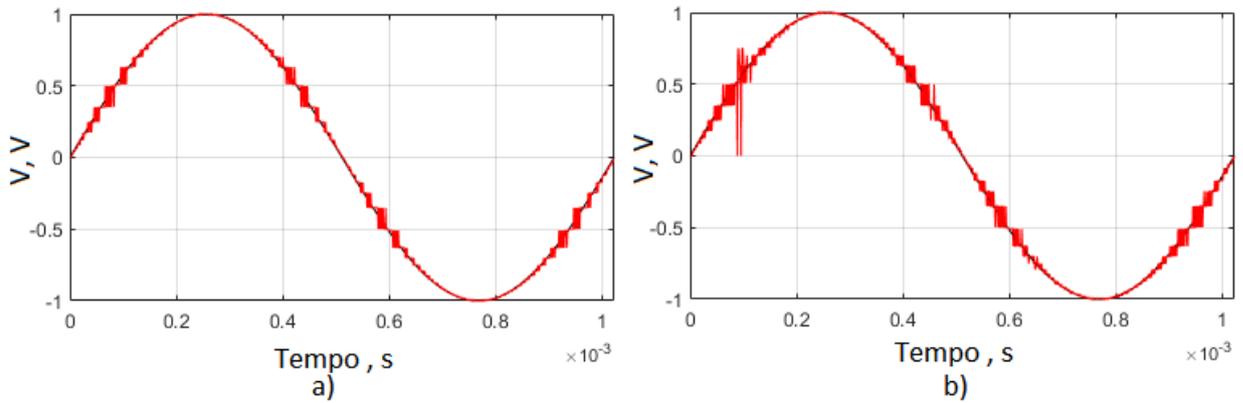


Figura 4.52 – Simulação do modulador Σ - Δ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador μ -C.

Também realizou-se uma análise em frequência por meio da *DFT*, com os mesmos parâmetros utilizados nos sistemas das seções anteriores. A Figura 4.53 mostra os espectros para os moduladores de 1ª (4.53-a) e de 2ª (4.53-b) ordem utilizando o quantizador μ -C.

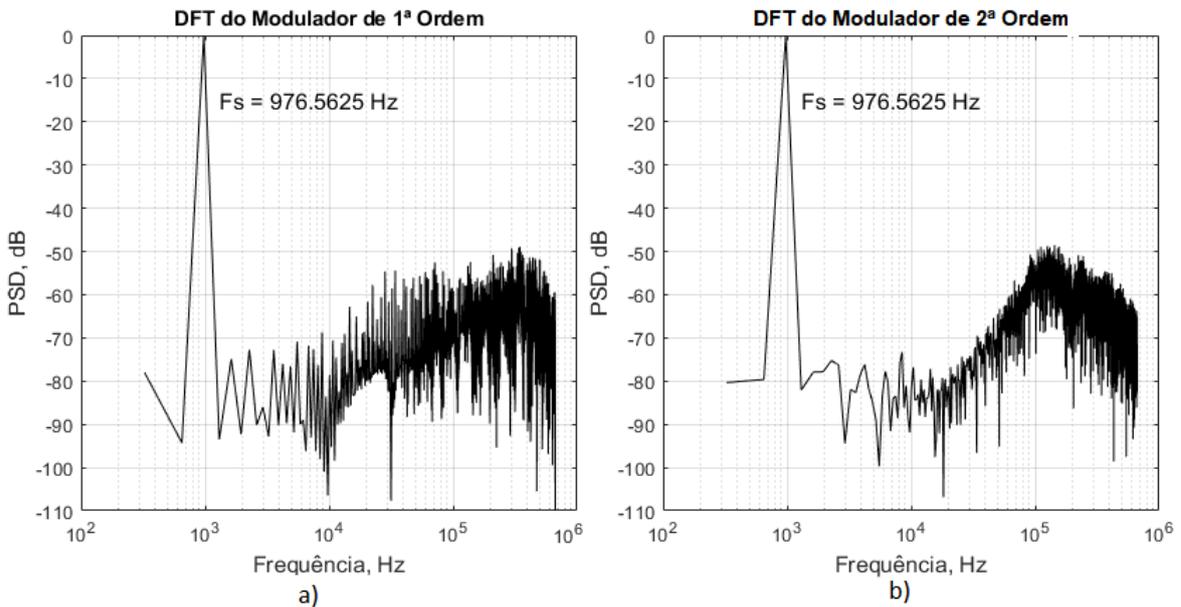


Figura 4.53 – *DFTs* dos moduladores Σ - Δ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador μ -C.

4.10 MODULADOR Σ - Δ A-C

Para o projeto dos moduladores com o quantizador A-C, também utilizou-se os mesmos blocos modelados para o modulador Σ - Δ e o quantizador A-C de 4 bits modelado na se-

ção 4.4. O esquemático utilizado na simulação dos moduladores é o mesmo da Figura 4.50, apresentado na seção anterior, com exceção do quantizador. As especificações dos moduladores projetados também são as mesmas mostradas na tabela 4.20 e além disso, utilizou-se o parâmetro $A = 87.56$ nos compressores e expansores.

A simulação do foi feita aplicando-se um sinal senoidal na entrada e observando-se os sinais do modulador. A Figura 4.54 mostra os resultados de simulação do modulador de 1ª ordem para o sinal de entrada 4.54-a, de saída 4.54-b, de erro 4.54-c e de integral 4.54-d.

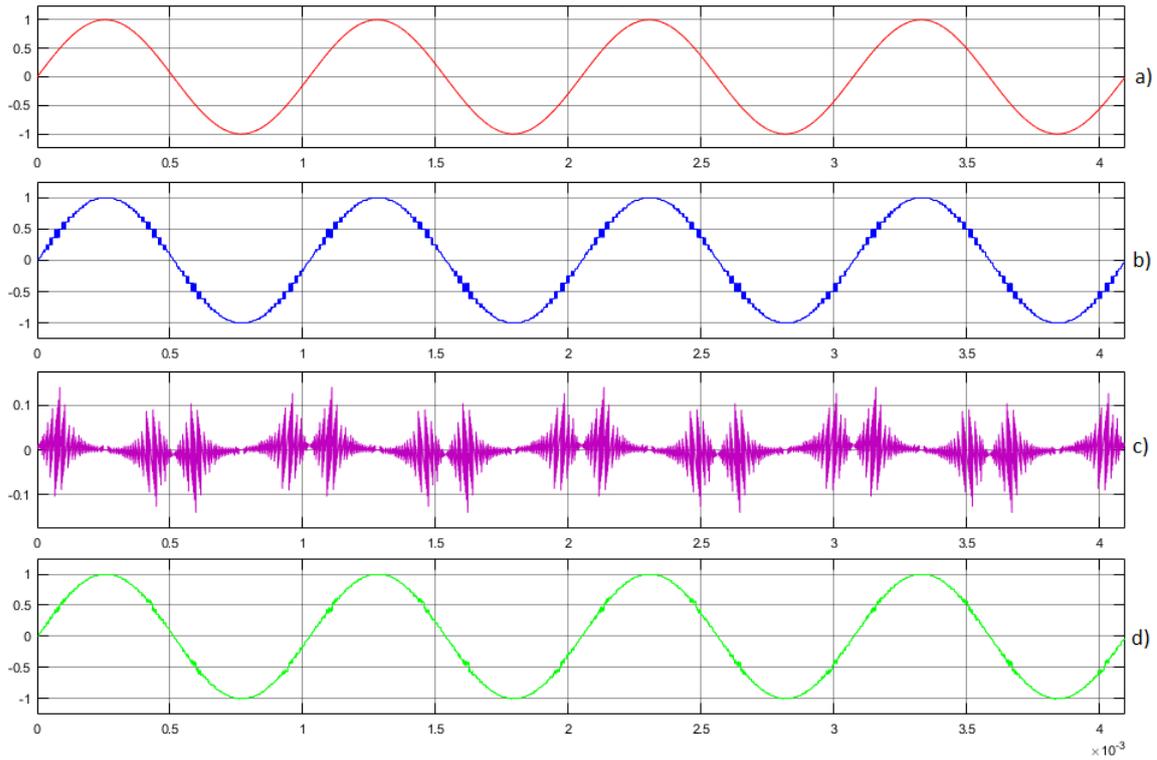


Figura 4.54 – Simulação do modulador Σ - Δ *Multi-bit* de 1ª ordem com quantizador A-C.

A Figura 4.55 mostra os sinais de entrada e saída do modulador de 1ª e de 2ª ordem, com quantizador A-C com 1 ciclo de simulação.

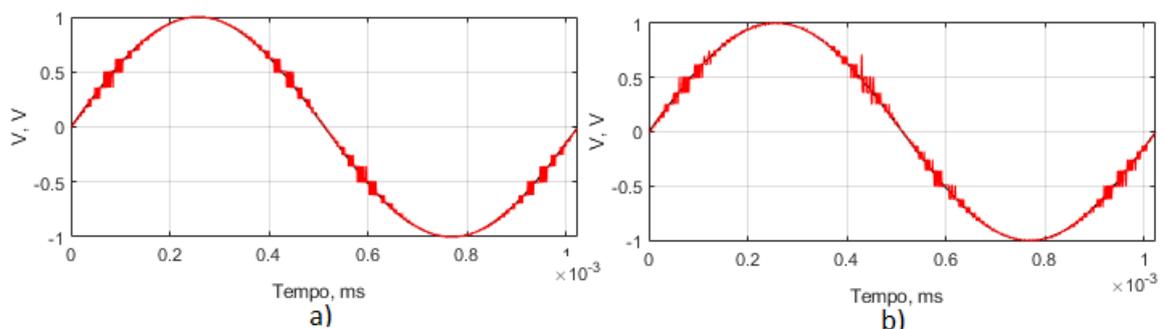


Figura 4.55 – Simulação do modulador Σ - Δ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador A-C.

Também realizou-se uma análise em frequência por meio da *DFT* utilizando-se os mesmos parâmetros das seções anteriores. A Figura 4.56 mostra os espectros para os moduladores de 1^a (4.56-a) e de 2^a (4.56-b) ordem utilizando o quantizador A-C.

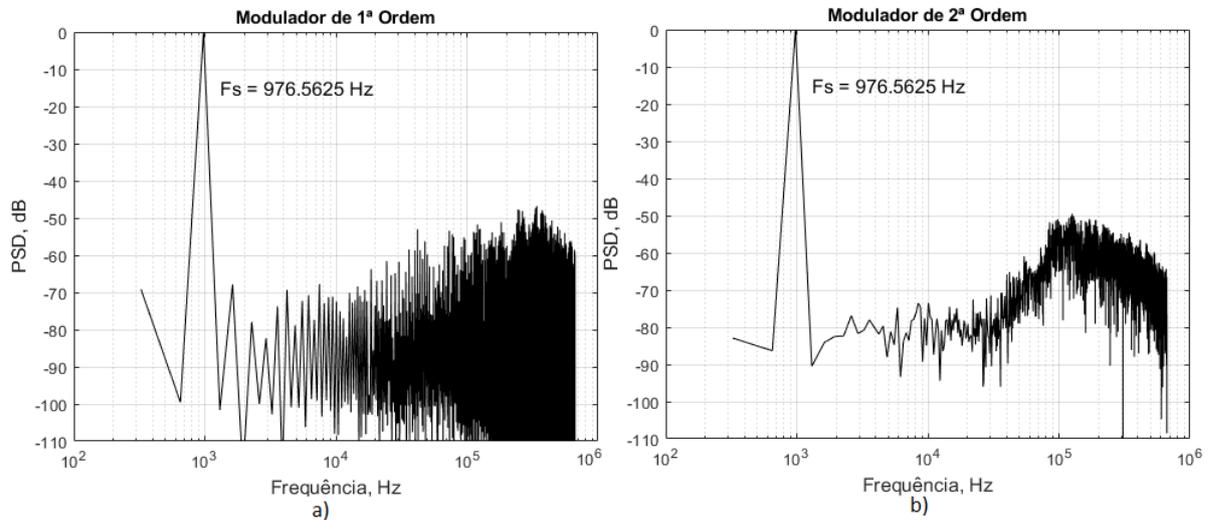


Figura 4.56 – *DFTs* dos moduladores Σ - Δ *multi-bit* de 1^a (a) e de 2^a (b) ordem com quantizador A-C.

4.11 COMPARAÇÃO ENTRE MODULADORES Σ - Δ *MULTI-BIT*

Nessa seção será feita a comparação entre o desempenho obtido pelos moduladores Σ - Δ *multi-bit* de 1^a ordem e de 2^a com os quantizadores Linear, Arco-seno, μ -C e A-C. A comparação foi feita analisando-se as respostas no domínio do tempo e da frequência para os moduladores e realizando-se a extração dos parâmetros dinâmicos de cada sistema sob as mesmas condições de simulação. Os parâmetros utilizados na simulação dos moduladores foram os mesmos descritos nas seções anteriores.

Iniciou-se a análise pelo modulador Linear que foi projetado como referência para os outros modelos. A Figura 4.57 mostra os sinais de entrada e saída dos moduladores de 1^a e de 2^a ordem e seus respectivos erros de quantização. Para os dois casos pode-se observar que os passos de quantização do sinal de saída são constantes para toda a faixa dinâmica e que o erro de quantização apresenta o padrão com formato dente de serra/sino. Por outro lado, o modulador arco-seno, cuja análise transiente é ilustrada na Figura 4.58, apresenta níveis de quantização menores nas extremidades do sinal e maiores no centro da faixa dinâmica. Essa resposta gera um erro de quantização modular que é menor nos extremos do sinal.

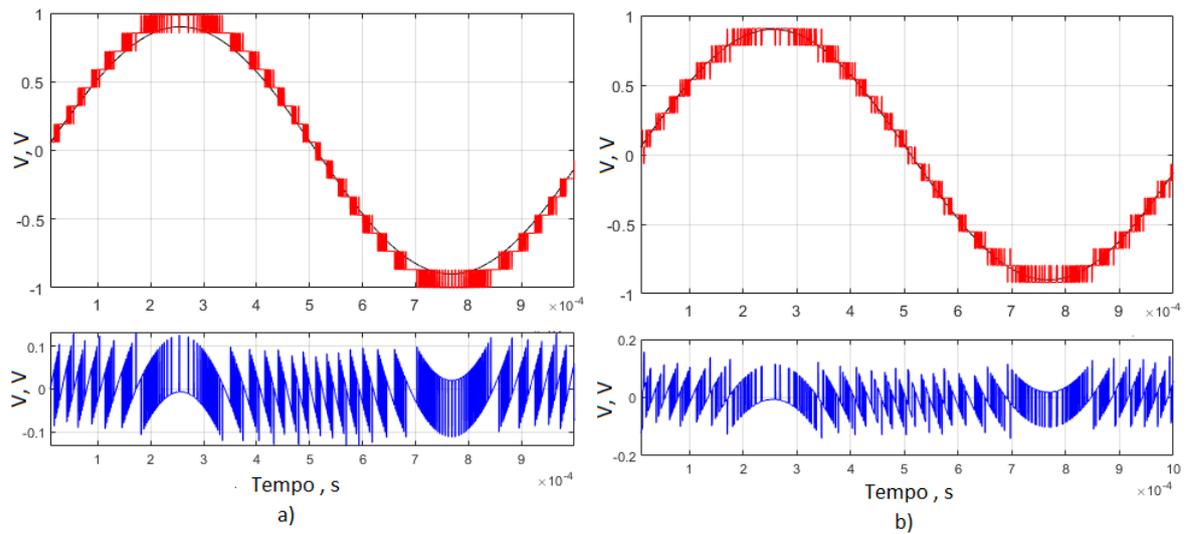


Figura 4.57 – Sinais de entrada, saída e erro de quantização para o modulador Σ - Δ *Multi-bit* de a) primeira e de b) segunda ordem com quantizador linear.

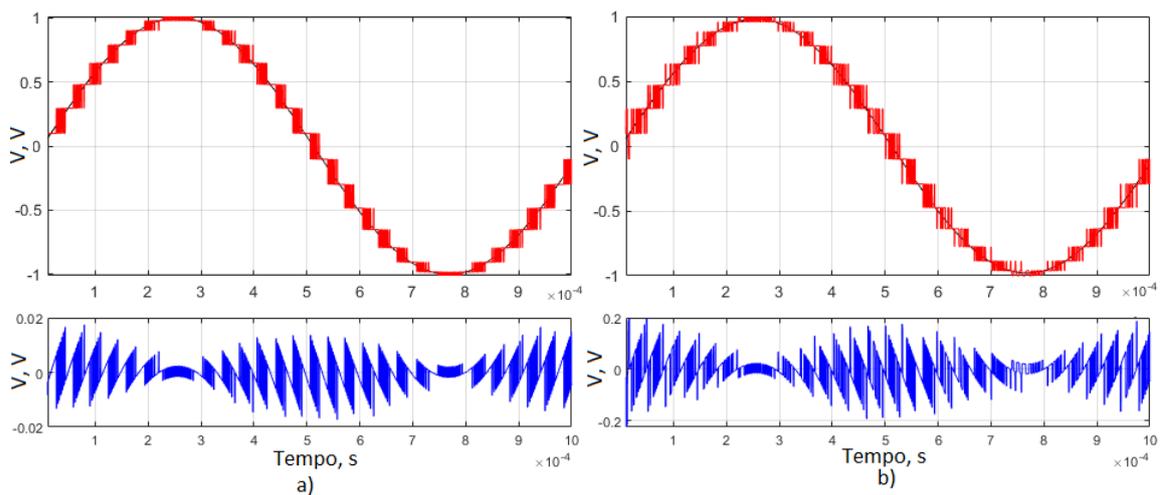


Figura 4.58 – Sinais de entrada, saída e erro de quantização para o modulador Σ - Δ *Multi-bit* de a) primeira e de b) segunda ordem com quantizador arco-seno.

Seguindo com a análise dos moduladores não lineares, a Figura 4.59 mostra os sinais de entrada, saída e o erro de quantização dos moduladores de 1^a e de 2^a ordem com quantizador μ -C e a Figura 4.60 com o quantizador A-C. Pode-se observar que os moduladores apresentam comportamentos bastante semelhantes. Assim como no caso do modulador arco-seno, os moduladores μ -C e A-C apresentam níveis de quantização que variam durante a faixa dinâmica mas nesse caso, a maior resolução do sinal é concentrada tanto nos extremos quanto no centro da faixa dinâmica. A menor resolução dos moduladores se concentra em 1/4 da faixa dinâmica que é onde ocorre o chaveamento entre os quantizadores que estão sendo combinados. Esse comportamento faz com que os erros de quantização também sejam modulares.

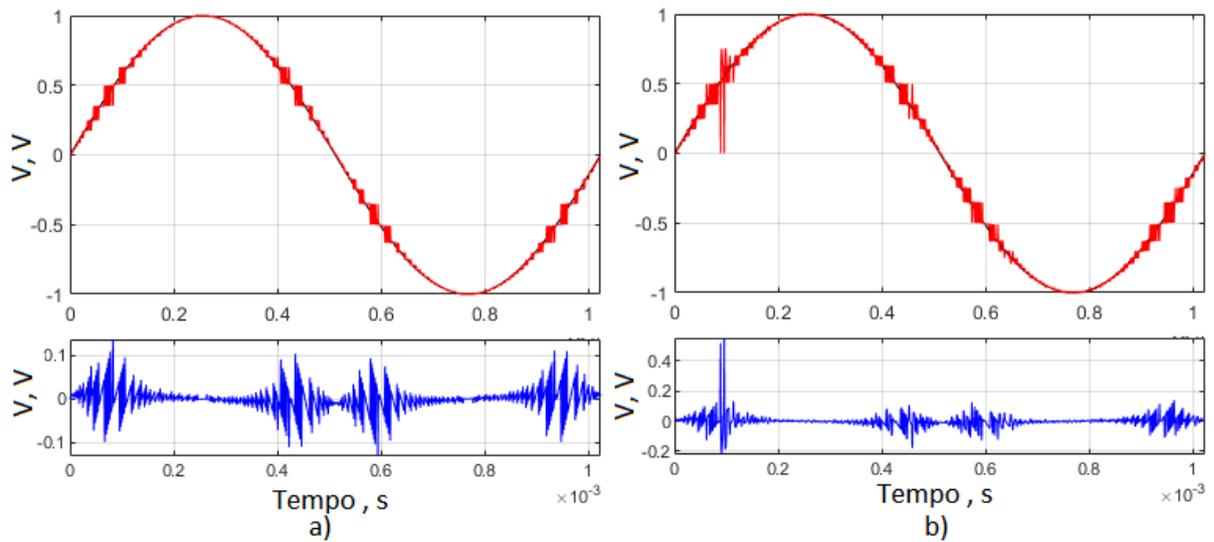


Figura 4.59 – Análise transiente do modulador Σ - Δ *Multi-bit* de a) primeira e de b) segunda ordem com quantizador μ -C.

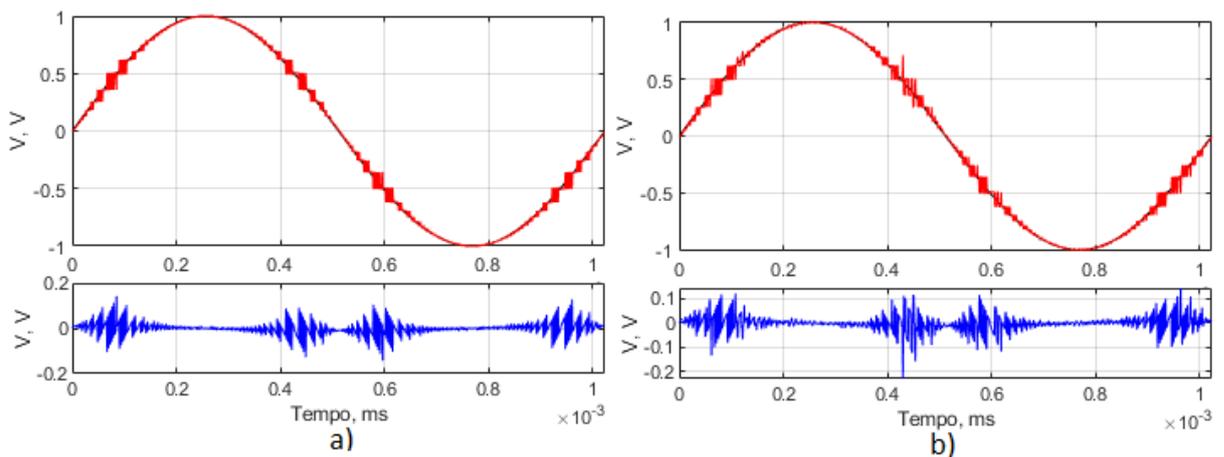


Figura 4.60 – Análise transiente do modulador Σ - Δ *Multi-bit* de a) primeira e de b) segunda ordem com quantizador A-C.

A comparação no domínio do tempo entre todos os moduladores projetados pode ser melhor visualizada no Anexo B.2.

A Figura 4.61 mostra as *DFTs* dos moduladores Linear e Arco-seno sobrepostas. Os gráficos mostram as respostas dos moduladores de 1ª e de 2ª ordem que apresentam o comportamento padrão de um modulador Σ - Δ onde o ruído é jogado para frequências mais altas devido ao efeito de *noise-shaping*. Esse efeito fica mais evidente a medida que se aumenta a ordem do modulador. Analisando-se os gráficos, pode-se perceber que o *noise floor* do modulador arco-seno é menor que o do modulador linear, principalmente perto da frequência fundamental.

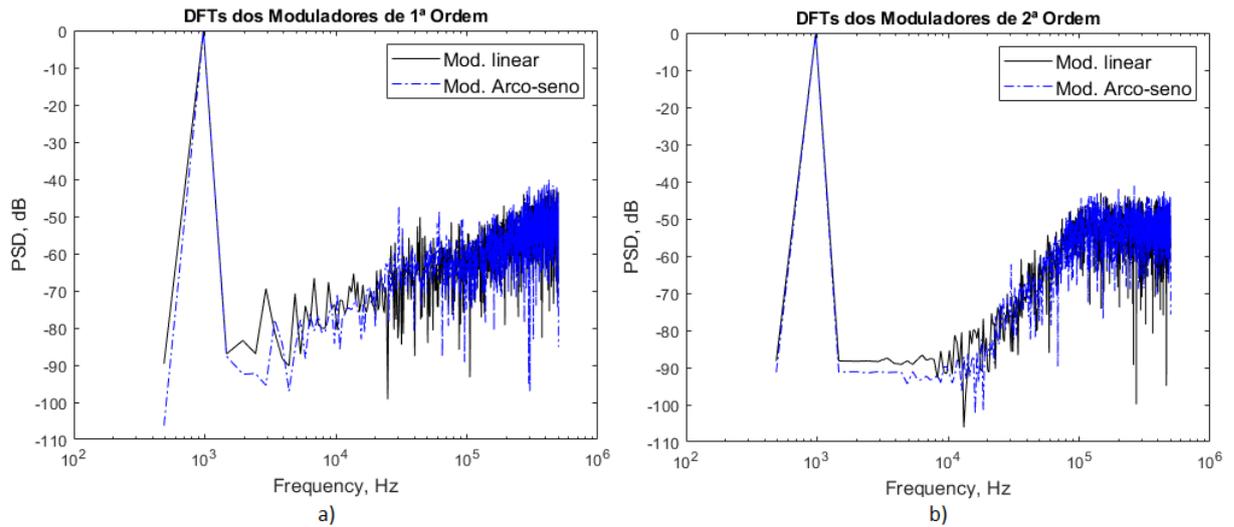


Figura 4.61 – *DFTs* dos moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizadores linear e arco-seno.

A Figura 4.62 mostra as *DFTs* dos moduladores Linear, Arco-seno e $\mu\text{-C}$ ou *A-C* (Figura 4.63) sobrepostas. Nesses casos pode-se observar que os moduladores $\mu\text{-C}$ e *A-C* geram menos ruído no espectro ao serem comparados com os moduladores linear e arco-seno. No entanto, perto da frequência fundamental esses moduladores apresentam uma amplitude de ruído ou de harmônicas maior do que os outros dois.

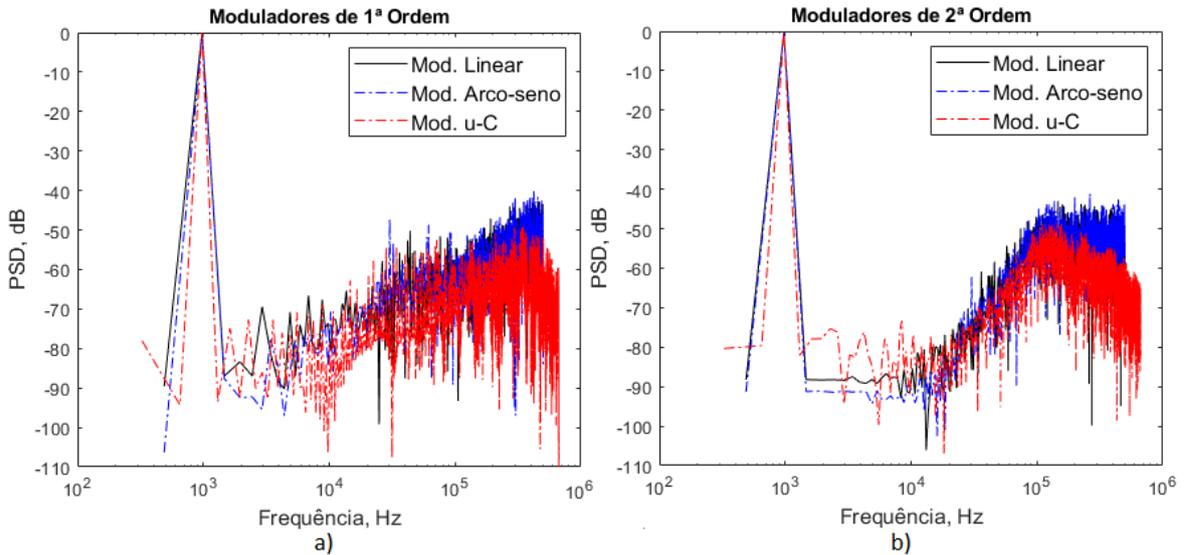


Figura 4.62 – *DFTs* dos moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª e de 2ª ordem com quantizadores linear e arco-seno e $\mu\text{-C}$.

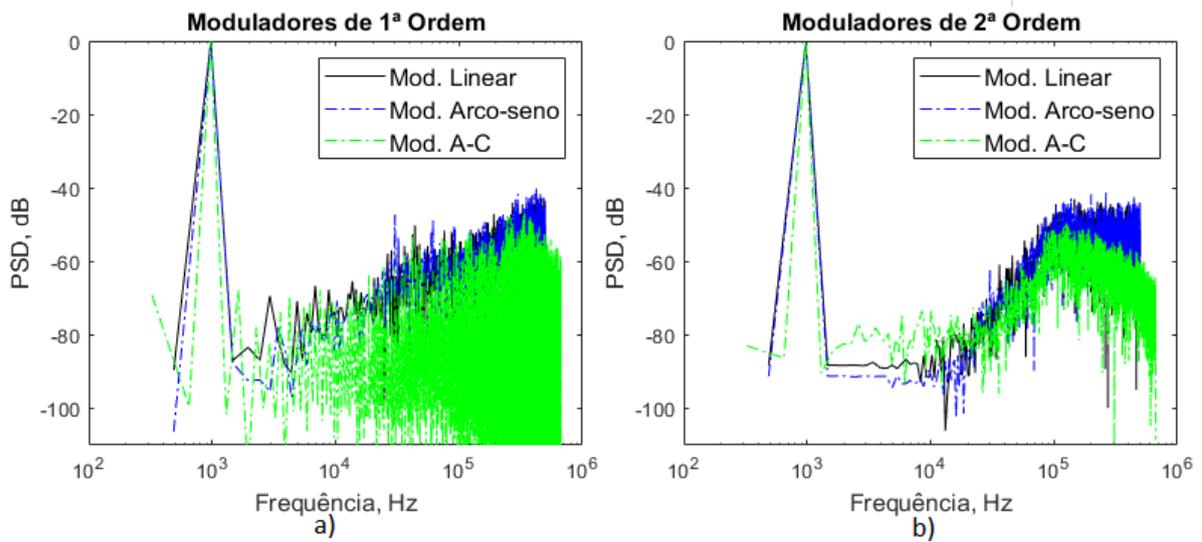


Figura 4.63 – DFTs dos moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizadores linear e arco-seno e A-C.

As Figuras 4.64 e 4.65 mostram um *zoom* do espectro dos moduladores $\mu\text{-C}$ e A-C, respectivamente, perto da frequência fundamental. Nelas estão destacadas as posições dos 9 primeiros harmônicos dos sinais. Como pode-se observar, os moduladores $\mu\text{-C}$ e A-C de 1ª ordem apresentam harmônicos com amplitudes menores do que os moduladores linear e arco-seno, sugerindo que geram uma menor distorção harmônica no sistema se comparado com os outros dois. Já no caso dos moduladores de 2ª ordem, os moduladores $\mu\text{-C}$ e A-C apresentam harmônicos com amplitudes maiores que os outros dois, principalmente as harmônicos ímpares, sugerindo uma maior distorção.

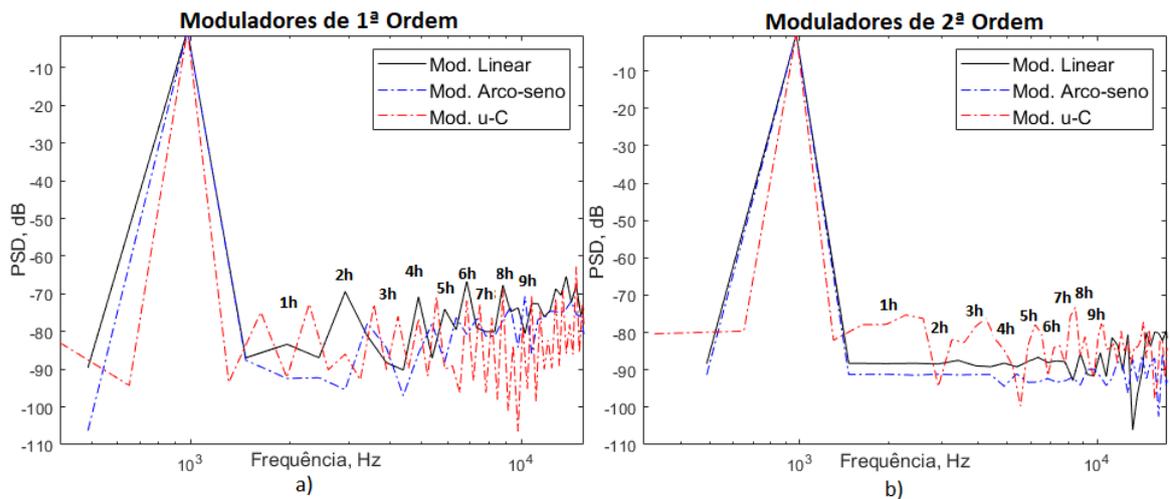


Figura 4.64 – DFTs dos moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizadores linear, arco-seno e $\mu\text{-C}$ e marcação dos harmônicos principais.

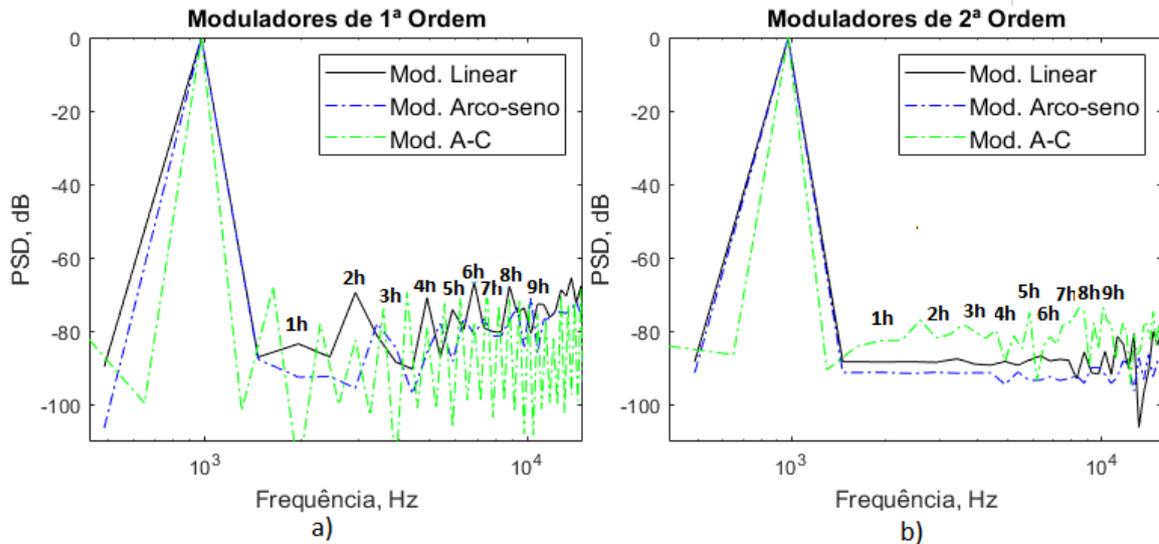


Figura 4.65 – *DFTs* dos moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizadores linear, arco-seno e *A-C* e marcação dos harmônicos principais.

A partir da análise em frequência dos moduladores e utilizando-se a ferramenta de análise espectral do Cadence, extraiu-se os seguintes parâmetros dinâmicos dos moduladores: *SNHR*, *SINAD*, *ENOB*, *SFDR* e *THD*, cujas características foram apresentadas na seção 2.4. Os parâmetros são apresentados abaixo nas Tabelas 4.22 e 4.23 e em forma de gráficos para facilitar a comparação de desempenho entre os quatro moduladores apresentados: Linear, Arco-seno, $\mu\text{-}c$ e *A-C*. A aquisição dos parâmetros foi feita variando-se a tensão de entrada dos moduladores. Essa medida é apresentada nos gráficos na unidade de *Decibels Full Scale (dBFS)* que mostra a quantidade de decibéis relativos à amplitude máxima do sinal de entrada como pode ser visto na equação 4.1:

$$Amplitude_{dBFS} = 20 \log_{10} \frac{V_{in}}{V_{full-scale}}, \quad (4.1)$$

onde V_{in} é a tensão de entrada e $V_{full-scale}$ é a tensão máxima de entrada.

A Tabela 4.21 mostra os resultados da distorção harmônica para os moduladores de 1ª e de 2ª ordem com os quantizadores projetados. No cálculo da *THD* foi utilizada a tensão de tensão máxima de entrada (1Vpp). Como pode-se observar, para os moduladores de 1ª ordem, o quantizador *A-C* foi o que gerou a menor distorção harmônica no modulador. Esse resultado esta de acordo com o que foi apresentado na Figura 4.65 pois ela mostra que comparado aos outros sistemas o modulador *A-C* apresentou harmônicos menores perto da frequência fundamental. Já para o caso dos moduladores de 2ª ordem, o quantizador Arco-seno foi o que gerou uma menor distorção harmônica. Esse resultado era esperado uma vez que [11] chegou a conclusão de que o quantizador arco-seno não reduz a distorção e o ruído de quantização é no espectro total, apenas os desloca para altas frequências.

Tabela 4.21 – THD dos moduladores Σ - Δ *multi-bit* de 1ª e de 2ª ordem projetados com os quantizadores Linear, Arco-seno, μ -C e A-C.

Parâmetro	Moduladores de 1ª Ordem				Moduladores de 2ª Ordem			
	Linear	Arco-seno	u-C	A-C	Linear	Arco-seno	u-C	A-C
THD (%)	6.90	6.78	4.07	3.85	11.34	8.66	11.93	12.01
THD (dB)	-23.21	-23.37	-28.27	-27.78	-18.90	-21.24	-18.46	-18.40

Tabela 4.22 – Parâmetros dinâmicos dos moduladores Σ - Δ *multi-bit* de 1ª e de 2ª ordem projetados com os quantizadores Linear e Arco-seno.

Modulador Σ - Δ *Multi-bit* com Quantizador Linear

Tensão de Entrada	Moduladores de 1ª Ordem				Moduladores de 2ª Ordem			
	SNHR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)	SNHR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
100 u	-200.00	17.37	20.04	2.59	-200.52	4.85	8.58	2.40
122.1 m	48.01	45.41	54.40	7.25	49.62	46.80	57.63	7.48
488.9 m	49.26	45.94	54.89	7.33	49.66	47.07	57.45	7.52
611.2 m	49.14	44.65	51.10	7.12	49.96	47.76	56.55	7.64
733.4 m	47.99	44.88	53.19	7.16	47.85	45.91	55.02	7.33
855.6 m	48.33	44.33	52.74	7.07	48.43	47.38	55.82	7.57
1.0	47.62	44.32	50.78	7.20	48.72	47.50	57.12	7.59
1.1	46.26	21.97	25.46	3.35	5.77	4.05	12.25	2.00

Modulador Σ - Δ *Multi-bit* com Quantizador Arco-seno

Tensão de Entrada	Moduladores de 1ª Ordem				Moduladores de 2ª Ordem			
	SNHR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)	SNHR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
100 u	-200.25	17.37	24.04	2.59	-200.21	17.37	24.04	2.45
122.1 m	53.14	44.76	51.69	7.14	46.62	44.80	54.06	7.15
488.9 m	58.04	45.07	53.85	7.19	46.83	45.61	55.14	7.28
611.2 m	60.03	45.25	54.49	7.22	49.02	46.70	55.76	7.46
733.4 m	62.55	44.76	51.69	7.14	49.27	47.75	58.35	7.63
855.6 m	71.89	45.32	54.54	7.23	47.45	45.22	55.54	7.22
1.0	85.46	45.49	52.99	7.26	49.30	47.02	57.01	7.51
1.1	26.78	21.97	25.41	3.35	8.44	5.15	10.40	2.12

Tabela 4.23 – Parâmetros dinâmicos dos moduladores Σ - Δ *multi-bit* de 1ª e de 2ª ordem projetados com os quantizadores μ -C e A-C.

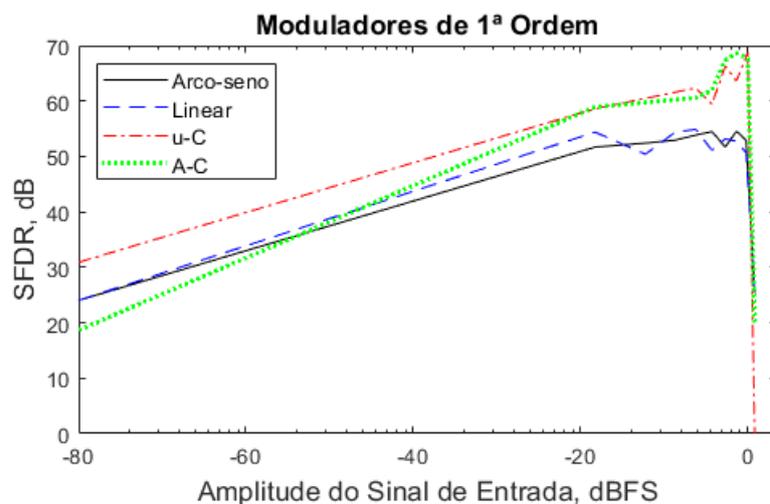
Modulador Σ - Δ *Multi-bit* com Quantizador A-C

Tensão de entrada (V)	Modulador de 1ª Ordem				Modulador de 2ª Ordem			
	SNHR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)	SNHR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
100 u	21.39	11.42	18.85	1.60	13.51	13.45	20.92	1.94
122.1 m	54.18	52.24	58.90	8.38	58.97	56.91	66.97	9.16
488.9 m	57.61	53.21	60.59	8.54	58.13	56.93	65.86	9.16
611.2 m	58.67	55.67	61.95	8.95	59.65	58.16	67.44	9.36
733.4 m	60.84	58.97	67.34	9.50	60.19	58.93	68.71	9.49
855.6 m	61.52	60.75	68.63	9.79	63.34	62.25	72.76	10.05
1.0	60.12	59.52	67.73	9.59	65.62	63.89	73.40	10.32
1.1	20.45	20.32	20.67	1.23	30.05	20.40	20.45	1.6

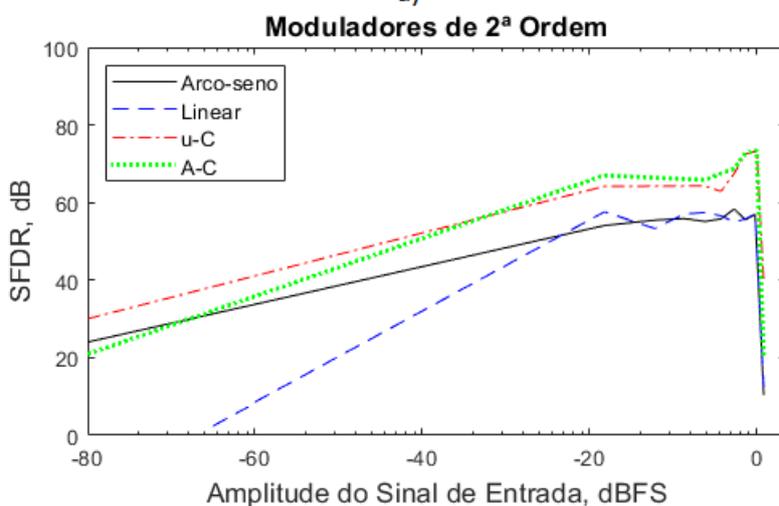
Modulador Σ - Δ *Multi-bit* com Quantizador u-C

Tensão de entrada (V)	Modulador de 1ª Ordem				Modulador de 2ª Ordem			
	SNHR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)	SNHR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
100 u	27.76	23.98	30.92	3.69	23.35	22.39	30.03	3.42
122.1 m	53.91	51.59	58.60	8.27	56.32	55.25	64.17	8.88
488.9 m	55.04	52.89	62.35	8.49	58.64	56.41	64.3	9.07
611.2 m	57.33	54.41	59.40	8.74	55.90	54.24	62.86	8.71
733.4 m	61.33	60.02	66.33	9.67	60.3	58.91	67.50	9.49
855.6 m	62.86	58.21	63.60	9.37	64.49	63.15	72.52	10.20
1.0	64.31	61.65	68.77	9.94	65.37	64.71	79.29	10.46
1.1	35.04	30.68	0.84	1.79	30.23	30.34	40.35	1.37

A Figura 4.66 mostra os gráficos da *SFDR* para os moduladores de 1ª e de 2ª ordem com os quatro quantizadores projetados. Esses gráficos foram extraídos das Tabelas 4.22 e 4.23 e como pode-se observar na figura, tanto para os moduladores de 1ª quanto para os de 2ª ordem, os quantizadores μ -C e A-C foram os que apresentaram uma maior *SFDR* principalmente para as amplitudes mais altas do sinal. O quantizador μ -C foi o que mostrou um melhor desempenho, apresentando uma melhoria de até 15.88 dBs comparado ao quantizador Arco-seno e de até 18.09 dBs comparado ao quantizador Linear. Para os moduladores de 2ª ordem essa melhoria foi de até 16.33 dBs em relação ao quantizador Arco-seno e de até 16.40 dBs em relação ao quantizador Linear.



a)



b)

Figura 4.66 – Gráfico da $SFDR$ para os moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador Linear, Arco-seno, $\mu\text{-C}$ e $A\text{-C}$.

A Figura 4.67 mostra os gráficos da $SNHR$ para os moduladores projetados. A $SNHR$ mostra a relação sinal-ruído do sistema sem o efeito da distorção harmônica. Como pode-se observar na figura, para os moduladores de 2ª ordem não houve uma variação muito grande entre a $SNHR$ dos moduladores, já para os moduladores de 1ª ordem, o quantizador Arco-seno foi o que apresentou a maior $SNHR$, com uma diferença de até 37.84 dBs em relação ao quantizador linear. Esse resultado está de acordo com as Figuras 4.64 e 4.65 que mostram que o quantizador Arco-seno é o que apresenta espúrios com menores amplitudes perto da frequência fundamental.

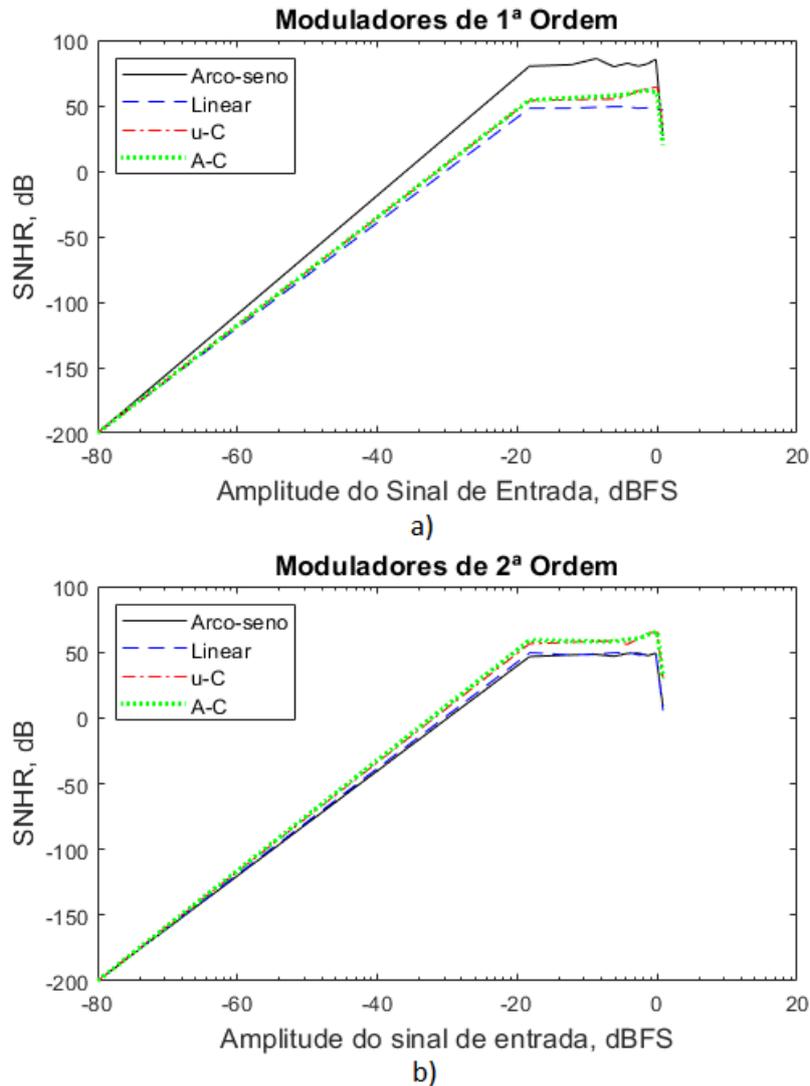


Figura 4.67 – Gráfico da *SNHR* para os moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador Linear, Arco-seno, $\mu\text{-C}$ e *A-C*.

A Figura 4.68 mostra os gráficos da *SINAD* para os moduladores projetados. Como explicado anteriormente, a *SINAD* é um parâmetro que leva em conta todo o ruído e distorção harmônica do sinal. Como pode-se observar, tanto para os moduladores de 1ª quanto para os de 2ª ordem, o quantizador $\mu\text{-C}$ é o que apresenta a melhor *SINAD*, seguido pelo quantizador *A-C*. Para o caso dos sistemas de 1ª ordem, o modulador $\mu\text{-C}$ apresentou uma melhoria de até 17.33 dBs em relação ao modulador Linear e de até 16.16 dBs em relação ao modulador Arco-seno. Para os sistemas de 2ª ordem essa melhoria foi de até 17.69 dBs em relação ao modulador Linear. Esse efeito foi alcançado devido a diminuição do ruído de quantização dos moduladores logarítmicos ao longo de todo o espectro.

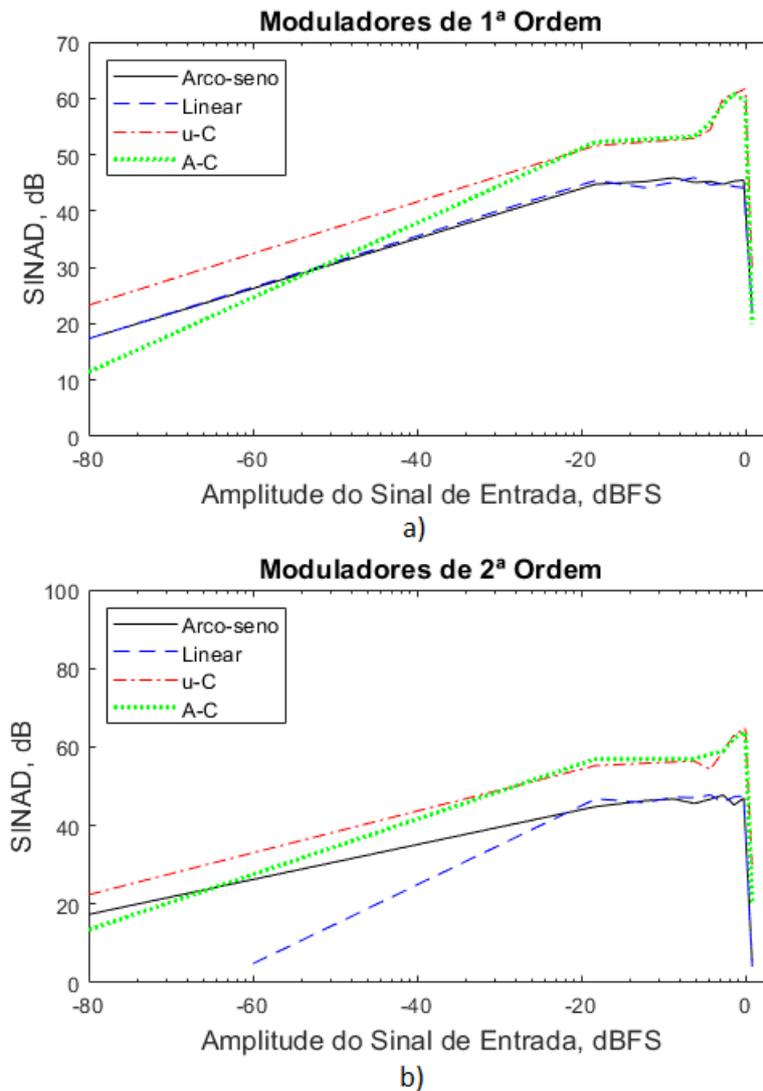
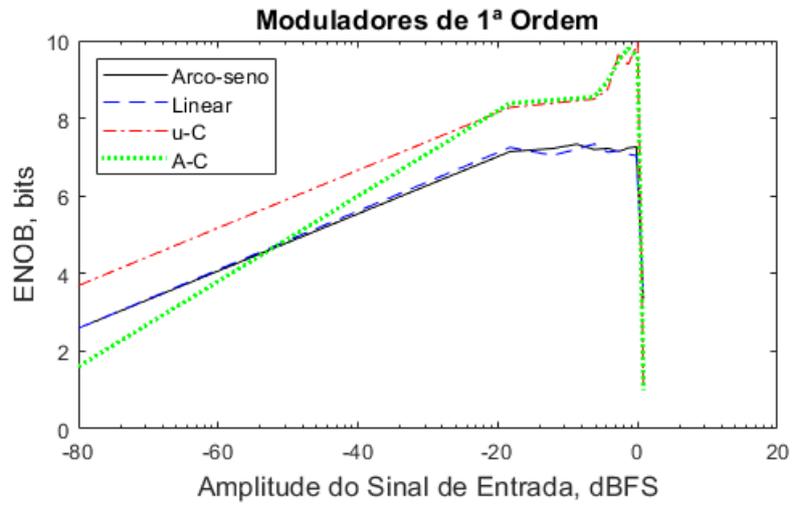
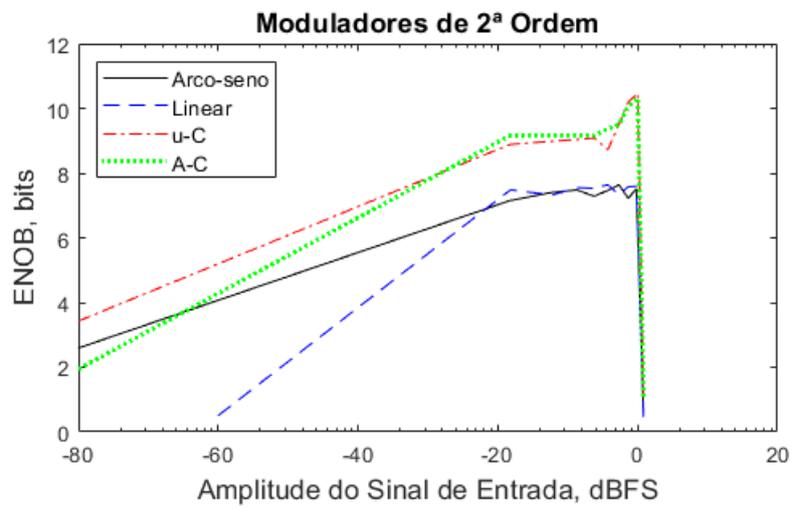


Figura 4.68 – Gráfico da *SINAD* para os moduladores $\Sigma\text{-}\Delta$ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador Linear, Arco-seno, $\mu\text{-C}$ e *A-C*.

A Figura 4.69 mostra os gráficos do número efetivo de bits (*ENOB*) para os moduladores projetados. Como pode-se observar, para ambas as ordens, os moduladores $\mu\text{-C}$ e *A-C* são os que apresentam o maior *ENOB* durante quase toda faixa dinâmica, sendo que o modulador $\mu\text{-C}$ é o que apresenta o melhor desempenho entre os sistemas. Para os sistemas de 1ª ordem ele apresenta uma melhoria de até 2.6 bits em relação ao modulador Arco-seno e de até 2.9 bits em relação ao modulador Linear. Para os sistemas de 2ª ordem, essa melhoria foi de até 2.9 bits em relação aos moduladores Arco-seno e Linear. Esses resultados foram alcançados devido a menor quantidade de ruído gerada por esses moduladores em comparação com os outros sistemas, dessa forma menos bits são perdidos por conta de ruído e distorção.



a)



b)

Figura 4.69 – Gráfico do *ENOB* para os moduladores Σ - Δ *multi-bit* de 1ª (a) e de 2ª (b) ordem com quantizador Linear, Arco-seno, μ -C e A-C.

5 CONCLUSÃO

No processo de conversão analógico-digital, a quantização é o processo onde o sinal analógico é substituído por amostras digitais aproximadas. Geralmente, os conversores utilizam quantizadores lineares, onde todos os níveis de quantização são iguais ao longo da faixa dinâmica. Para muitos tipos de sinais, no entanto, a quantização linear não é a mais eficiente podendo gerar grandes erros de quantização e conseqüentemente a perda de informação do sinal. Uma alternativa eficiente para resolver esse problema é a implementação de um sistema com quantização não linear, onde os níveis de quantização não são constantes ao longo da faixa dinâmica.

A presente dissertação, apresentou os conceitos fundamentais e diferentes técnicas para a implementação de quantizadores não lineares. A partir da fundamentação teórica, foram modelados três quantizadores não lineares e um quantizador linear, utilizado como referência para os outros sistemas. Todos os quantizadores foram projetados com 4-bits e utilizaram a topologia *flash* como base. O primeiro quantizador não linear, chamado de quantizador Arco-seno, foi projetado com base na Transformada da Incerteza. Esse quantizador, utilizou uma distribuição arco-seno para o mapeamento dos níveis de quantização. Essa topologia foi sugerida por [11] e foi modelada e analisada neste trabalho. Assim como o quantizador Linear, este quantizador teve todos os seus blocos modelados utilizando-se a *HDL Verilog-A*.

Os outros dois quantizadores não lineares foram projetados com base na técnica de *companding*. Esta técnica utilizou como base as leis logarítmicas, lei μ e lei A, para realizar a expansão e compressão do sinal entre o processo de quantização. Alternando-se a ordem do compressor e expensor e combinando os dois sistemas, conseguiu-se projetar quantizadores com uma resolução maior ao longo de quase toda a faixa dinâmica, são eles os quantizadores μ -C (baseado na lei μ) e A-C (baseado na lei A). Diferente dos outros dois quantizadores, os quantizadores logarítmicos foram modelados em alto nível utilizando-se o Matlab.

Na segunda parte dessa dissertação, realizou-se a modelagem e validação dos blocos necessários para a implementação de um modulador Σ - Δ *multi-bit* de 1ª de 2ª ordem. Esses moduladores foram projetados para a aplicação dos quantizadores desenvolvidos. A topologia Σ - Δ foi escolhida por se tratar de uma arquitetura conhecida, dessa forma foi possível analisar se os resultados obtidos estavam de acordo com o esperado pela teoria. Isso e o fato dela suportar um quantizador *multi-bit* a tornaram uma boa escolha para a implementação e teste dos quantizadores. Além disso, foram projetados moduladores de 1ª e de 2ª ordem para verificar a consistência dos resultados.

Após o projeto dos quantizadores, que foram aplicados nos moduladores Σ - Δ , realizou-

se a análise no tempo e na frequência dos quatro sistemas utilizando-se os mesmos parâmetros de simulação e análise. Com os resultados obtidos na análise no tempo, pode-se observar que o quantizador Linear apresentou passos de quantização constantes e o erro de quantização clássico dente de serra/sino. O quantizador Arco-seno, por outro lado, apresentou passos de quantização menores e conseqüentemente maior resolução, nas extremidades do sinal e passos de quantização maiores no centro da faixa dinâmica. Esse efeito gerou um erro de quantização modular que é menor nas extremidades do sinal.

Assim como o quantizador Arco-seno, os quantizadores μ -C e A-C também geraram passo de quantização variáveis ao longo da faixa dinâmica. Esses quantizadores apresentaram uma maior resolução nas extremidades e no centro da faixa dinâmica e uma menor resolução em $\pm 1/4$ da faixa dinâmica, que é onde ocorre o chaveamento dos quantizadores combinados que estão sendo utilizados. Comparados aos quantizadores Linear e Arco-seno, os quantizadores logarítmicos apresentaram menores passos de quantização durante a maior parte da faixa dinâmica, sugerindo assim um menor ruído de quantização. O erro de quantização dos dois sistemas também apresentou um comportamento modular.

Na análise em frequência dos quantizadores, já aplicados nos moduladores Σ - Δ , calculou-se as *DFTs* de todos os sistemas e a partir disso extraiu-se os parâmetros dinâmicos: *THD*, *ENOB*, *SNHR*, *SINAD* e *SFDR*. Comparando-se os espectros em frequência de todos os moduladores, pode-se observar que para os moduladores de 1ª ordem, o quantizador A-C foi o que gerou menor distorção harmônica (-27.78 dBs) pois ele apresentou harmônicos com menor potência principalmente perto da frequência fundamental. Essa melhoria na distorção foi de até 5.06 dBs em relação ao modulador Linear. Já nos sistemas de 2ª ordem, o modulador Arco-seno foi o que apresentou menor *THD* (-21.24 dBs), apresentando uma melhoria de até 2.34 dBs com relação ao modulador Linear. Também foi possível notar nos espectros, que o ruído de quantização para os moduladores μ -C e A-C foi bem menor em comparação aos outros dois, sendo assim eles apresentaram uma maior *SFDR* para quase toda faixa dinâmica, sendo que o modulador μ -C foi o que apresentou o melhor desempenho nesse parâmetro, alcançando uma melhoria de até 18.09 dBs em relação ao modulador Linear nos sistemas de 1ª ordem e de até 16.40 dBs nos sistemas de 2ª ordem.

Na análise da relação sinal-ruído sem distorção (*SNHR*), para os sistemas de 1ª ordem, o modulador Arco-seno foi o que gerou a maior *SNHR* apresentando uma melhoria de até 37.34 dBs em relação ao modulador Linear. Esse efeito pode ser observado no espectro em frequência pois o quantizador Arco-seno apresenta menor potência de ruído perto da frequência fundamental e para o cálculo da *SNHR* utilizou-se uma janela de 11 KHz como filtro. Já para os sistemas de 2ª ordem a *SNHR* permaneceu aproximadamente a mesma para os 4 sistemas devido ao *noise-shaping* maior causado pela 2ª ordem do modulador, sendo um pouco maior para os moduladores μ -C e A-C.

Na análise da relação sinal-ruído levando em conta também a distorção harmônica (*SI-*

NAD) o modulador μ -C foi o que obteve o melhor desempenho em relação aos outros sistemas. Nos sistemas de 1ª ordem, o modulador μ -C apresentou uma melhoria de até 17.33 dBs em relação ao modulador Linear e nos sistemas de 2ª ordem essa melhoria foi de até 17.69 dBs. Por fim na análise do número efetivo de bits (*ENOB*), o modulador μ -C foi o que apresentou um melhor desempenho para os sistemas de ambas as ordens, alcançando uma melhoria de até 2.9 bits em relação aos moduladores Lineares. Com a diminuição do ruído de quantização nos moduladores logarítmicos, menos bits foram corrompidos por conta do ruído, aumentando assim a resolução dos moduladores.

5.1 TRABALHOS FUTUROS

Os trabalhos futuros incluem primeiramente, a modelagem dos quantizadores μ -C e A-C utilizando-se a *HDL Verilog-A* com o objetivo de se realizar uma comparação de desempenho mais assertiva entre os quantizadores projetados. Após isso deverá ser feita a extração dos parâmetros estáticos de todos os quantizadores modelados para melhor caracterização dos sistemas.

É importante também o estudo sobre a análise de quantizadores não lineares uma vez que os parâmetros estáticos e dinâmicos apresentados têm como base a análise de conversores lineares e sendo assim, podem não incluir todos os efeitos necessários para a extração de parâmetros em sistemas não lineares.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 LATHI, B. P. *Modern digital and analog communication systems*. [S.l.]: Oxford University Press, Inc., 2009.
- 2 ALLEN, P.; HOLBERG, D. *CMOS Analog Circuit Design, vol. 2*. [S.l.]: Oxford: New York, 2002.
- 3 JOSÉ, M. Sigma-delta modulators: Tutorial overview, design guide, and state-of-the-art survey. *IEEE Transactions on Circuits and Systems I: Regular Papers*, IEEE, v. 58, n. 1, p. 1–21, 2010.
- 4 FLORES, M. d. G. C. C. Teste embarcado de conversores analógico-digitais. 2003.
- 5 MARIN, J. V. De la C. *INTEGRATED CIRCUIT DESIGN OF SIGMA-DELTA MODULATOR FOR ELECTRIC ENERGY MEASUREMENT APPLICATIONS*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2013.
- 6 CUBAS, A.; GROVER, H. *Projeto de um modulador sigma-delta de baixo consumo para sinais de áudio*. Tese (Doutorado) — Universidade de São Paulo, 2013.
- 7 BAKER, R. J. *CMOS: mixed-signal circuit design*. [S.l.]: John Wiley & Sons, 2008.
- 8 PARK, S. Principles of sigma-delta modulation for analog-to-digital converters. Motorola, 1999.
- 9 ALI, M. N.; ZOHDY, M. A. Interactive kalman filtering for differential and gaussian frequency shift keying modulation with application in bluetooth. *Journal of Signal and Information Processing*, Scientific Research Publishing, v. 3, n. 01, p. 63, 2012.
- 10 COSTA, L. L. et al. Nonlinear quantization technique for multibit sigma-delta modulators. In: IEEE. *2020 18th IEEE International New Circuits and Systems Conference (NEWCAS)*. [S.l.], 2020. p. 42–45.
- 11 MEDEIROS, J. E. G. d. *Unscented transform framework for quantization modeling in data conversion systems*. Tese (Doutorado) — Universidade de Brasília, 2017.
- 12 OGUNFUNMI, T.; NARASIMHA, M. *Principles of speech coding*. [S.l.]: CRC Press, 2010.
- 13 COSTA, L. L. O. Projeto de quantizadores não lineares para conversores analógico-digitais com base na transformada da incerteza. TCC – Universidade de Brasília, 2017.
- 14 SANTOS, M.; HORTA, N.; GUILHERME, J. A survey on nonlinear analog-to-digital converters. *Integration*, Elsevier, v. 47, n. 1, p. 12–22, 2014.
- 15 DAS, A. *Digital Communication: Principles and system modelling*. [S.l.]: Springer Science & Business Media, 2010.
- 16 KUNDERT, K.; CHANG, H. Top-down design and verification of mixed-signal circuits. *www.designers-guide.com*, 2005.

- 17 ZINKE, K. S. K. *The designer's guide to Verilog-AMS*. 1st edition. ed. [S.l.]: Kluwer Academic Publishers, 2004. (Designer's guide book series).
- 18 RAZAVI, B. *Principles of data conversion system design*. [S.l.]: IEEE press New York, 1995. v. 126.
- 19 TILDEN, S. J.; LINNENBRINK, T. E.; GREEN, P. J. Overview of ieee-std-1241 "standard for terminology and test methods for analog-to-digital converters". In: IEEE. *Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE*. [S.l.], 1999. v. 3, p. 1498–1503.
- 20 CUBAS, A.; GROVER, H. *Projeto de um modulador sigma-delta de baixo consumo para sinais de áudio*. Tese (Doutorado) — Universidade de São Paulo, 2013.
- 21 TILDEN, S. J.; LINNENBRINK, T. E.; GREEN, P. J. Overview of ieee-std-1241" standard for terminology and test methods for analog-to-digital converters". In: IEEE. *IMTC/99. Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference (Cat. No. 99CH36309)*. [S.l.], 1999. v. 3, p. 1498–1503.
- 22 BALESTRIERI, E.; RAPUANO, S. Defining dac performance in the frequency domain. *Measurement*, Elsevier, v. 40, n. 5, p. 463–472, 2007.
- 23 KESTER, W. Analog digital conversion. usa: Analog devices. *Inc*, v. 5, 2004.
- 24 SCHREIER, R.; TEMES, G. C. et al. *Understanding delta-sigma data converters*. [S.l.]: IEEE press Piscataway, NJ, 2005. v. 74.
- 25 LI, Y.; HE, L. First-order continuous-time sigma-delta modulator. In: IEEE. *Quality Electronic Design, 2007. ISQED'07. 8th International Symposium on*. [S.l.], 2007. p. 229–232.
- 26 MAEYER, J. D.; ROMBOUTS, P.; WEYTEN, L. Efficient multibit quantization in continuous-time *sigma-delta* modulators. *IEEE Transactions on Circuits and Systems I: Regular Papers*, IEEE, v. 54, n. 4, p. 757–767, 2007.
- 27 UHLMANN, J. K. *Dynamic map building and localization: New theoretical foundations*. Tese (Doutorado) — University of Oxford Oxford, 1995.
- 28 UHLMANN, J. K. *Simultaneous map building and localization for real time applications*. [S.l.], 1994.
- 29 MERWE, R. van der; DOUCET, A. N de freitas n and e wan, the unscented particle filter. *Adv. Neural Inform. Process. Syst*, 2000.
- 30 WANG, Y. Image coding standards: Jpeg and jpeg2000. *Multimedia Communication Systems*, Polytechnic University, Brooklyn, 2006.
- 31 BUCHANAN, B. *The Handbook of Data Communications and Networks: Volume 1. Volume 2*. [S.l.]: Springer Science & Business Media, 2010. v. 2.
- 32 MAGHARI, N.; TEMES, G.; MOON, U. Noise-shaped integrating quantisers in $\delta\sigma$ modulators. *Electronics letters*, IET, v. 45, n. 12, p. 612–613, 2009.

- 33 AJANYA, M.; VARGHESE, G. T. Thermometer code to binary code converter for flash adc-a review. In: IEEE. *2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCT)*. [S.l.], 2018. p. 502–505.
- 34 HARRIS, D. M. et al. Digital design and computer architecture. O'Reilly (ISBN 978-1-4919-0475-6), 2019.
- 35 DIAS, J. et al. Desenvolvimento e construção de sistema eletrônico indicador de limiar máximo de ruído sonoro em uti neonatal.

APÊNDICE

A APÊNDICES

A.1 CÓDIGOS EM VERILOG A

A.1.1 Comparador

```
1 // VerilogA for ADC_flash, decoder_16bits
2 // Autora : Laryssa Lorrany Olinda Costa
3 // Este programa foi inicialmente desenvolvido pela autora, em [13].
4 `include "constants.vams"
5 `include "disciplines.vams"
6
7 module comparator(in, ref, Clk, out, vdd, vss);
8   inout in, ref, Clk, out, vdd, vss;
9   parameter td =1f, tf =1f;
10  electrical in, ref, Clk, out, vdd, vss;
11  parameter real vth = 0;
12  parameter real dir = +1 from [-1:1] exclude 0;
13  real state;
14
15  analog begin
16    @(cross((V(Clk)-vth), +1) or initial_step) begin
17      if (V(in)>=V(ref))
18        state = V(vdd);
19      else
20        state= V(vss);
21    end
22    V(out) <+ transition(state, td, tf);
23  end
24
25 endmodule
```

A.1.2 Decoder

```
27 // VerilogA for ADC_flash, decoder_16bits
28 // Autora : Laryssa Lorrany Olinda Costa
29 // Este programa foi inicialmente desenvolvido pela autora, em [13].
30
31 `include "constants.vams"
32 `include "disciplines.vams"
33
34 module decoder_16bits(t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,
35   d1,d2,d3,d4, avdd, agnd);
36   inout t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,d1,d2,d3,d4,avdd
```

```

    , agnd;
36 electrical t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,d1,d2,d3,d4
    , avdd, agnd;
37 real tone, ttwo, tthree, tfour, tfive, tsix, tseven, teight, tnine, tten,
    televen, ttwelve, tthirteen, tfourteen, tfifteen;
38
39 analog begin
40 if ((V(d1) == V(agnd)) && (V(d2) == V(agnd)) && (V(d3) == V(agnd)) && (V(d4)
    == V(agnd)))
41 begin
42 tone = V(agnd);
43 ttwo= V(agnd);
44 tthree= V(agnd);
45 tfour= V(agnd);
46 tfive= V(agnd);
47 tsix= V(agnd);
48 tseven= V(agnd);
49 teight= V(agnd);
50 tnine= V(agnd);
51 tten= V(agnd);
52 televen= V(agnd);
53 ttwelve= V(agnd);
54 tthirteen= V(agnd);
55 tfourteen= V(agnd);
56 tfifteen= V(agnd);
57
58 end
59 if ((V(d1) == V(agnd)) && (V(d2) == V(agnd)) && (V(d3) == V(agnd)) && (V(d4)
    == V(avdd)))
60 begin
61 tone = V(agnd);
62 ttwo= V(agnd);
63 tthree= V(agnd);
64 tfour= V(agnd);
65 tfive= V(agnd);
66 tsix= V(agnd);
67 tseven= V(agnd);
68 teight= V(agnd);
69 tnine= V(agnd);
70 tten= V(agnd);
71 televen= V(agnd);
72 ttwelve= V(agnd);
73 tthirteen= V(agnd);
74 tfourteen= V(agnd);
75 tfifteen= V(avdd);
76
77 end

```

```

78  if ((V(d1) == V(agnd)) && (V(d2) == V(agnd)) && (V(d3) == V(avdd)) && (V(d4)
    == V(agnd)))
79  begin
80  tone =    V(agnd);
81  ttwo=    V(agnd);
82  tthree=  V(agnd);
83  tfour=   V(agnd);
84  tfive=   V(agnd);
85  tsix=    V(agnd);
86  tseven=  V(agnd);
87  teight=  V(agnd);
88  tnine=   V(agnd);
89  tten=    V(agnd);
90  televen= V(agnd);
91  ttwelve= V(agnd);
92  tthirteen= V(agnd);
93  tfourteen= V(avdd);
94  tfifteen= V(agnd);
95
96  end
97  if ((V(d1) == V(agnd)) && (V(d2) == V(agnd)) && (V(d3) == V(avdd)) && (V(d4)
    == V(avdd)))
98  begin
99  tone =    V(agnd);
100  ttwo=    V(agnd);
101  tthree=  V(agnd);
102  tfour=   V(agnd);
103  tfive=   V(agnd);
104  tsix=    V(agnd);
105  tseven=  V(agnd);
106  teight=  V(agnd);
107  tnine=   V(agnd);
108  tten=    V(agnd);
109  televen= V(agnd);
110  ttwelve= V(agnd);
111  tthirteen= V(avdd);
112  tfourteen= V(agnd);
113  tfifteen= V(agnd);
114
115  end
116  if ((V(d1) == V(agnd)) && (V(d2) == V(avdd)) && (V(d3) == V(agnd)) && (V(d4)
    == V(agnd)))
117  begin
118  tone =    V(agnd);
119  ttwo=    V(agnd);
120  tthree=  V(agnd);
121  tfour=   V(agnd);

```

```

122  tfive=    V(agnd);
123  tsix=    V(agnd);
124  tseven=  V(agnd);
125  teight=  V(agnd);
126  tnine=   V(agnd);
127  tten=    V(agnd);
128  televen= V(agnd);
129  ttwelve= V(avdd);
130  tthirteen= V(agnd);
131  tfourteen= V(agnd);
132  tfifteen= V(agnd);
133
134  end
135  if ((V(d1) == V(agnd)) && (V(d2) == V(avdd)) && (V(d3) == V(agnd)) && (V(d4)
    == V(avdd)))
136  begin
137  tone =    V(agnd);
138  ttwo=    V(agnd);
139  tthree=  V(agnd);
140  tfour=   V(agnd);
141  tfive=   V(agnd);
142  tsix=    V(agnd);
143  tseven=  V(agnd);
144  teight=  V(agnd);
145  tnine=   V(agnd);
146  tten=    V(agnd);
147  televen= V(avdd);
148  ttwelve= V(agnd);
149  tthirteen= V(agnd);
150  tfourteen= V(agnd);
151  tfifteen= V(agnd);
152
153  end
154  if ((V(d1) == V(agnd)) && (V(d2) == V(avdd)) && (V(d3) == V(avdd)) && (V(d4)
    == V(agnd)))
155  begin
156  tone =    V(agnd);
157  ttwo=    V(agnd);
158  tthree=  V(agnd);
159  tfour=   V(agnd);
160  tfive=   V(agnd);
161  tsix=    V(agnd);
162  tseven=  V(agnd);
163  teight=  V(agnd);
164  tnine=   V(agnd);
165  tten=    V(avdd);
166  televen= V(agnd);

```

```

167  ttwelve=  V(agnd);
168  tthirteen=  V(agnd);
169  tfourteen=  V(agnd);
170  tfifteen=  V(agnd);
171
172  end
173  if ((V(d1) == V(agnd)) && (V(d2) == V(avdd)) && (V(d3) == V(avdd)) && (V(d4)
    == V(avdd)))
174  begin
175  tone =  V(agnd);
176  ttwo=  V(agnd);
177  tthree=  V(agnd);
178  tfour=  V(agnd);
179  tfive=  V(agnd);
180  tsix=  V(agnd);
181  tseven=  V(agnd);
182  teight=  V(agnd);
183  tnine=  V(avdd);
184  tten=  V(agnd);
185  televen=  V(agnd);
186  ttwelve=  V(agnd);
187  tthirteen=  V(agnd);
188  tfourteen=  V(agnd);
189  tfifteen=  V(agnd);
190
191  end
192  if ((V(d1) == V(avdd)) && (V(d2) == V(agnd)) && (V(d3) == V(agnd)) && (V(d4)
    == V(agnd)))
193  begin
194  tone =  V(agnd);
195  ttwo=  V(agnd);
196  tthree=  V(agnd);
197  tfour=  V(agnd);
198  tfive=  V(agnd);
199  tsix=  V(agnd);
200  tseven=  V(agnd);
201  teight=  V(avdd);
202  tnine=  V(agnd);
203  tten=  V(agnd);
204  televen=  V(agnd);
205  ttwelve=  V(agnd);
206  tthirteen=  V(agnd);
207  tfourteen=  V(agnd);
208  tfifteen=  V(agnd);
209
210  end
211  if ((V(d1) == V(avdd)) && (V(d2) == V(agnd)) && (V(d3) == V(agnd)) && (V(d4)

```

```

    == V(avdd))
212 begin
213     tone =     V(agnd);
214     ttwo=     V(agnd);
215     tthree=   V(agnd);
216     tfour=    V(agnd);
217     tfive=    V(agnd);
218     tsix=     V(agnd);
219     tseven=   V(avdd);
220     teight=   V(agnd);
221     tnine=    V(agnd);
222     tten=     V(agnd);
223     televen=  V(agnd);
224     ttwelve=  V(agnd);
225     tthirteen= V(agnd);
226     tfourteen= V(agnd);
227     tfifteen= V(agnd);
228
229 end
230 if ((V(d1) == V(avdd)) && (V(d2) == V(agnd)) && (V(d3) == V(avdd)) && (V(d4)
    == V(agnd)))
231 begin
232     tone =     V(agnd);
233     ttwo=     V(agnd);
234     tthree=   V(agnd);
235     tfour=    V(agnd);
236     tfive=    V(agnd);
237     tsix=     V(avdd);
238     tseven=   V(agnd);
239     teight=   V(agnd);
240     tnine=    V(agnd);
241     tten=     V(agnd);
242     televen=  V(agnd);
243     ttwelve=  V(agnd);
244     tthirteen= V(agnd);
245     tfourteen= V(agnd);
246     tfifteen= V(agnd);
247
248 end
249 if ((V(d1) == V(avdd)) && (V(d2) == V(agnd)) && (V(d3) == V(avdd)) && (V(d4)
    == V(avdd)))
250 begin
251     tone =     V(agnd);
252     ttwo=     V(agnd);
253     tthree=   V(agnd);
254     tfour=    V(agnd);
255     tfive=    V(avdd);

```

```

256  tsix=    V(agnd);
257  tseven=  V(agnd);
258  teight=  V(agnd);
259  tnine=   V(agnd);
260  tten=    V(agnd);
261  televen= V(agnd);
262  ttwelve= V(agnd);
263  tthirteen= V(agnd);
264  tfourteen= V(agnd);
265  tfifteen= V(agnd);
266
267  end
268  if ((V(d1) == V(avdd)) && (V(d2) == V(avdd)) && (V(d3) == V(agnd)) && (V(d4)
    == V(agnd)))
269  begin
270  tone =    V(agnd);
271  ttwo=    V(agnd);
272  tthree=  V(agnd);
273  tfour=   V(avdd);
274  tfive=   V(agnd);
275  tsix=    V(agnd);
276  tseven=  V(agnd);
277  teight=  V(agnd);
278  tnine=   V(agnd);
279  tten=    V(agnd);
280  televen= V(agnd);
281  ttwelve= V(agnd);
282  tthirteen= V(agnd);
283  tfourteen= V(agnd);
284  tfifteen= V(agnd);
285
286  end
287  if ((V(d1) == V(avdd)) && (V(d2) == V(avdd)) && (V(d3) == V(agnd)) && (V(d4)
    ==V(avdd)))
288  begin
289  tone =    V(agnd);
290  ttwo=    V(agnd);
291  tthree=  V(avdd);
292  tfour=   V(agnd);
293  tfive=   V(agnd);
294  tsix=    V(agnd);
295  tseven=  V(agnd);
296  teight=  V(agnd);
297  tnine=   V(agnd);
298  tten=    V(agnd);
299  televen= V(agnd);
300  ttwelve= V(agnd);

```

```

301  tthirteen= V(agnd);
302  tfourteen= V(agnd);
303  tfifteen= V(agnd);
304
305  end
306  if ((V(d1) == V(avdd)) && (V(d2) == V(avdd)) && (V(d3) == V(avdd)) && (V(d4)
    == V(agnd)))
307  begin
308  tone = V(agnd);
309  ttwo= V(avdd);
310  tthree= V(agnd);
311  tfour= V(agnd);
312  tfive= V(agnd);
313  tsix= V(agnd);
314  tseven= V(agnd);
315  teight= V(agnd);
316  tnine= V(agnd);
317  tten= V(agnd);
318  televen= V(agnd);
319  ttwelve= V(agnd);
320  tthirteen= V(agnd);
321  tfourteen= V(agnd);
322  tfifteen= V(agnd);
323
324  end
325  if ((V(d1) == V(avdd)) && (V(d2) == V(avdd)) && (V(d3) == V(avdd)) && (V(d4)
    == V(avdd)))
326  begin
327  tone = V(avdd);
328  ttwo= V(agnd);
329  tthree= V(agnd);
330  tfour= V(agnd);
331  tfive= V(agnd);
332  tsix= V(agnd);
333  tseven= V(agnd);
334  teight= V(agnd);
335  tnine= V(agnd);
336  tten= V(agnd);
337  televen= V(agnd);
338  ttwelve= V(agnd);
339  tthirteen= V(agnd);
340  tfourteen= V(agnd);
341  tfifteen= V(agnd);
342
343  end
344  V(t1)<+ transition(tone,1f,1f);
345  V(t2)<+ transition(ttwo,1f,1f);

```

```

346 V(t3)<+ transition(tthree,1f,1f);
347 V(t4)<+ transition(tfour,1f,1f);
348 V(t5)<+ transition(tfive,1f,1f);
349 V(t6)<+ transition(tsix,1f,1f);
350 V(t7)<+ transition(tseven,1f,1f);
351 V(t8)<+ transition(teight,1f,1f);
352 V(t9)<+ transition(tnine,1f,1f);
353 V(t10)<+ transition(tten,1f,1f);
354 V(t11)<+ transition(televen,1f,1f);
355 V(t12)<+ transition(ttwelve,1f,1f);
356 V(t13)<+ transition(tthirteen,1f,1f);
357 V(t14)<+ transition(tfourteen,1f,1f);
358 V(t15)<+ transition(tfifteen,1f,1f);
359
360 end
361 endmodule

```

A.1.3 Encoder

```

362 // VerilogA for ADC_flash, therm-to-binary-4bits
363 // Autora: Laryssa Lorrany Olinda Costa
364 // Esse programa foi inicialmente desenvolvido pela autora, em [13].
365
366 `include "constants.vams"
367 `include "disciplines.vams"
368
369 module therm4bits(t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,d1,
    d2,d3,d4,avdd,agnd);
370 inout t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,d1,d2,d3,d4,avdd
    ,agnd;
371 parameter td =1f, tf =1f;
372 electrical t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,d1,d2,d3,d4
    ,avdd,agnd;
373 real done, dtwo, dthree, dfour;
374
375 analog begin
376
377
378     if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
        ) == V(agnd)) && (V(t5) == V(agnd)) && (V(t6) == V(agnd)) && (V(t7) == V(
        agnd)) && (V(t8) == V(agnd)) && (V(t9) == V(agnd)) && (V(t10) == V(agnd)) && (
        V(t11) == V(agnd)) && (V(t12) == V(agnd)) && (V(t13) == V(agnd)) && (V(t14)
        == V(agnd)) && (V(t15) == V(agnd)))
379         begin
380             done = V(agnd);
381             dtwo = V(agnd);
382             dthree = V(agnd);

```

```

383     dfour= V(agnd);
384 end
385
386
387 if ((V(t1) == V(agnd))&&(V(t2) == V(agnd))&&(V(t3) == V(agnd))&&(V(t4)
    == V(agnd))&&(V(t5) == V(agnd))&&(V(t6) == V(agnd))&&(V(t7) == V(agnd)
) &&(V(t8) == V(agnd))&&(V(t9) == V(agnd))&&(V(t10) == V(agnd))&&(V(t11
) == V(agnd))&&(V(t12) == V(agnd))&&(V(t13) == V(agnd))&&(V(t14) == V(
agnd))&&(V(t15) == V(avdd)))
388     begin
389         done = V(agnd);
390         dtwo = V(agnd);
391         dthree = V(agnd);
392         dfour= V(avdd);
393 end
394
395 if ((V(t1) == V(agnd))&&(V(t2) == V(agnd))&&(V(t3) == V(agnd))&&(V(t4)
    == V(agnd))&&(V(t5) == V(agnd))&&(V(t6) == V(agnd))&&(V(t7) == V(agnd)
) &&(V(t8) == V(agnd))&&(V(t9) == V(agnd))&&(V(t10) == V(agnd))&&(V(
t11) == V(agnd))&&(V(t12) == V(agnd))&&(V(t13) == V(agnd))&&(V(t14) ==
V(avdd))&&(V(t15) == V(avdd)))
396     begin
397         done = V(agnd);
398         dtwo = V(agnd);
399         dthree = V(avdd);
400         dfour= V(agnd);
401     end
402
403 if ((V(t1) == V(agnd))&&(V(t2) == V(agnd))&&(V(t3) == V(agnd))&&(V(t4)
    == V(agnd))&&(V(t5) == V(agnd))&&(V(t6) == V(agnd))&&(V(t7) == V(agnd)
) &&(V(t8) == V(agnd))&&(V(t9) == V(agnd))&&(V(t10) == V(agnd))&&(V(t11
) == V(agnd))&&(V(t12) == V(agnd))&&(V(t13) == V(avdd))&&(V(t14) == V(
avdd))&&(V(t15) == V(avdd)))
404     begin
405         done = V(agnd);
406         dtwo = V(agnd);
407         dthree = V(avdd);
408         dfour= V(avdd);
409     end
410 if ((V(t1) == V(agnd))&&(V(t2) == V(agnd))&&(V(t3) == V(agnd))&&(V(t4)
    == V(agnd))&&(V(t5) == V(agnd))&&(V(t6) == V(agnd))&&(V(t7) == V(agnd)
) &&(V(t8) == V(agnd))&&(V(t9) == V(agnd))&&(V(t10) == V(agnd))&&(V(t11
) == V(agnd))&&(V(t12) == V(avdd))&&(V(t13) == V(avdd))&&(V(t14) == V(
avdd))&&(V(t15) == V(avdd)))
411     begin
412         done = V(agnd);
413         dtwo = V(avdd);

```

```

414     dthree = V(agnd);
415     dfour= V(agnd);
416     end
417     if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
    == V(agnd)) && (V(t5) == V(agnd)) && (V(t6) == V(agnd)) && (V(t7) == V(agnd)
    ) && (V(t8) == V(agnd)) && (V(t9) == V(agnd)) && (V(t10) == V(agnd)) && (V(t11
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
418         begin
419             done = V(agnd);
420             dtwo = V(avdd);
421             dthree = V(agnd);
422             dfour= V(avdd);
423             end
424             if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
    == V(agnd)) && (V(t5) == V(agnd)) && (V(t6) == V(agnd)) && (V(t7) == V(agnd)
    )) && (V(t8) == V(agnd)) && (V(t9) == V(agnd)) && (V(t10) == V(avdd)) && (V(
    t11) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) ==
    V(avdd)) && (V(t15) == V(avdd)))
425                 begin
426                     done = V(agnd);
427                     dtwo = V(avdd);
428                     dthree = V(avdd);
429                     dfour= V(agnd);
430                     end
431
432             if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
    == V(agnd)) && (V(t5) == V(agnd)) && (V(t6) == V(agnd)) && (V(t7) == V(agnd)
    ) && (V(t8) == V(agnd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
433                 begin
434                     done = V(agnd);
435                     dtwo = V(avdd);
436                     dthree = V(avdd);
437                     dfour= V(avdd);
438                 end
439             if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
    == V(agnd)) && (V(t5) == V(agnd)) && (V(t6) == V(agnd)) && (V(t7) == V(agnd)
    ) && (V(t8) == V(avdd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
440                 begin
441                     done = V(avdd);
442                     dtwo = V(agnd);
443                     dthree = V(agnd);
444                     dfour= V(agnd);

```

```

445     end
446     if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
    == V(agnd)) && (V(t5) == V(agnd)) && (V(t6) == V(agnd)) && (V(t7) == V(avdd)
    ) && (V(t8) == V(avdd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
447         begin
448             done = V(avdd);
449             dtwo = V(agnd);
450             dthree = V(agnd);
451             dfour= V(avdd);
452         end
453     if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
    == V(agnd)) && (V(t5) == V(agnd)) && (V(t6) == V(avdd)) && (V(t7) == V(avdd)
    ) && (V(t8) == V(avdd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
454         begin
455             done = V(avdd);
456             dtwo = V(agnd);
457             dthree = V(avdd);
458             dfour= V(agnd);
459         end
460
461     if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
    == V(agnd)) && (V(t5) == V(avdd)) && (V(t6) == V(avdd)) && (V(t7) == V(avdd)
    ) && (V(t8) == V(avdd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
462         begin
463             done = V(avdd);
464             dtwo = V(agnd);
465             dthree = V(avdd);
466             dfour= V(avdd);
467         end
468
469     if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(agnd)) && (V(t4)
    == V(avdd)) && (V(t5) == V(avdd)) && (V(t6) == V(avdd)) && (V(t7) == V(avdd)
    ) && (V(t8) == V(avdd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
470         begin
471             done = V(avdd);
472             dtwo = V(avdd);
473             dthree = V(agnd);
474             dfour= V(agnd);
475         end

```

```

476  if ((V(t1) == V(agnd)) && (V(t2) == V(agnd)) && (V(t3) == V(avdd)) && (V(t4)
    == V(avdd)) && (V(t5) == V(avdd)) && (V(t6) == V(avdd)) && (V(t7) == V(avdd)
    ) && (V(t8) == V(avdd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11)
    == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
477      begin
478          done = V(avdd);
479          dtwo = V(avdd);
480          dthree = V(agnd);
481          dfour= V(avdd);
482      end
483  if ((V(t1) == V(agnd)) && (V(t2) == V(avdd)) && (V(t3) == V(avdd)) && (V(t4)
    == V(avdd)) && (V(t5) == V(avdd)) && (V(t6) == V(avdd)) && (V(t7) == V(avdd)
    ) && (V(t8) == V(avdd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11)
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
484      begin
485          done = V(avdd);
486          dtwo = V(avdd);
487          dthree = V(avdd);
488          dfour= V(agnd);
489      end
490  if ((V(t1) == V(avdd)) && (V(t2) == V(avdd)) && (V(t3) == V(avdd)) && (V(t4)
    == V(avdd)) && (V(t5) == V(avdd)) && (V(t6) == V(avdd)) && (V(t7) == V(avdd)
    ) && (V(t8) == V(avdd)) && (V(t9) == V(avdd)) && (V(t10) == V(avdd)) && (V(t11)
    ) == V(avdd)) && (V(t12) == V(avdd)) && (V(t13) == V(avdd)) && (V(t14) == V(
    avdd)) && (V(t15) == V(avdd)))
491      begin
492          done = V(avdd);
493          dtwo = V(avdd);
494          dthree = V(avdd);
495          dfour= V(avdd);
496      end
497
498  V(d1)<+ transition(done,td,tf);
499  V(d2)<+ transition(dtwo,td,tf);
500  V(d3)<+ transition(dthree,td,tf);
501  V(d4)<+ transition(dfour,td,tf);
502 end
503
504 endmodule

```

A.1.4 Chave

```

506 // VerilogA for Relay
507 // Autora : Laryssa Lorrany Olinda Costa
508 // Este programa foi inicialmente desenvolvido pela autora, em [13].

```

```

509
510
511 `include " constants . vams "
512 `include "disciplines.vams"
513
514 module relay (p, n, ps, ns);
515 parameter real thresh=0; // threshold
516 output p, n;
517 input ps, ns; // contacts
518 electrical p, n, ps, ns;
519
520 analog begin
521
522     if (V(ps,ns) > thresh)
523         V(p,n) <+ 0;
524     else
525         I(p,n) <+ 0;
526     end
527
528 endmodule

```

A.2 MODULADOR Σ - Δ *MULTI-BIT*

A.2.1 Integrador

```

529 // VerilogA for Modulador-SD, integrator
530 // Autora : Laryssa Lorrany Olinda Costa
531 // Este programa foi inicialmente desenvolvido pela autora, em [13].
532
533 `include "constants.vams"
534 `include "disciplines.vams"
535
536 module integrador(in, out, Clk);
537     inout in, out, Clk;
538     electrical in, out, Clk;
539     parameter real scale = 1000000;
540     parameter real vref = 0.5;
541     parameter real vth = 0;
542     real state;
543     analog begin
544         @(cross((V(Clk)-vth), +1)) begin
545             state = V(in);
546         end
547         V(out) <+ idt(scale*(state-vref), 0.5);
548     end
549
550 endmodule

```

A.2.2 Subtrator

```
552
553 // VerilogA for Modulador-SD, subtrator
554 // Autora : Laryssa Lorrany Olinda Costa
555 // Este programa foi inicialmente desenvolvido pela autora, em [13].
556
557 `include "constants.vams"
558 `include "disciplines.vams"
559
560 module subtrator(in1,in2,out);
561
562     inout in1,in2,out;
563     electrical in1,in2,out;
564     analog begin
565         V(out) <+ (V(in1)-V(in2));
566     end
567 endmodule
```

A.2.3 Amplificador

```
569
570 // VerilogA for Modulador-SD, amplifier
571 // Autora : Laryssa Lorrany Olinda Costa
572 // Este programa foi inicialmente desenvolvido pela autora, em [13].
573
574 `include "constants.vams"
575 `include "disciplines.vams"
576
577 module amplificador(in, out);
578     parameter real gain = 5;
579     inout in, out;
580     electrical in, out;
581
582     analog
583         V(out) <+ gain*V(in);
584 endmodule
```

B ANEXOS

B.1 QUANTIZADORES

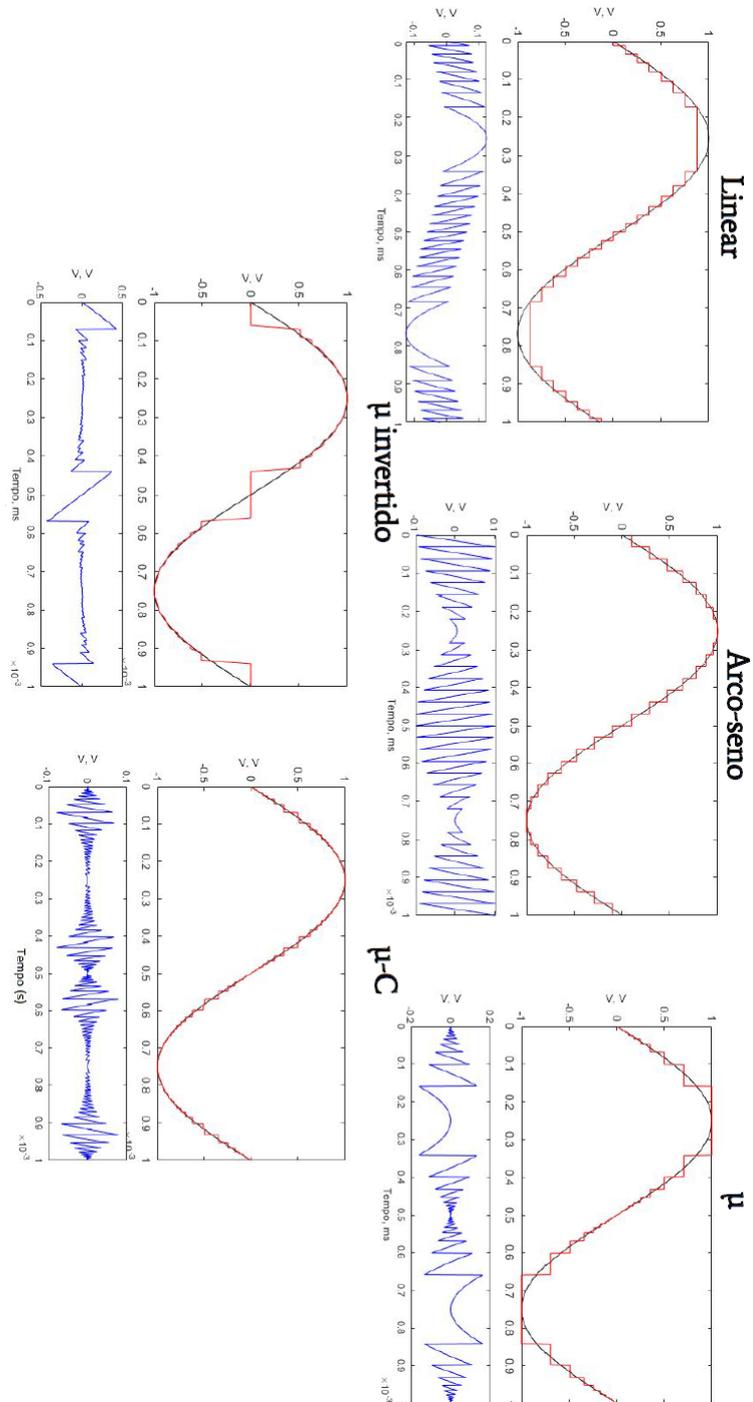


Figura B.1 – Respostas dos quantizadores.

B.2 MODULADORES Σ - Δ MULTI-BIT

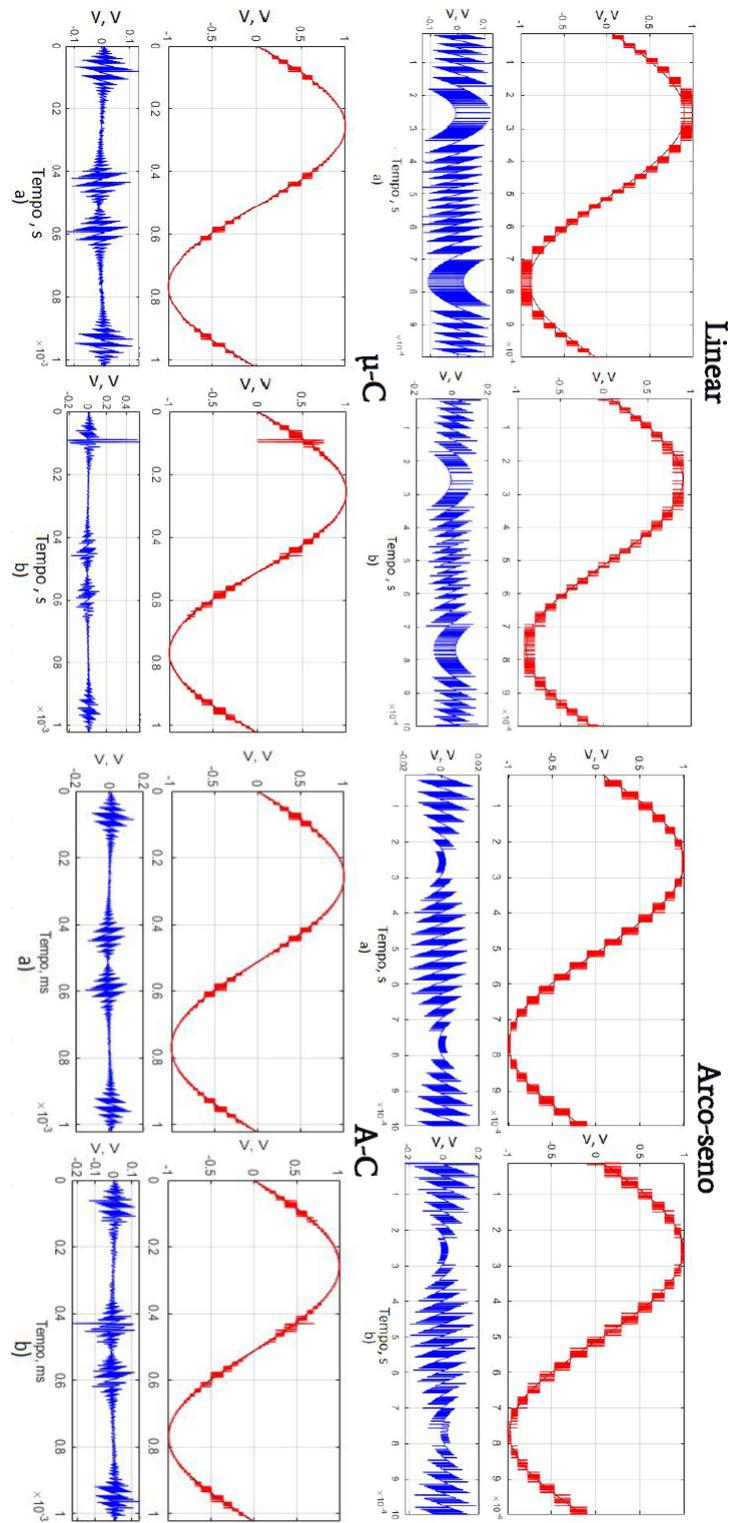


Figura B.2 – Respostas dos moduladores Σ - Δ multi-bit de 1^a (a) e de 2^a (b) ordem.