

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E  
INFORMÁTICA INDUSTRIAL

MAICON BRUNO HOFMANN

**Implementação em FPGA de compensadores de desvios para Conversor  
Analógico Digital Intercalado**

DISSERTAÇÃO

CURITIBA

2016

MAICON BRUNO HOFMANN

**Implementação em FPGA de compensadores de desvios para Conversor  
Analógico Digital Intercalado**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Ciências” – Área de Concentração: Engenharia de Automação e Sistemas.

Orientador: Prof. Ph.D. André Augusto  
Mariano

Co-orientadora: Prof<sup>a</sup>. Dra. Sibilla Batista da Luz  
França

**CURITIBA**

**2016**

Dados Internacionais de Catalogação na Publicação

---

H713i  
2016 Hofmann, Maicon Bruno  
Implementação em FPGA de compensadores de desvios para  
conversor analógico digital intercalado / Maicon Bruno Hofmann.-  
- 2016.  
98 f. : il. ; 30 cm.

Dissertação (Mestrado) - Universidade Tecnológica Federal  
do Paraná. Programa de Pós-graduação em Engenharia Elétrica  
e Informática Industrial, Curitiba, 2016  
Bibliografia: f. 94-97

1. Conversores analógicos digitais. 2. Arranjos de lógica pro-  
gramável em campo. 3. VHDL (Linguagem descritiva de hardwa-  
re). 4. Engenharia elétrica - Dissertações. I. Mariano, André Au-  
gusto, orient. II. França, Sibilla Batista da Luz, coorient. III. Uni-  
versidade Tecnológica Federal do Paraná. Programa de Pós-  
Graduação em Engenharia Elétrica e Informática Industrial. IV.  
Título.

---

CDD: Ed. 22 -- 621.3

Biblioteca Central da UTFPR, Câmpus Curitiba

Título da Dissertação Nº. \_\_\_\_\_

## **Implementação Em FPGA De Compensadores De Desvios Para Conversor Analógico Digital Intercalado**

por

**Maicon Bruno Hofmann**

**Orientador: Prof. Dr. André Augusto Mariano (UTFPR)**

**Coorientadora: Prof<sup>a</sup>. Dr. Sibilla Batista da Luz França(UFPR)**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM CIÊNCIAS – Área de Concentração: **Engenharia de Automação e Sistemas** do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR, às 14 h do dia 15 de março 2016. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores doutores:

---

Prof. Dr. André Augusto Mariano  
(Presidente – UTFPR)

---

Prof. Dr. Luis Henrique Assumpção Lolis  
(UFPR)

---

Prof. Dr. Glauber Gomes de Oliveira Brante  
(UTFPR)

Visto da coordenação:

---

Prof. Dr. Emilio Carlos Gomes Wille  
(Coordenador do CPGEI)

*Dedico este trabalho a minha amada Suélin, que esteve ao meu lado desde o início desta jornada, me incentivando, apoiando e comemorando comigo todas as conquistas; mas principalmente por compreender os motivos de minha ausência durante todo este período.*

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus e a minha família.

Agradeço ao meu orientador e co-orientadora, pela paciência e compreensão ao longo destes anos, principalmente nos momentos em que cometi falhas. Sinto-me lisonjeado pela oportunidade que tive de receber seus ensinamentos e orientação.

Agradeço aos professores da UFPR e aos alunos do grupo GICS, pelas valiosas dicas e observações dadas durante as minhas apresentações no grupo. Em especial ao Raphael, pelo tempo que disponibilizou para me ensinar os métodos de teste utilizados pelo Allan; e ao professor Lima, pelas conversas de assuntos aleatórios e companheirismo durante o SBRT.

Agradeço aos membros da banca, professor Glauber Rocha e Luis Lolis, pelas contribuições ao texto final deste trabalho.

Agradeço aos meus professores de graduação na UDESC, Aleksander Paterno e Pedro Bertemes, que me deram o primeiro incentivo para a pesquisa acadêmica.

E por último, mas não menos importante, a minha noiva, pelo incentivo durante todo o desenvolvimento deste trabalho, e por tirar a minha atenção dele de vez em quando.

*Todos podem ver as táticas de minhas conquistas, mas ninguém consegue discernir a estratégia que gerou as vitórias.*

Sun Tzu

## RESUMO

HOFMANN, Maicon. Implementação em FPGA de compensadores de desvios para Conversor Analógico Digital Intercalado. 99 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Este trabalho apresenta a modelagem e implementação em FPGA de sistemas digitais de compensação de desvios para TIADC (*Time Interleaved Analog to Digital Converter*). O desenvolvimento de todo este trabalho seguiu uma metodologia *top-down*. Seguindo esta metodologia foi elaborada a modelagem comportamental de um TIADC de dois canais e seus respectivos desvios de *offset*, ganho e *clock skew* em Simulink. A modelagem comportamental de sistemas digitais de compensação para estes desvios também foi realizada seguindo esta metodologia. Para o desvio de *clock skew* foi utilizada a compensação através de filtros de *delay* fracionário, mais especificamente, a eficiente estrutura de Farrow. A definição de qual método seria utilizado para o projeto do filtro, e da estrutura de Farrow, exigiu um estudo de diversos métodos de projeto apresentados na literatura. Os sistemas digitais de compensação modelados foram convertidos em código VHDL, para implementação e validação em FPGA. A validação destes sistemas foi realizada utilizando a metodologia de teste *FPGA In Loop*. Os resultados obtidos com os compensadores de desvio do TIADC demonstram o elevado ganho de desempenho fornecido por estas estruturas. Além deste resultado, este trabalho ilustra o potencial das metodologias de desenvolvimento, implementação e teste em FPGA utilizadas para a obtenção destes compensadores.

**Palavras-chave:** Conversor analógico digital intercalado, Farrow, Clock Skew, FPGA, Modelagem Simulink



## ABSTRACT

HOFMANN, Maicon. FPGA IMPLEMENTATION OF TIME INTERLEAVED ANALOG TO DIGITAL CONVERTER MISMATCHES COMPENSATORS. 99 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

This work presents the modeling and FPGA implementation of digital TIADC mismatches compensation systems in a top-down methodology. A two channel TIADC and their respective offset, gain and clock skew mismatches behavior modeling was done on Simulink. In addition, was developed digital mismatch compensation system behavior modeling. Fractional delay filters were used for clock skew mismatch compensation, more specifically, the efficient Farrow structure. To define the filter design, and Farrow structure design, methodology a study of various design methods present in literature were required. The digital compensation systems models were converted to VHDL, for FPGA implementation and validation. These system validation was carried out using the test methodology FPGA In Loop. The results obtained show the high performance gain provided by these digital systems. Beyond this result, these work illustrates the potential of design, implementation and FPGA test methodologies.

**Keywords:** Time Interleaved ADC, Farrow, Clock Skew, FPGA, Simulink Modeling

## LISTA DE FIGURAS

FIGURA 1	– Função de transferência de um ADC ideal .....	20
FIGURA 2	– SFDR .....	22
FIGURA 3	– TIADC .....	23
FIGURA 4	– Função de transferência de ADC com <i>offset</i> .....	25
FIGURA 5	– Espectro TIADC com <i>offset</i> .....	25
FIGURA 6	– Função de transferência de um ADC com desvio no ganho .....	26
FIGURA 7	– Espectro TIADC com desvio de ganho .....	26
FIGURA 8	– Aperture Delay .....	27
FIGURA 9	– Espectro do TIADC com <i>clock skew</i> .....	28
FIGURA 10	– SNR versus $f_{in}$ de um TIADC com <i>clock skew</i> .....	29
FIGURA 11	– Resposta ao impulso filtro de <i>delay</i> fracionário ideal .....	36
FIGURA 12	– Magnitude do filtro projetado através do método Least Square .....	38
FIGURA 13	– Atraso de fase do filtro projetado através do método Least Square .....	38
FIGURA 14	– Magnitude de filtros projetados utilizando método GLS para $\alpha = 0,5$ ...	41
FIGURA 15	– Atraso de fase de filtros projetados utilizando método GLS para $\alpha = 0,5$ .	41
FIGURA 16	– Magnitude de filtros projetados utilizando método Minimax para banda de $0,5\pi$ .....	44
FIGURA 17	– Atraso de fase de filtros projetados utilizando método Minimax para banda de $0,5\pi$ . .....	44
FIGURA 18	– Magnitude de filtros projetados utilizando Lagrange .....	45
FIGURA 19	– Atraso de fase de filtros projetados utilizando Lagrange. ....	46
FIGURA 20	– Estrutura de Farrow .....	48
FIGURA 21	– Visão geral da modelagem do TIADC .....	53
FIGURA 22	– Modelo interno do ADC .....	54
FIGURA 23	– Estrutura geral dos compensadores de desvio .....	55
FIGURA 24	– Modelo teste da ordem do filtro .....	57
FIGURA 25	– Modelo de teste da estrutura de Farrow Lagrange .....	57
FIGURA 26	– Modelo de comparação dos filtros .....	59
FIGURA 27	– Modelagem do compensador de <i>clock skew</i> com estrutura de Farrow ...	61
FIGURA 28	– Modelo do estimador de <i>clock skew</i> proposto por Razavi. ....	62
FIGURA 29	– Modelo do estimador de <i>clock skew</i> proposto por Razavi com modificações. ....	63
FIGURA 30	– Modelo do estimador de ganho .....	64
FIGURA 31	– Modelo do estimador de <i>offset</i> .....	65
FIGURA 32	– Fluxograma do processo de desenvolvimento dos subsistemas de compensação. ....	67
FIGURA 33	– Fluxograma do processo de desenvolvimento do sistema completo de compensação. ....	68
FIGURA 34	– Ilustração do processo de teste FPGA In Loop. ....	70
FIGURA 35	– Diagrama de interface da simulação FPGA In Loop .....	71
FIGURA 36	– Gráfico da comparação de ordem da estrutura de Farrow Lagrange. ....	74
FIGURA 37	– Gráfico do SNR para comparação da estrutura de Farrow Lagrange com	

	$L_2$ Laakso. ....	75
FIGURA 38	– Gráfico do SFDR para comparação da estrutura de Farrow Lagrange com $L_2$ Laakso. ....	75
FIGURA 39	– Gráfico do SFDR para comparação da estrutura de Farrow Lagrange com $L_2$ Laakso em ponto fixo. ....	76
FIGURA 40	– Gráfico do SFDR para comparação de diferentes filtros de delay fracionário. ....	76
FIGURA 41	– Gráfico do SFDR para comparação da ordem e da faixa de <i>delay</i> da estrutura $L_2$ Laakso com filtro GLS. ....	77
FIGURA 42	– Gráfico da magnitude do erro de aproximação da estrutura de Farrow $L_2$ Laakso com filtro de Lagrange. ....	78
FIGURA 43	– Gráfico da magnitude do erro de aproximação da estrutura de Farrow $L_2$ Laakso com filtro GLS. ....	79
FIGURA 44	– Curva do valor estimado de <i>delay</i> para compensação do clock skew. ...	80
FIGURA 45	– Espectro do TIADC com <i>clock skew</i> e sem compensação. ....	81
FIGURA 46	– Espectro do TIADC com <i>clock skew</i> e com compensação. ....	82
FIGURA 47	– Espectro TIADC com desvio de ganho e sem compensação. ....	83
FIGURA 48	– Espectro TIADC com desvio de ganho e com compensação. ....	83
FIGURA 49	– Curva do multiplicador de ganho ao longo do tempo. ....	84
FIGURA 50	– Espectro TIADC com desvio de offset e sem compensação ....	85
FIGURA 51	– Espectro TIADC com desvio de <i>offset</i> e com compensação ....	86
FIGURA 52	– Espectro do TIADC com todos os desvios e sem compensação. ....	87
FIGURA 53	– Espectro do TIADC com todos os desvios e compensação do <i>clock skew</i> . ....	87
FIGURA 54	– Espectro do TIADC com todos os desvios e compensação do <i>clock skew</i> e do ganho. ....	88
FIGURA 55	– Espectro do TIADC com todos os desvios e compensação de todos os desvios. ....	88

## LISTA DE TABELAS

TABELA 1	– Tabela de características do TIADC e sistema de compensação. ....	17
TABELA 2	– Tabela comparativa dos métodos de compensação .....	31
TABELA 3	– Tabela comparativa dos métodos de projeto de FDF. ....	47
TABELA 4	– Tabela de comparação entre as estruturas de Farrow com filtro GLS. ...	80
TABELA 5	– Tabela de comparação do desempenho do TIADC antes e depois da compensação de <i>clock skew</i> em 180MHz. ....	81
TABELA 6	– Tabela de comparação do desempenho do TIADC antes e depois da compensação de ganho em 180MHz. ....	83
TABELA 7	– Tabela de comparação do desempenho do TIADC antes e depois da compensação de <i>offset</i> em 180MHz. ....	85
TABELA 8	– Tabela de comparação do desempenho do TIADC antes e depois do sistema completo de compensação em 180MHz. ....	89
TABELA 9	– Tabela de utilização de recursos e desempenho dos estimadores e compensadores de desvio. ....	89
TABELA 10	– Tabela de utilização de recursos para o projeto utilizando <i>FPGA In Loop</i> . ....	90
TABELA 11	– Tabela de comparação do desempenho do TIADC antes e depois do sistema completo de compensação em 180MHz. Teste com <i>FPGA In Loop</i> . ....	90

## LISTA DE SIGLAS

ADC	Analog to Digital Converter
AFB	Advanced Filter Bank
DC	Direct Current
DSP	Digital Signal Processing
ENOB	Effective Number of Bits
FDL	Filtro de Delay Fracionário
FFT	Fast Fourier Transform
FIL	FPGA In Loop
FIR	Finite Impulse Response
FPGA	Field Programable Gate Array
<i>FS</i>	Full Scale
GHz	Gigahertz
GLS	Generalized Least Square
HDL	Hardware Description Language
JTAG	Joint Test Action Group
LS	Least Square
LUT	Look Up Table
MCM	Multiple Constant Multiplication
MHz	Megahertz
MS/s	Mega Samples per Second
RF	Rádio frequência
RMS	Root Mean Square
RSS	Root Sum Square
SDR	Software Defined Radio
SFDR	Spurious Free Dynamic Range
SINAD	Signal to Noise and Distortion Ratio
SNR	Signal to Noise Ratio
TIADC	Time Interleaved Analog to Digital Converter
USB	Universal Serial Bus
VDL	Variable Delay Line
VHDL	Very high speed integrated circuit Hardware Description Language

## LISTA DE SÍMBOLOS

$M$	Número de ADCs de um TIADC
$F_{ADC}$	Frequência de amostragem de um ADC do TIADC
$F_{TIADC}$	Frequência de amostragem do TIADC
$T_s$	Intervalo de amostragem
$F_s$	Frequência de amostragem
$\delta(t)$	Função Dirac ou impulso
$x(t)$	Sinal contínuo no tempo
$t$	Variável de tempo
$p(t)$	Função Dirac periódica
$x_p(t)$	Sinal contínuo amostrado no tempo
$\omega_s$	Frequência de amostragem em radianos por segundo
$X_p(j\omega)$	Transformada de Fourier de $x_p(t)$
$\omega$	Frequência angular
$X(j\omega)$	Transformada de Fourier de $x(t)$
$\omega_m$	Frequência angular do sinal $x(t)$
$x_q$	Sinal $x(t)$ quantizado na amplitude
$N_b$	Número de <i>bits</i>
$q_a$	Quantum de um ADC
$e_q$	Erro de quantização do ADC
$e_q(t)$	Erro de quantização do ADC em função do tempo
$s$	Inclinação da reta da onda dente-de-serra
$r$	Ruído de quantização
$RMS_{sinal}$	Valor RMS do sinal de entrada de um ADC
$RSS_{ruído}$	Valor RSS de todos os componentes espectrais presentes na saída de um ADC, excluindo-se o sinal de entrada e o nível DC
$dB$	Decibéis
$BW$	Banda espectral de um sinal
$dBFS$	Decibéis em relação ao fundo de escala
$CLK_i$	Iésimo sinal de <i>clock</i> para controle da amostragem de um TIADC
$x[n]$	Sinal $x(t)$ discretizado no tempo
$n$	índice da amostra
$f_{offset}$	Frequência do tom espúrio do TIADC devido ao desvio de <i>offset</i>
$k$	Variável de índice
$f_{ganho}$	Frequência do tom espúrio do TIADC devido ao desvio de ganho
$f_{in}$	Frequência do sinal senoidal de entrada do TIADC
$\Delta T_s$	Diferença no intervalo de amostragem em relação a $T_s$
$\Delta V$	Diferença na amplitude do valor amostrado devido ao desvio de <i>clock skew</i>
$\Delta t$	Erro no instante de aquisição de um dos canais de um TIADC
$y(n)$	Sinal discreto no tempo saída do TIADC
$f_0$	Sinal senoidal de entrada do TIADC
$T_{TIADC}$	Intervalo de amostragem do TIADC

$\phi$	Fase do sinal de entrada do TIADC, em radianos
$h(t)$	Função no tempo contínuo do filtro ideal de atraso
$t_l$	Intervalo de tempo contínuo em que o sinal de entrada de $h(t)$ é atrasado
$y(t)$	Sinal contínuo no tempo de saída de um sistema
$h(nT_s)$	Função no tempo discreto do filtro ideal de atraso
$D$	Atraso total de um sinal
$x(nT_s)$	Sinal $x(t)$ amostrado com um intervalo de tempo $T_s$
$y(nT_s)$	Sinal $y(t)$ amostrado com um intervalo de tempo $T_s$
$Y(j\omega)$	Transformada de Fourier de $y(t)$
$H(j\omega)$	Função de transferência de $h(t)$
$\Phi_{H(j\omega)}$	Fase de $H(j\omega)$
$\tau_{H(j\omega)}$	Atraso de fase de $H(j\omega)$
$H_{ap}(j\omega)$	Função de transferência do filtro de aproximação de $h(nT_s)$
$E(e^{j\omega})$	Erro de aproximação de $H_{ap}(j\omega)$
$E_2$	Integral de erro quadrático
$h_{ap}(n)$	Resposta ao impulso do filtro discreto de aproximação do filtro ideal $h(n)$
$N$	Ordem de um filtro
$M_i$	índice do primeiro valor diferente de zero na resposta ao impulso de um filtro
$\alpha$	Parâmetro da largura de banda passante
$W(\omega)$	Função peso no domínio da frequência
$\mathbf{h}$	Vetor dos coeficientes do filtro FIR
$\mathbf{e}$	Vetor de exponenciais complexas
$\mathbf{P}$	Matriz de coeficientes para o projeto de filtros GLS
$\mathbf{p}_1$	Vetor de coeficientes para o projeto de filtros GLS
$K$	Constante inteira
$\Omega_k$	Frequência angular onde a magnitude do filtro é zero (filtro Minimax)
$E_\Omega$	Matriz de exponenciais complexas em $\Omega_k$
$\mathbf{e}_D$	Vetor de exponenciais complexas em $D\Omega_k$
$\mathbf{P}_\Omega$	Matriz da parte real e imaginária de $E_\Omega$
$\mathbf{p}_\Omega$	Matriz da parte real e imaginária de $\mathbf{e}_D$
$Q$	Constante inteira
$\mu$	<i>Delay</i> fracionário
$M_O$	Ordem polinomial da estrutura de Farrow
$C_m(n)$	Vetor de coeficientes do m-ésimo filtro FIR da estrutura de Farrow
$h_m(n)$	Filtro composto dos coeficientes polinomiais em $m$ da estrutura de Farrow
$x$	Variável real
$a_n$	Variável de coeficiente real
$y$	Variável real
$q$	Índice inteiro
$z$	Variável da transformada $z$
$H_m(z)$	Função de transferência do filtro $h_m(n)$
$\mathbf{V}$	Matriz de Vandermonde
$\mathbf{c}$	Vetor de coeficientes em $z$

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>15</b>
1.1 MOTIVAÇÃO	15
1.2 OBJETIVOS	16
1.2.1 Objetivo Geral	16
1.2.2 Objetivos Específicos	16
1.3 ESTRUTURA DA DISSERTAÇÃO	17
<b>2 CONVERSOR ANALÓGICO DIGITAL INTERCALADO</b>	<b>18</b>
2.1 CONCEITOS GERAIS	18
2.1.1 Introdução e conceito de amostragem	18
2.1.2 Quantização	19
2.1.3 Métricas de desempenho para conversores analógico digital	21
2.2 PRINCÍPIO DE OPERAÇÃO DO CONVERSOR ANALÓGICO DIGITAL INTERCALADO	23
2.3 DESVIOS DO CONVERSOR ANALÓGICO DIGITAL INTERCALADO	24
2.3.1 Offset	24
2.3.2 Ganho	25
2.3.3 Clock Skew	27
2.4 COMPENSAÇÃO DOS DESVIOS	29
2.4.1 Offset	32
2.4.2 Ganho	32
2.4.3 Clock Skew	32
<b>3 FILTROS DE DELAY FRACIONÁRIO</b>	<b>34</b>
3.1 FILTROS DE DELAY FRACIONÁRIO	34
3.1.1 Projeto do filtro pela integral do mínimo erro quadrático	36
3.1.2 Projeto do filtro pela aproximação generalizada dos mínimos quadrados	39
3.1.3 Projeto do filtro utilizando método Minimax	42
3.1.4 Projeto do filtro utilizando interpolação de Lagrange	44
3.1.5 Sumário do projeto de filtros de delay fracionário	46
3.2 ESTRUTURA DE FARROW	47
<b>4 MODELAGEM DO SISTEMA E METODOLOGIA APLICADA</b>	<b>52</b>
4.1 MODELAGEM DO TIADC	53
4.2 MODELAGEM DAS ESTRUTURAS DE COMPENSAÇÃO	54
4.2.1 Compensação do clock skew	55
4.2.2 Compensação do ganho	63
4.2.3 Compensação do offset	64
4.3 METODOLOGIA DE DESENVOLVIMENTO E TESTE	66
4.3.1 Metodologia de desenvolvimento	66
4.3.2 Metodologia de teste	69
<b>5 DISCUSSÃO DOS RESULTADOS</b>	<b>73</b>
5.1 CLOCK SKEW	73
5.2 GANHO	82



5.3	OFFSET .....	84
5.4	SISTEMA COMPLETO .....	86
5.5	SIMULAÇÃO COM FPGA IN LOOP .....	89
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>91</b>
	<b>REFERÊNCIAS .....</b>	<b>94</b>
	<b>Apêndice A – DEDUÇÃO MATEMÁTICA DA IMAGEM NA SAÍDA DO TIADC ..</b>	<b>98</b>

# 1 INTRODUÇÃO

## 1.1 MOTIVAÇÃO

Nos modernos receptores de rádio frequência (RF) digitais para aplicações móveis são necessárias ao menos duas translações de frequência para o início do processo de demodulação do sinal. A translação da frequência de portadora do sinal RF para uma frequência intermediária aumenta a complexidade dos circuitos utilizados nos receptores. Estes circuitos também apresentam limitações de desempenho tais como limitação do *range* dinâmico e adição de ruído (RIVET et al., 2010).

Uma solução interessante para este problema está no conceito de *rádio definido por software* (SDR, *Software Defined Radio*) proposta por Mitola (MITOLA, 2000), que consiste em uma estrutura de receptor formada por uma antena, um conversor analógico digital (ADC, *Analog to Digital Converter*) e um bloco de processamento digital de sinais (DSP, *Digital Signal Processing*). Neste conceito em que o ADC é colocado logo após a antena, funções como filtros e *mixers* analógicos são implementadas digitalmente no bloco DSP. Entretanto, nesta topologia o ADC torna-se um ponto crítico do sistema, já que deste é requerida uma elevada taxa de amostragem e elevada resolução (VOGEL; JOHANSSON, 2006).

É neste contexto que a estrutura do Conversor Analógico Digital Intercalado (TIADC, *Time Interleaved Analog to Digital Converter*) surge como uma interessante solução (RAZAVI, 2013). A estrutura do TIADC consiste na aquisição do sinal de entrada por um número  $M$  de ADCs paralelos, cada qual adquirindo o sinal de entrada à uma taxa de amostragem de  $F_{ADC}$ .

Cada ADC adquire o sinal de entrada em instante diferente dos demais, de forma intercalada no tempo, gerando na saída do sistema um sinal equivalente com taxa de amostragem  $F_{TIADC} = M \cdot F_{ADC}$ . Idealmente, o desempenho do TIADC, que pode ser indicado por métricas tais como o SNR (*Signal to Noise Ratio*) e o SFDR (*Spurious Free Dynamic Range*), é similar a dos seus ADCs constituintes. Mas, devido a desvios no instante de aquisição (*clock skew*) e no casamento do ganho e *offset* individual de cada conversor, o desempenho é degradado (VOGEL; JOHANSSON, 2006) (RAZAVI, 2013).

Entretanto, os problemas oriundos dos circuitos analógicos destes conversores podem ser tratados digitalmente (ROERMUND et al., 2005). A compensação digital destes desvios possui complexidade significativa, em especial para a compensação do *clock skew*, sendo esta uma área de intensa pesquisa.

## 1.2 OBJETIVOS

### 1.2.1 OBJETIVO GERAL

O principal objetivo deste trabalho foi implementar e validar em FPGA (*Field Programmable Gate Array*) sistemas digitais de compensação dos desvios do TIADC, responsáveis pela degradação do desempenho do conversor. A validação destes sistemas é realizada com a sua implementação em FPGA, e simulação dos sinais resultantes do TIADC em *software*.

A implementação e validação do sistema de compensação em FPGA permitirá então a sua futura inserção em um circuito integrado de um TIADC.

### 1.2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho foram:

- Modelar o TIADC e os sistemas digitais de compensação utilizando *software* Simulink.
- Avaliar o desempenho de diferentes métodos de projeto de filtros de atraso fracionário utilizando aritmética de ponto fixo.
- Gerar código HDL (*Hardware Description Language*) para síntese em FPGA dos sistemas digitais de compensação.
- Utilizar uma interface entre o modelo do TIADC em Simulink e o sistema sintetizado em FPGA para validação.
- Validar os sistemas de compensação em FPGA, utilizando um modelo comportamental do TIADC em Simulink para isto.

O TIADC que será modelado conterá dois ADCs, cada um operando com uma frequência de amostragem  $F_{ADC}$  de 1 GHz e com resolução de 8 *bits*. A frequência de amostragem do TIADC ( $F_{TIADC}$ ) será portanto 2 GHz. A banda mínima de interesse que

será considerada para o TIADC é de 200 MHz, ou seja, espera-se que o TIADC realize o *oversampling* do sinal de entrada. Como será discutido no capítulo 2, a operação de um ADC com *oversampling* do sinal de entrada permite obter ganhos no seu desempenho. O sistema de compensação que será implementado deverá compensar os desvios do TIADC ao menos para esta banda mínima de interesse. As características desejadas do TIADC e do sistema de compensação são então resumidas na tabela 1.

**Tabela 1: Tabela de características do TIADC e sistema de compensação.**

Descrição da característica	Valor
Número de ADCs	2
Frequência de amostragem de cada ADC	1 GHz
Resolução do ADC	8 bits
Frequência de amostragem do TIADC	2 GHz
Frequência máxima do sinal de entrada do TIADC	200 MHz

**Fonte: Autoria própria**

### 1.3 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação é dividida em seis capítulos. No segundo capítulo são apresentados os conceitos básicos do TIADC, a formulação teórica dos desvios e seu impacto no desempenho do sistema, assim como é realizada uma revisão da literatura com os principais métodos de compensação dos desvios. No terceiro capítulo o conceito dos filtros de *delay* fracionário e a estrutura de Farrow são apresentados. Os conceitos mostrados nestes dois capítulos são essenciais para a compreensão da estrutura de compensação implementada nos capítulos seguintes. O quarto capítulo descreve os modelos comportamentais do TIADC e a estrutura de compensação em Simulink. Este capítulo apresenta também a descrição da conversão dos modelos comportamentais em código HDL e a interface de teste entre a FPGA e o Simulink. Os resultados obtidos com estas estruturas e o uso de recursos de *hardware* utilizados em FPGA são apresentados no quinto capítulo. E, finalmente, no sexto capítulo são apresentadas as conclusões finais e propostas para futuros trabalhos.

## 2 CONVERSOR ANALÓGICO DIGITAL INTERCALADO

### 2.1 CONCEITOS GERAIS

#### 2.1.1 INTRODUÇÃO E CONCEITO DE AMOSTRAGEM

O conversor analógico digital é o dispositivo que converte sinais analógicos contínuos no tempo em sinais digitais discretizados no tempo. A conversão do sinal analógico em digital no ADC pode ser entendida como dois processos distintos: *amostragem* e *quantização*. O primeiro processo, a *amostragem*, consiste na discretização no tempo do sinal. A *amostragem* do sinal ocorre em intervalos igualmente espaçados no tempo,  $T_s$ , ou similarmente, a uma frequência  $F_s$ . Se respeitado o *teorema da amostragem* (OPPENHEIM et al., 1996), a sequência de amostras obtidas do sinal o representa de forma unívoca. Uma forma conveniente de representar a amostragem é utilizar um trem de impulsos (função Dirac  $\delta(t)$ ) periódico multiplicado pelo sinal analógico. Considere o sinal analógico  $x(t)$  (contínuo no tempo  $t$ ) e a função Dirac periódica  $p(t)$ , com período de amostragem  $T_s$ . Então o sinal amostrado  $x_p(t)$  pode ser representado por:

$$x_p(t) = p(t)x(t) \quad (1)$$

onde,

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s). \quad (2)$$

Aplicando-se (2) em (1) pode-se representar o sinal  $x_p(t)$  pela equação:

$$x_p(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s). \quad (3)$$

Aplicando-se a transformada de Fourier à equação (3) é obtido:

$$X_p(j\omega) = \frac{1}{2\pi}[X(j\omega) * P(j\omega)] = \frac{1}{2\pi}[X(j\omega) * \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s)], \quad (4)$$

onde  $\omega_s = Fs/2\pi$ . Como a convolução de um sinal por um impulso o desloca no eixo do tempo, ou no eixo da frequência, é obtida a equação (5).

$$X_p(j\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(j(\omega - k\omega_s)). \quad (5)$$

$X_p(j\omega)$  é uma função periódica em  $\omega$  constituída da superposição das réplicas deslocadas de  $X(j\omega)$  (OPPENHEIM et al., 1996). Para respeitar o *critério de Nyquist*, a maior frequência  $\omega_m$  de  $X(j\omega)$  deve ser menor do que  $\omega_s/2$ . Se esta condição não for respeitada ocorre o *aliasing* do sinal amostrado, a sobreposição no espectro das réplicas de  $X_p(j\omega)$ . Esta situação é denominada de *subamostragem* (ou *subsampling*). A condição inversa, quando a frequência de amostragem  $\omega_s$  é maior do que  $2\omega_m$ , é denominada *oversampling*. O *oversampling* permite um ganho de desempenho no processo de aquisição por parte do ADC. O segundo processo da conversão analógico digital, a quantização, será discutido na subseção seguinte.

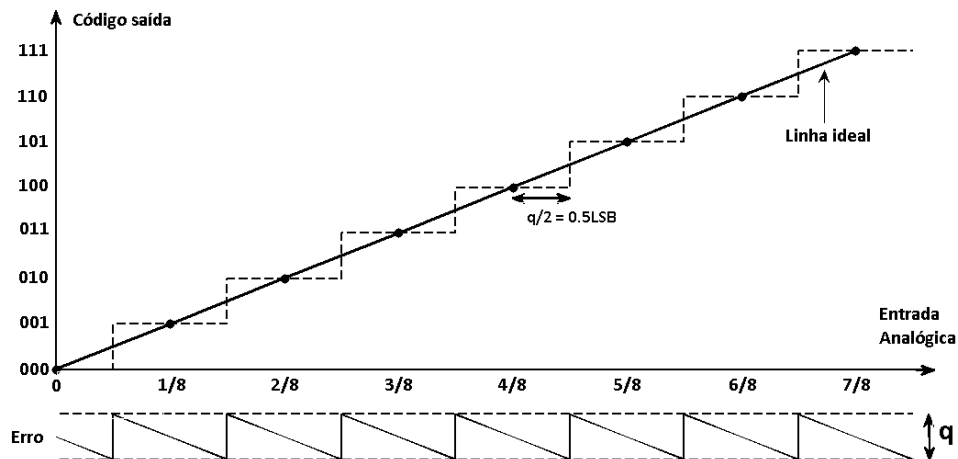
### 2.1.2 QUANTIZAÇÃO

A quantização consiste na função de transferência que associa ao sinal analógico amostrado  $x(nT_s)$  um valor codificado  $x_q$  (denominado *código*), sendo  $x_q$  restrito à um conjunto limitado de valores. A cada valor  $x_q$  obtido está associado um intervalo de amplitude de  $x(nT_s)$ . Este intervalo de amplitude é igualmente espaçado para todos os valores de  $x_q$  e a sua amplitude é chamada de *quantum*. Este representa a menor variação na amplitude do sinal de entrada que causa uma mudança no *código*. Normalmente o código é um número binário representado por  $N_b$  bits. O número  $N_b$  de bits utilizado para representar o código binário, e conseqüentemente o número de faixas representadas pelo ADC, é a resolução do conversor. O fundo de escala ( $FS$ , *Full Scale*) do conversor é correlacionado com o *quantum* pela equação (6), onde  $q_a$  representa a amplitude do *quantum* e  $N_b$  representa o número de bits de resolução do ADC.

$$FS = q_a(2^{N_b} - 1). \quad (6)$$

A figura 1 ilustra a função de transferência e a quantização do ADC.

A quantização introduz um erro sistemático no ADC, o erro de quantização. Este erro decorre da diferença entre a amplitude do sinal analógico de entrada e de sua saída digital. O erro de quantização  $e_q$  está contido no intervalo  $[-q_a/2; q_a/2]$  e, para qualquer sinal de frequência maior do que zero, pode ser aproximado por uma forma de onda dente-



**Figura 1: Função de transferência de um ADC ideal e erro de quantização.**

**Fonte: Autoria própria**

de-serra contendo uma amplitude pico-a-pico igual à  $q_a$  (KESTER, 2005). Considerando esta simplificação no modelo do erro de quantização, o mesmo pode ser expresso pela equação (7).

$$e_q(t) = st, \frac{-q_a}{2s} < t < \frac{q_a}{2s}. \quad (7)$$

Onde  $e_q(t)$  representa o erro de quantização e  $s$  representa a inclinação da reta da onda dente-de-serra. O erro médio quadrático de  $e_q(t)$  é dado pela equação (8).

$$\overline{e_q^2(t)} = \frac{s}{q_a} \int_{-\frac{q_a}{2s}}^{\frac{q_a}{2s}} (st)^2 dt \quad (8)$$

Resolvendo a integral da equação (8) e simplificando-a é obtida a equação (9).

$$\overline{e_q^2(t)} = \frac{q_a^2}{12}. \quad (9)$$

E o valor *RMS* (*Root Mean Square*) do ruído de quantização ( $r$ ) é dado pela equação (10).

$$\sqrt{r^2(t)} = \frac{q_a}{\sqrt{12}}. \quad (10)$$

A análise simplificada do ruído de quantização o considera não correlacionado com o sinal de entrada, e que a distribuição estatística de probabilidade do erro de quantização é aproximadamente uniforme. Do exposto conclui-se que o erro de quantização é proporcional ao *quantum*  $q_a$ . Medições mais precisas de um ADC dependem portanto de um *quantum* menor, ou seja, de uma maior resolução do ADC.

### 2.1.3 MÉTRICAS DE DESEMPENHO PARA CONVERSORES ANALÓGICO DIGITAL

Para a adequada especificação e avaliação dos conversores analógico digitais são necessárias métricas de desempenho. Diversos aspectos dos ADCs devem ser avaliados antes de sua seleção ou especificação, onde a relevância de cada aspecto é dependente da aplicação. As principais métricas para análise que serão aqui consideradas são a *SNR* (*Signal to Noise Ratio*), a *SINAD* (*Signal to Noise and Distortion Ratio*), o *SFDR* (*Spurious Free Dynamic Range*) e o *ENOB* (*Effective Number of Bits*). A primeira delas, a *SNR*, é definida como a razão entre o valor RMS do sinal de entrada e o valor médio do RSS (*Root Sum Square*) de todos os demais componentes espectrais, excluindo as cinco primeiras harmônicas e o nível DC (KESTER, 2005). A *SINAD* é definida da mesma forma que a *SNR*, mas com a adição das cinco primeiras harmônicas do espectro. O cálculo da *SINAD* é dado por:

$$SINAD = 10 \log_{10} \frac{RMS_{sinal}}{RSS_{ruído}} \text{ dB}, \quad (11)$$

onde  $RMS_{sinal}$  é o valor RMS do sinal de entrada e  $RSS_{ruído}$  é o RSS de todos os demais componentes espectrais, exceto o nível DC.

Ambas as métricas são indicadores do desempenho dinâmico do ADC e, usualmente, ambas são fornecidas em unidades de *dB* (decibéis). Em um ADC ideal de  $N_b$  bits de resolução, onde somente o ruído de quantização está presente, a *SNR* para um sinal senoidal cujo a amplitude de pico é o fundo de escala do ADC é dada por:

$$SNR = 6,02N_b + 1,76 \text{ dB} \quad (12)$$

Na forma apresentada na equação (12) é considerada toda a banda de Nyquist, ou seja, do nível contínuo do sinal até a frequência  $F_s/2$ . Em muitas aplicações o sinal de interesse pode ocupar uma banda *BW* menor do que Nyquist. Nestas aplicações é possível retirar o ruído externo à banda *BW* utilizando filtros digitais e aumentar o desempenho da *SNR*. Este aumento do desempenho na *SNR* é chamado de *ganho de processo* (KESTER, 2005) e pode ser incluído na equação (12), considerando que o ruído do ADC é branco, como:

$$SNR = 6,02N + 1,76 + 10 \log_{10} \frac{F_s}{2BW}. \quad (13)$$

O ganho de processo dado pela equação (13) é obtido com o uso de *oversampling* somente, sem a utilização de quaisquer outros componentes em conjunto com o ADC. Ganhos

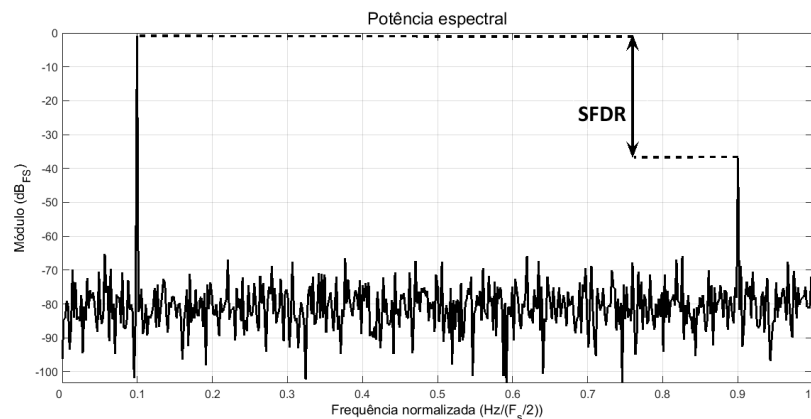


na SNR maiores do que os obtidos na equação (13) podem ser obtidos do ADC quando este é utilizado em conjunto com um modulador Sigma-Delta (MARIANO, 2008).

O desempenho de um ADC real, entretanto, é inferior ao de um ADC ideal devido a outras fontes de ruído e erros no conversor, além do ruído de quantização. Uma forma de avaliar o desempenho de ADCs considerando estes ruídos é utilizando o *ENOB*. O *ENOB* permite a comparação do desempenho de um ADC real com relação à um ADC ideal, contendo um valor efetivo de bits. O valor de *ENOB* pode ser obtido da *SINAD* do ADC, substituindo a *SNR* na equação (12) pela *SINAD* e reorganizando-a em função de  $N_b$ . Desta forma o *ENOB* pode ser calculado pela equação (14).

$$ENOB = \frac{SINAD - 1,76dB}{6,02} \quad (14)$$

Em aplicações de ADCs para sistemas de comunicação o *SFDR* assume um significado especial. Este está relacionado com o menor sinal capaz de ser detectado (sensibilidade) por um receptor de rádio. O *SFDR* é definido como a razão da amplitude RMS do sinal de entrada pela amplitude do componente espectral espúrio de pico medido na banda de interesse (KESTER, 2005). Usualmente o *SFDR* é especificado em relação à um sinal que atinge o valor de fundo de escala do ADC, em *dBFS* (decibéis em relação ao fundo de escala).



**Figura 2:** Espectro de um ADC e seu respectivo *SFDR* (*Spurious Free Dynamic Range*). O eixo da frequência é normalizado em relação a frequência de amostragem  $F_s$ .

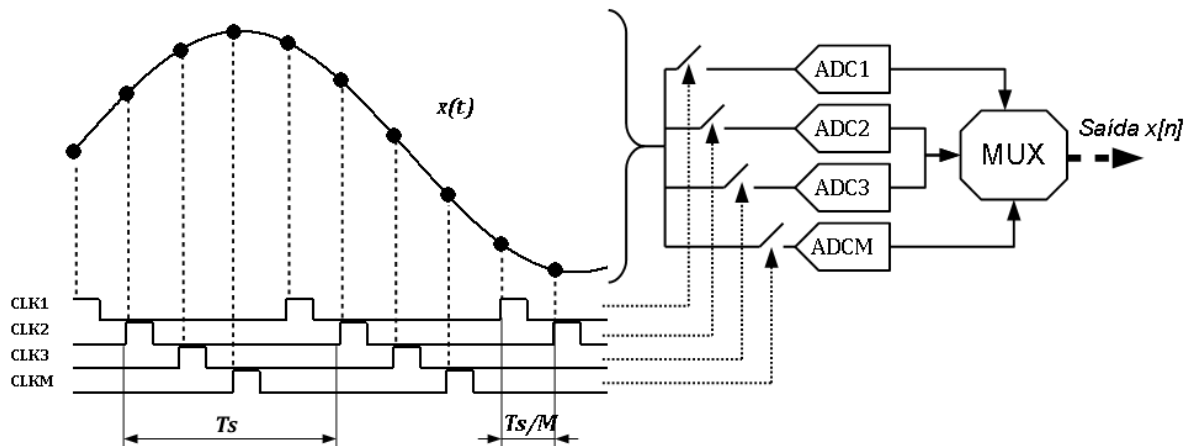
**Fonte:** Autoria própria

A figura 2 ilustra o espectro de um ADC e seu respectivo *SFDR*. Na figura 2 há um sinal de entrada que atinge o fundo de escala do ADC na frequência normalizada 0,1. O sinal de ruído espúrio de maior amplitude ocorre na frequência normalizada de 0,9. O *SFDR* neste exemplo será portanto menor do que 40 *dBFS*, pois esta é a diferença entre a amplitude do sinal

de entrada e o sinal espúrio de pico, como ilustrado pela seta dupla presente na figura.

## 2.2 PRINCÍPIO DE OPERAÇÃO DO CONVERSOR ANALÓGICO DIGITAL INTERCALADO

Considere o TIADC formado por  $M$  ADCs idênticos ilustrado na figura 3. Nesta estrutura cada um dos ADCs realiza uma amostragem do sinal de entrada contínuo no tempo  $x(t)$  na transição de borda positiva do sinal de *clock*  $CLK_i$ . Cada ADC possui o mesmo intervalo de tempo de aquisição  $T_s$ , mas as bordas de transição de *clock* estão defasadas no tempo por um intervalo de tempo  $T_s/M$ . O sinal digital de saída dos conversores é interligado a um multiplexador, e da saída deste é obtido o sinal  $x[n]$  formado pelas amostras intercaladas no tempo dos  $M$  ADCs. A saída  $x[n]$  possui um intervalo de tempo de aquisição  $T_s/M$ , ou seja, a frequência de aquisição  $F_{TIADC}$  do TIADC é igual à  $M \cdot F_{ADC}$ .



**Figura 3: Estrutura do TIADC com o detalhamento do controle dos instantes de aquisição. No lado esquerdo da figura o sinal  $x(t)$  é amostrado pelos  $M$  ADCs, à direita. O instante de aquisição destes ADCs são controlados pelo sinal de *clock*  $CLK1, CLK2, \dots, CLKM$ . Na saída dos ADCs é interligado um multiplexador.**

**Fonte: Autoria própria**

A principal vantagem do TIADC está na obtenção de maiores taxas de amostragem a partir de múltiplos ADCs que operam a taxas de amostragem menores. Desta forma esta estrutura pode ser aplicada em qualquer topologia de ADC para a obtenção de taxas de amostragem elevadas, inclusive além do que seria possível utilizando um único ADC (BLACK JR.; HODGES, 1980). A estrutura foi originalmente proposta por *Black* e *Hodges* com o objetivo de reduzir a área de ADCs do tipo *flash* de alta velocidade. Mas foi a pesquisa de (SUMANEN et al., 2001), um TIADC de quatro canais com 280mW de potência a uma taxa

de amostragem de 200 MS/s (*Mega Samples per Second*), que atraiu o interesse acadêmico para a sua aplicação. O aspecto que não havia sido explorado anteriormente é a relação custo benefício que existe entre a potência dissipada por esta estrutura em comparação com a sua taxa de amostragem (RAZAVI, 2013). A vantagem do TIADC deve-se ao fato de que os amplificadores operacionais dos conversores, quando são consideradas taxas de amostragem próximas do limite de banda dos amplificadores, possuem aumento de potência quadrático com relação ao aumento da taxa de amostragem (SUMANEN et al., 2001). Operar os sub-ADCs a taxas de amostragem menores mantêm a relação entre o aumento de potência e a taxa de amostragem linear.

## 2.3 DESVIOS DO CONVERSOR ANALÓGICO DIGITAL INTERCALADO

Idealmente o desempenho do TIADC é idêntico ao desempenho de seus sub-ADCs constituintes. Entretanto, devido a diferenças intrínsecas no *offset*, no *ganho* e no *instante de amostragem (clock skew)* entre os sub-ADCs, o desempenho do TIADC é degradado. Estas diferenças aparecem no espectro do sinal amostrado como *imagens*, sinais espúrios de erro, que ocorrem em frequências dependentes da taxa de amostragem do TIADC e do sinal de entrada. Nesta seção estes *desvios* do TIADC e seus efeitos são explanados.

### 2.3.1 OFFSET

A função de transferência de um ADC real contém um erro constante, chamado de *offset*. Para um único conversor este erro é estático. Na figura 4 a função de transferência de um ADC com *offset* é representada.

Entretanto, em um TIADC o *offset* de cada sub-ADC é diferente, gerando assim um sinal de erro periódico. Este sinal de erro periódico é independente do sinal de entrada no domínio do tempo e da frequência (KUROSAWA et al., 2001), possuindo tons espúrios no espectro do TIADC nas frequências:

$$f_{offset} = k \frac{F_{TIADC}}{M}, \quad k = 1, 2, \dots, M. \quad (15)$$

A figura 5 ilustra o espectro de um TIADC contendo somente o desvio de *offset*. Nesta figura é apresentada a potência espectral obtida da aquisição de um TIADC de quatro canais. Pode-se observar no espectro o sinal senoidal de entrada, identificado na figura, e os sinais espúrios resultantes do desvio de *offset*.

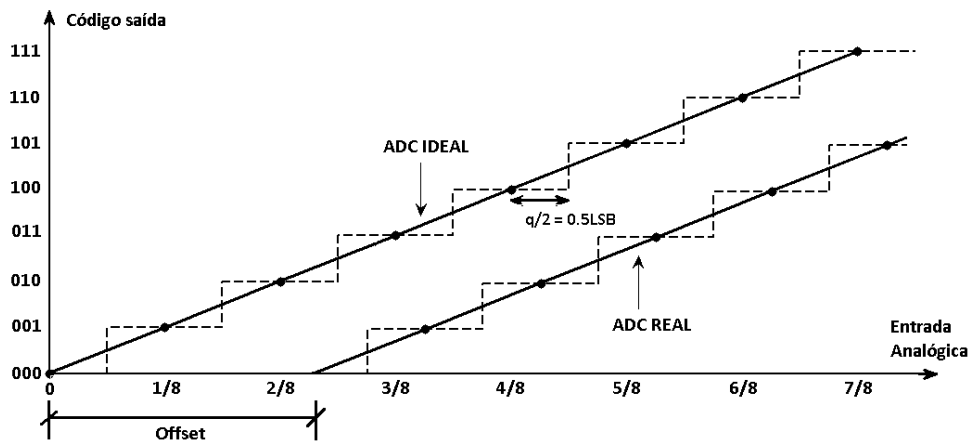


Figura 4: Função de transferência de um ADC com *offset*

Fonte: Autoria própria

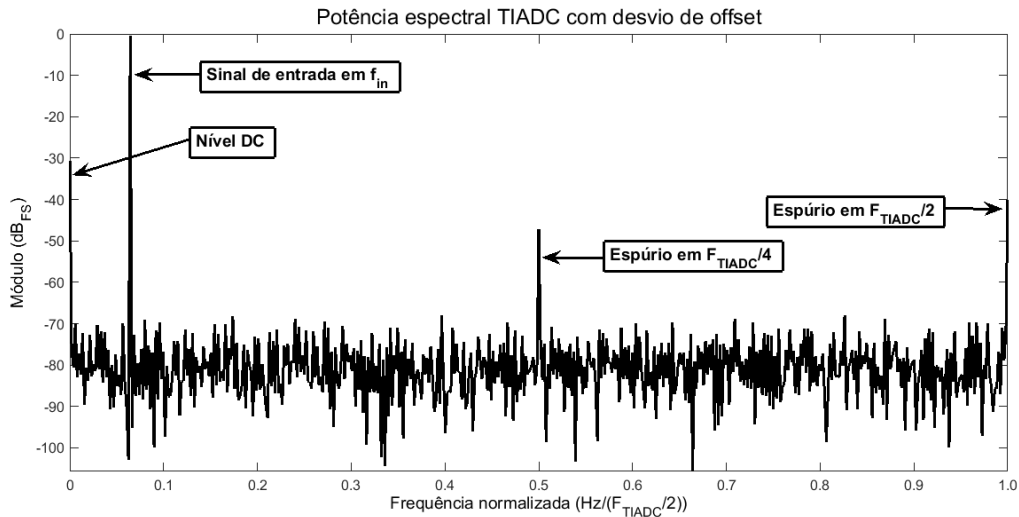


Figura 5: Espectro de um TIADC de 4 canais contendo somente o desvio de *offset*. O espectro está normalizada em relação a  $F_{TIADC}/2$ .

Fonte: Autoria própria

### 2.3.2 GANHO

A inclinação da reta da função de transferência de um ADC real é diferente de seu valor nominal, havendo assim um erro de *ganho* do conversor. Esta diferença na função de transferência é ilustrada pela figura 6. Em um TIADC cada sub-ADC pode possuir um ganho real diferente dos demais. Esta diferença no ganho dos sub-ADCs gera um sinal espúrio de erro na saída do TIADC, com picos nas frequências dadas pela equação (16).

$$f_{ganho} = \pm f_{in} + k \frac{F_{TIADC}}{M}, \quad k = 1, 2, \dots, M. \quad (16)$$

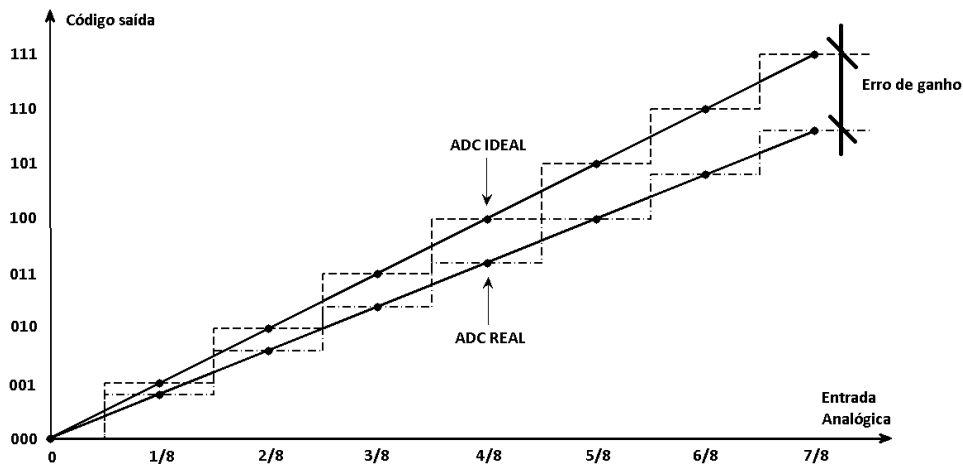


Figura 6: Função de transferência de um ADC real com desvio no valor de ganho

Fonte: Autoria própria

Onde  $f_{in}$  é a frequência do sinal de entrada do TIADC. O erro devido as diferenças de ganho no TIADC é modulado em amplitude pelo sinal de entrada (KUROSAWA et al., 2001). Desta forma, a magnitude no domínio da frequência destes sinais espúrios depende da amplitude do sinal de entrada. Deve-se observar que as frequências em que os sinais espúrios ocorrem não dependem da amplitude do sinal de entrada, somente de sua frequência. A degradação no desempenho do TIADC (observada através de seu SNR) pelo desvio de ganho é constante em função da frequência de entrada do sinal (KUROSAWA et al., 2001). A figura 7 ilustra o espectro de um TIADC, com 4 sub-ADCs, contendo somente o desvio de ganho.

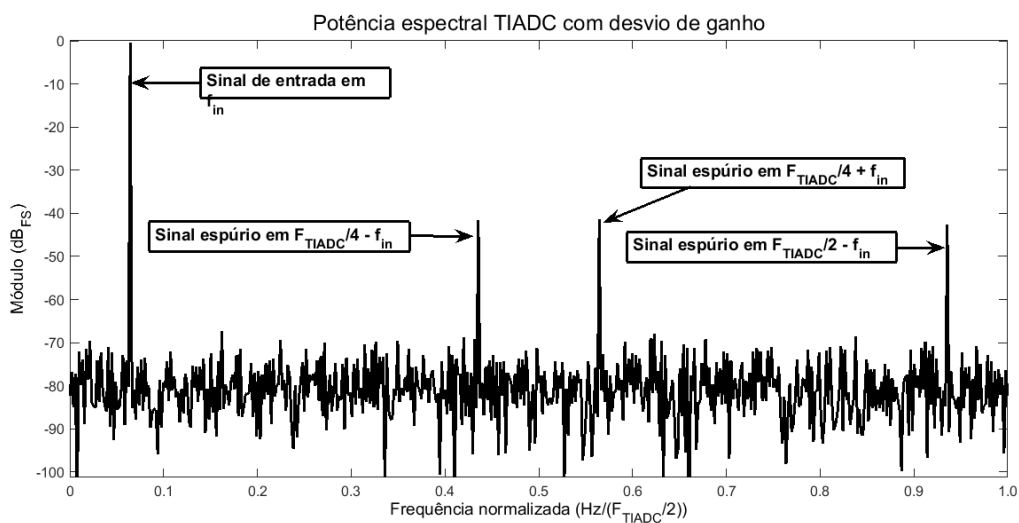
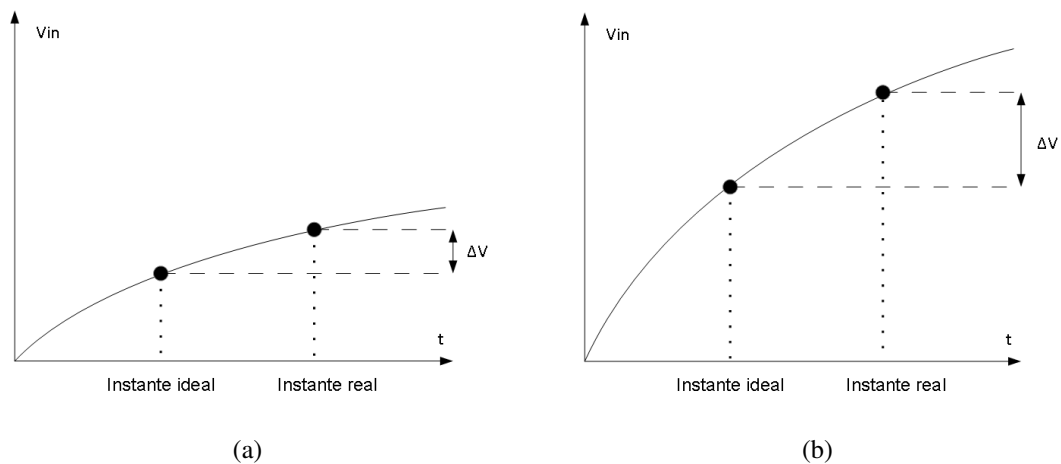


Figura 7: Espectro de um TIADC de 4 canais contendo somente o desvio de ganho. O espectro está normalizado em relação a  $F_{TIADC}/2$ .

Fonte: Autoria própria

### 2.3.3 CLOCK SKEW

O desempenho de conversores analógico digital também é degradado por erros no instante de aquisição do sinal analógico. Os principais erros são o *aperture jitter* e o *aperture delay*. O *aperture jitter* é um erro randômico, amostra por amostra, no instante de aquisição do sinal de entrada, causado por diversas fontes de ruído (ruído térmico, ruído da fonte de alimentação, etc) (MARIANO, 2008). O seu efeito sobre o conversor é a degradação do SNR do sinal adquirido. Já o *aperture delay* é um erro constante no ADC causado pelo atraso entre o comando de amostragem do sinal de entrada e o instante em que efetivamente o sinal é amostrado (MARIANO, 2008). Para compreensão do efeito do *aperture delay* considere o sinal contínuo  $x(t)$ , amostrado por um ADC com intervalo de amostragem  $T_s$ . Idealmente a saída  $y(n)$  deste ADC será o sinal amostrado  $x(nT_s)$ . Se entretanto, houver um erro  $\Delta T_s$  no intervalo de amostragem, a saída  $y(n)$  será dada por  $x(n(T_s + \Delta T_s))$ , gerando uma diferença  $\Delta V = x(n(T_s + \Delta T_s)) - x(nT_s)$  na amplitude do sinal de saída. Esta diferença de amplitude será maior quanto maior for o erro  $\Delta T_s$  e a derivada do sinal  $x(t)$ , como é ilustrado pela figura 8. Como sinais de maiores frequências possuem derivada proporcionalmente maiores, pode-se concluir que o efeito do *aperture delay* será mais significativo em frequências elevadas.



**Figura 8: Diferença na amplitude do sinal amostrado devido ao *aperture delay* para um sinal (a) de derivada baixa e (b) de derivada elevada.**

**Fonte: Autoria própria**

Em um TIADC as diferenças no delay de cada sub-ADC produz tons espúrios no espectro do sinal de saída, nas mesmas frequências dadas pela equação (16). Esta fonte de erro do TIADC é chamada de *clock skew*. Em (JAMAL et al., 2004) um TIADC de dois canais, contendo um erro  $\Delta t$  no instante de aquisição de um dos canais, é utilizado para a dedução do sinal de saída  $y(n)$  do TIADC. Considere que  $x(t) = \cos(2\pi f_0 t + \phi)$ , onde  $0 < 2\pi f_0 < \pi F_{TIADC}$ ,

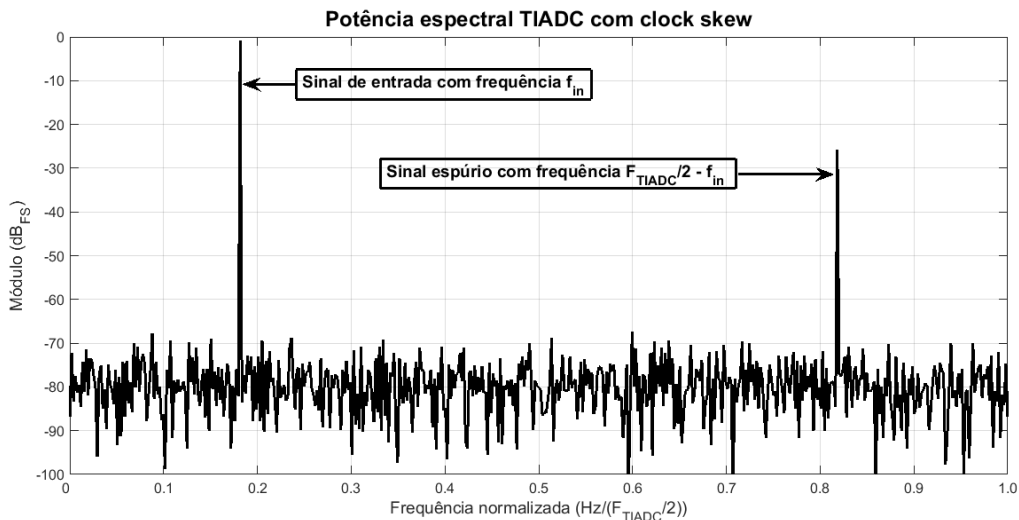
e que  $x(t)$  é amostrado por um TIADC de dois canais em que  $T_{TIADC} = 1/F_{TIADC}$ . A saída  $y(n)$  do TIADC pode ser dada então pela equação (17) (a dedução desta equação é apresentada no Apêndice A).

$$y(n) = \cos(\pi f_0 \Delta t) \cos(2\pi f_0 T_{TIADC} + \pi f_0 \Delta t + \phi) + \sin(\pi f_0 \Delta t) \sin((2\pi f_0 - \pi F_{TIADC})nT_{TIADC} + \pi f_0 \Delta t + \phi) \quad (17)$$

Na equação (17) se  $\|\Delta t\| \ll T_{TIADC}$ ,  $\cos(\pi f_0 \Delta t) \approx 1$  e  $\sin(\pi f_0 \Delta t) \approx \pi f_0 \Delta t$ . Portanto esta equação pode ser simplificada para:

$$y(n) = \cos(2\pi f_0 n T_{TIADC} + \pi f_0 \Delta t + \phi) - \pi f_0 \Delta t \sin(2\pi \left(\frac{F_{TIADC}}{2} - f_0\right) n T_{TIADC} + \pi f_0 \Delta t + \phi) \quad (18)$$

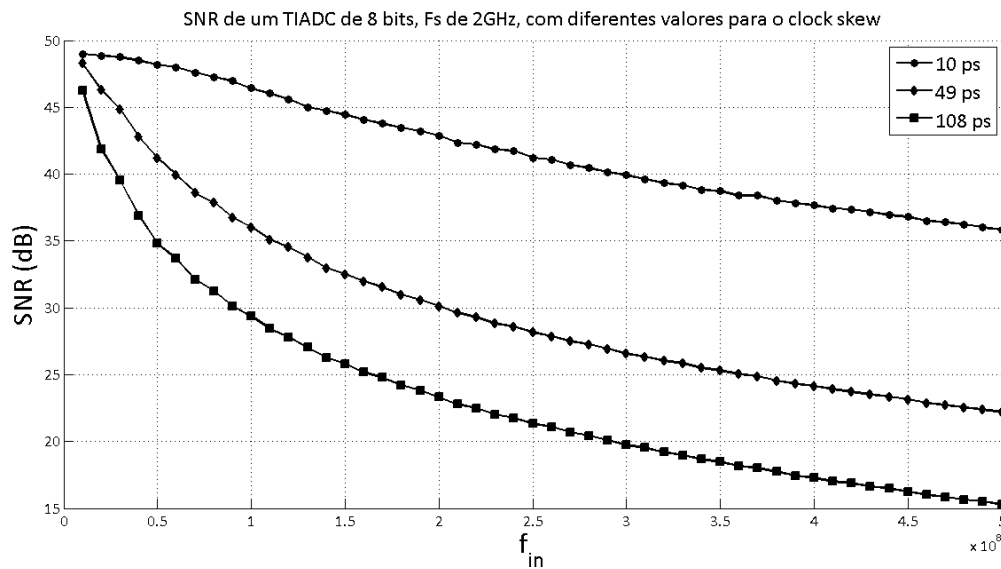
O primeiro termo da equação (18) refere-se ao sinal de entrada  $x(t)$  amostrado, o segundo termo refere-se ao sinal espúrio (chamado de *imagem*) decorrente do *clock skew*. Deste termo pode-se concluir que o sinal espúrio ocorre na frequência  $F_{TIADC}/2 - f_0$  e que, a sua amplitude é proporcional à frequência do sinal de entrada e ao erro  $\Delta t$ . A figura 9 ilustra o espectro deste TIADC de dois canais com o sinal amostrado e o sinal de imagem gerado.



**Figura 9:** Espectro do TIADC de dois canais com a presença do sinal espúrio gerado pelo *clock skew*.  $f_{in}$  é a frequência do sinal de entrada.

**Fonte:** Autoria própria

Comparado com o sinal de erro de ganho, o sinal de erro do *clock skew* possui um deslocamento de  $90^\circ$  com relação ao sinal de entrada (KUROSAWA et al., 2001). A degradação no SNR do sinal causada pelo *clock skew* aumenta com a frequência do sinal de entrada ( $f_{in}$ ), como pode ser visto pela figura 10.



**Figura 10: SNR versus  $f_{in}$  de um TIADC com desvio de *clock skew***

**Fonte: Autoria própria**

## 2.4 COMPENSAÇÃO DOS DESVIOS

A presença de tons espúrios no espectro do TIADC, devido aos desvios de *offset*, *ganho* e *clock skew*, degradam significativamente seu desempenho. Durante o projeto do circuito integrado, um *layout* cuidadoso do TIADC pode minimizar os seus efeitos (CONROY et al., ). Mas a eficácia desta abordagem é limitada (VOGEL; JOHANSSON, 2006). Desde a publicação de Black e Hodges (BLACK JR.; HODGES, 1980), diferentes métodos de compensação para cada um dos desvios do TIADC são apresentados na literatura.

A compensação dos desvios do TIADC pode ser dividida em duas partes: 1) estimação dos desvios, e 2) calibração, ou reconstrução do sinal de saída (VOGEL; JOHANSSON, 2006). A estimação dos desvios geralmente é realizada com o sinal de saída do TIADC. Do desvio estimado é gerado um sinal de controle para calibração, ou então, para a adequada reconstrução do sinal de saída. Os diferentes métodos de calibração podem ser classificados como: *online* e *offline*, que podem ser realizados em *foreground* ou *background*, com compensação *ativa* ou *passiva* (PARKEY; MIKHAEL, 2013). Entende-se por *online* os métodos que compensam os desvios com o TIADC em uso, atualizando estruturas e parâmetros, ou recalibrando, o TIADC de forma contínua ou periódica. *Offline* são os métodos que aplicam calibrações, ou pareamentos de características, no TIADC antes do seu uso na aplicação. *Softwares* que aplicam compensações nos dados obtidos do TIADC, na forma de um pós-processamento, e não verificam por variações temporais nos desvios do TIADC também são considerados *offline*.



Quando se faz necessário interromper o processo de conversão do TIADC para calibração, ou ajuste de parâmetros, e realizar estimativas dos desvios, (ou quando isto é feito sempre antes de ser iniciado o processo de amostragem) a compensação dos desvios ocorre em *foreground*. Este tipo de compensação é interessante para sistemas de instrumentação e testes, tais como osciloscópios. Quando a calibração, ou ajuste de parâmetros, e estimativas dos desvios ocorrem sem a interrupção da conversão do TIADC, a compensação ocorre em *background*. A compensação em *background* é necessária em aplicações de sistemas de comunicação, pois nestas aplicações geralmente o processo de conversão não pode ser interrompido (VOGEL; JOHANSSON, 2006).

Ainda pode-se dividir os métodos em *cegos*, ou *passivos*, quando não há informação prévia do sinal de entrada (na melhor das hipóteses os limites estatísticos do sinal são conhecidos), e em *não cegos*, ou *ativos*, quando é utilizado um sinal conhecido de entrada para calibração e estimação dos desvios. A tabela 2, modificada de (PARKEY; MIKHAEL, 2013), resume a classificação utilizada para os métodos de compensação, incluindo uma breve descrição das principais vantagens e desvantagens de cada categoria.

**Tabela 2: Tabela comparativa dos métodos de compensação de desvios do TIADC**

Ativo	Passivo	Foreground	Background	Online	Offline	Exemplo	Vantagens	Desvantagens
✓		✓		✓		Instrumento que provê um modo de calibração ativado por <i>software</i> .	Compensação dos desvios pode ser mais precisa com o uso de sinais de teste conhecidos e adequados.	Requer interrupção do sistema e não corrige variações nos desvios ao longo do tempo.
✓			✓	✓		Injeção de um tom em uma área livre do espectro.	Variações dos desvios rápidas e que ocorrem em períodos longos de tempo podem ser compensadas.	Limitado a correção dos desvios que podem ser medidos na frequência do tom especificado.
✓		✓			✓	Um sinal do tipo <i>chirp</i> é utilizado em produção para caracterização da resposta do TIADC.	Permite redução da complexidade computacional requerida do TIADC ao utilizar compensação de desvios como estáticos.	Cada sistema precisa ser testado individualmente e variações ao longo do tempo não são compensadas.
	✓	✓		✓		Um receptor de rádio recebe um preâmbulo de sincronização a cada pacote recebido e o utiliza para adaptar a compensação.	Disponibilidade de um sinal conhecido, permitindo adaptação mais rápidas ou precisas dos desvios.	Adição de <i>overhead</i> na transmissão de mensagens e possibilidade de desempenho reduzido se não houver compensação adequada durante a sincronização.
	✓		✓	✓		Um método cego adaptativo ajusta um filtro digital para compensar o <i>clock skew</i> .	Nenhum sinal adicional é requerido. O filtro pode compensar variações rápidas ou que ocorram após um longo período de tempo.	A complexidade computacional pode ser elevada. Uma FPGA adicional pode ser requerida para a compensação.
	✓	✓			✓	Utilizando um sistema de teste durante a produção para calibração do TIADC.	Não requer esforço computacional adicional no TIADC.	Incapaz de se adaptar a quaisquer alterações nos desvios.

Fonte: Modificado de (PARKEY; MIKHAEL, 2013)

### 2.4.1 OFFSET

O desvio de *offset* é um dos mais simples de ser compensado digitalmente, pois requer somente o uso de somadores (VOGEL; JOHANSSON, 2006). O *offset* pode ser estimado a partir do valor médio do sinal de entrada de cada sub-ADC (VOGEL, 2005). O uso de um circuito de *chopper* randômico na entrada do TIADC para estimativa e compensação do *offset* é apresentado em (JAMAL et al., 2002) e em (EKLUND; GUSTAFSSON, 2000). O circuito analógico do *chopper* de entrada foi implementado em circuitos integrados em ambos os casos. Porém os circuitos digitais de compensação foram implementados em Matlab ou em DSP.

### 2.4.2 GANHO

A compensação de ganho requer somente um multiplicador no canal à ser compensado (VOGEL; JOHANSSON, 2006). Métodos adaptativos, para operação em *background*, foram propostos na literatura para estimar o desvio de ganho em (JAMAL et al., 2004), (DYER et al., 1998), (FU et al., 1998). Jamal propôs um método de compensação e estimação do desvio totalmente digital. A implementação do método porém foi realizada somente em *software*. Em (DYER et al., 1998) o compensador de ganho é implementado utilizando circuitos mistos para um TIADC com frequência de amostragem de 40 MHz. Os mesmos autores de (DYER et al., 1998) implementam métodos de compensação de ganho totalmente digitais em (FU et al., 1998). Nos dois casos os métodos foram implementados em circuitos integrados. Uma abordagem diferente, baseada em propriedades ortogonais dos sinais, é apresentada em (FERRÉ et al., 2008).

### 2.4.3 CLOCK SKEW

Os métodos para compensação do *clock skew* requerem soluções de maior complexidade, e apresentam seus próprios desafios (RAZAVI, 2013). Apesar de ser possível eliminar o *clock skew* com o uso de um *sample-and-hold* na entrada do TIADC (DYER et al., 1998), (SUMANEN et al., 2001), esta solução implica em aumento de potência e em aumento de ruído no sinal do TIADC (STEPANOVIC, 2012). Para estimar e calibrar o *clock skew* (POULTON et al., 2003) utiliza a injeção de um sinal conhecido no TIADC, e obtém estimativas do desvio através da FFT (*Fast Fourier Transform*) do sinal amostrado. *Advanced Filter Bank (AFB)* foi utilizado em (SEO et al., 2005) como um método *offline* para compensação de *clock skew*. Apesar de ter sido utilizado comercialmente com sucesso (LOONEY, 2003), o método requer múltiplos filtros FIR (*Finite Impulse Response*) longos (até 61-taps em (SEO

et al., 2005)). Em (SEO et al., 2005) os métodos de compensação são implementados em *software* somente. O método também possui o seu desempenho degradado devido a variações que possam ocorrer no *clock skew* ao longo do tempo.

Em (JAMAL et al., 2002) um filtro de *delay* fracionário é utilizado para compensação dos desvios. Johansson (JOHANSSON; LOWENBORG, 2002), (JOHANSSON et al., 2006) construiu a base teórica para o uso de filtros de *delay* fracionário, e da estrutura de Farrow (FARROW, 1988), para compensação do *clock skew*. O uso de filtros de *delay* fracionário e da estrutura de Farrow, que permite o ajuste do desvio de tempo sem o recálculo de coeficientes de filtros, é um dos métodos mais populares para a reconstrução da saída do TIADC em virtude do *clock skew*. O método proposto por Johansson entretanto exige a operação do TIADC em *oversampling*. Este é no máximo igual a 2 e ocorre quando o TIADC possui um número par de ADCs. Em (RAZAVI, 2013) um estimador de baixa complexidade de implementação, baseado na autocorrelação do sinal de saída do TIADC, é utilizado em conjunto de um VDL (*Variable Delay Line*). Razavi implementou em um circuito integrado a compensação utilizando VDL. Entretanto, o estimador foi implementado em Matlab, utilizando a saída de um TIADC após o *downsampling* do seu sinal de saída. O ajuste analógico do *delay* no *sample-and-hold* em cada sub-ADC individual também é utilizado em (EL-CHAMMAS; MURMANN, 2011) e em (HUANG et al., 2011). A vantagem deste método sobre a compensação digital está no menor consumo de potência das estruturas VDL. Entretanto o VDL adiciona ruído na fase do sinal, o que causa um efeito no espectro do TIADC similar ao efeito do *jitter*. Os menores níveis de controle destas estruturas são da ordem de femtosegundos, o que pode ser relativamente complexo de ser obtido através de circuitos analógicos.

Os métodos de compensação digitais possuem as vantagens dos sistemas digitais de processamento de sinal. Em especial pode-se citar o controle do ruído adicionado ao sinal e a invariabilidade do ruído em função de interferências externas ao circuito, tais como variações de temperatura e tensões de alimentação. A tendência na utilização de sistemas digitais para melhorar o desempenho de conversores analógico-digital já foi apontada em (ROERMUND et al., 2005), denominado-os como conversores inteligentes. A compensação do desvio de *clock skew* através de filtros de *delay* fracionário é um dos principais exemplos dos ganhos obtidos com o uso de conversores inteligentes.

### 3 FILTROS DE DELAY FRACIONÁRIO

#### 3.1 FILTROS DE DELAY FRACIONÁRIO

Considere o sinal contínuo no tempo  $x(t)$  como entrada de um filtro  $h(t)$  ideal, que atrasa o sinal  $x(t)$  por um intervalo de tempo  $t_l$  em sua saída  $y(t)$ . A saída  $y(t)$  deste filtro é expressa pela equação (19).

$$y(t) = x(t - t_l). \quad (19)$$

Agora, considere que o sinal  $x(t)$  é uniformemente amostrado, com um intervalo de tempo  $T_s$ . Sendo  $h(nT_s)$  um filtro discreto que adiciona um atraso (*delay*)  $D$  ao sinal  $x(nT_s)$ , a saída  $y(nT_s)$  deste filtro será dada por:

$$y(nT_s) = x((n - D)T_s). \quad (20)$$

Onde  $n$  representa o índice da amostra. Um valor de *delay* inteiro para este filtro, múltiplo do intervalo de amostragem  $T_s$ , é facilmente obtido através de elementos de memória das amostras anteriores (DIAZ-CARMONA; DOLECEK, 2011). Mas, se ao invés de um *delay* inteiro  $D$  for necessário um *delay* não múltiplo do intervalo de amostragem, a saída  $y(nT_s)$  será um valor entre duas amostras de  $x(t)$ . Por não possuir este valor entre as amostras de  $x(nT_s)$ , o valor  $y(nT_s)$  somente pode ser obtido através da interpolação das amostras de  $x(nT_s)$  (LAAKSO et al., 1996). A interpolação das amostras pode ser realizada por um filtro de *delay* fracionário (FDF), um tipo de filtro digital capaz de atrasar um sinal por um intervalo de tempo não múltiplo do intervalo de amostragem. Idealmente este filtro deve manter a amplitude do sinal e deslocá-lo de uma unidade de tempo fracionária  $D$ . Dada a transformada de Fourier de  $x(t)$  como  $X(j\omega)$ , pode-se representar a transformada de Fourier  $Y(j\omega)$  de  $y(t)$  como:

$$Y(j\omega) = X(j\omega)e^{-j\omega D}. \quad (21)$$

A função de transferência  $H(j\omega)$  deste filtro será portanto:

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = \frac{X(j\omega)e^{-j\omega D}}{X(j\omega)} = e^{-j\omega D}. \quad (22)$$

Da equação (22) pode-se concluir que a magnitude do filtro é sempre unitária e que a fase deverá ser linear, sendo dada por:

$$\Phi_{H(j\omega)} = -D\omega. \quad (23)$$

E o atraso de fase  $\tau_{H(j\omega)}$  deste filtro é expresso pela equação (24). O atraso de fase é uma medida útil para expressar o tempo de atraso que um sinal de entrada sofre quando aplicado em um sistema linear.

$$\tau_{H(j\omega)} = -\frac{\Phi_{H(j\omega)}}{\omega} = -\frac{-D\omega}{\omega} = D. \quad (24)$$

Se aplicada a transformada inversa de Fourier sobre  $H(j\omega)$  é obtido :

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) \cdot e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-j\omega D} \cdot e^{j\omega n} d\omega, \quad (25)$$

e resolvendo a integral de (25),

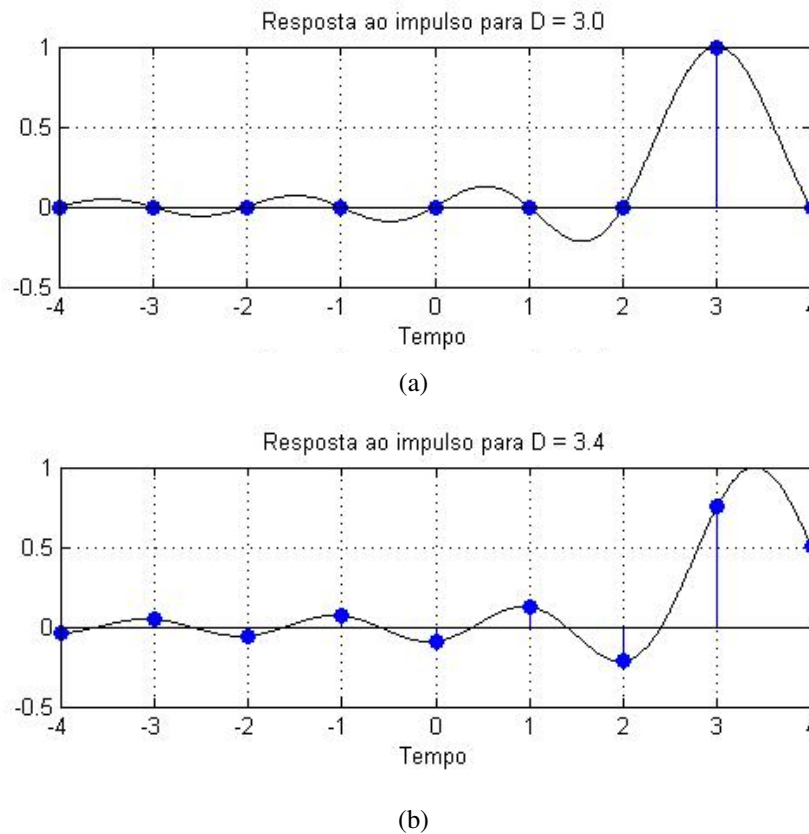
$$h(n) = \frac{1}{2\pi} \left. \frac{e^{j\omega(n-D)}}{j(n-D)} \right|_{-\pi}^{\pi} = \frac{1}{2\pi} \frac{e^{j\pi(n-D)} - e^{-j\pi(n-D)}}{j(n-D)}. \quad (26)$$

Utilizando a fórmula de *Euler* em (26) é então obtido:

$$h(n) = \frac{\text{sen}(\pi(n-D))}{\pi(n-D)} = \text{sinc}(n-D). \quad (27)$$

De (27) pode-se concluir que, para valores fracionários de  $D$ , o filtro ideal  $h(n)$  é não causal, já que o filtro possui infinitos valores diferentes de zero para  $n < 0$ , tornando-o irrealizável. Entretanto, para valores inteiros e positivos de  $D$ , o filtro  $h(n)$  somente é diferente de zero para  $n = D$ , justificando assim porque *delays* inteiros são causais e relativamente simples de serem implementados. A resposta ao impulso para um valor inteiro e fracionário de  $D$  é visualizada na figura 11. Na figura 11.a) pode ser visualizada a resposta ao impulso para o filtro com *delay* inteiro. Pode-se verificar nesta figura que o sinal de função  $\text{sinc}(n-D)$ , considerando somente os valores inteiros de tempo, é diferente de zero somente no instante de tempo 3, valor do *delay* inteiro. A figura 11.b) apresenta a resposta ao impulso para o filtro com *delay* fracionário. Como pode ser visto na figura, nos intervalos inteiros de tempo o sinal de função  $\text{sinc}(n-D)$  sempre possui um valor diferente de zero.

Sendo o filtro ideal de *delay* fracionário não causal, somente é possível obter um filtro aproximado, contendo um erro com relação ao ideal. O filtro aproximado pode ser projetado



**Figura 11: Resposta ao impulso para filtro de *delay* fracionário ideal com a)  $D = 3.0$  e b)  $D = 3.4$ .**

**Fonte: Autoria própria**

como um filtro de estrutura FIR de ordem  $N$ . O projeto do filtro consiste na minimização de uma função de erro, dada por (VÄLIMÄKI, 1995b):

$$E(e^{j\omega}) = H_{ap}(j\omega) - H(j\omega). \quad (28)$$

Onde  $H_{ap}(j\omega)$  é a função de transferência do filtro aproximado.

### 3.1.1 PROJETO DO FILTRO PELA INTEGRAL DO MÍNIMO ERRO QUADRÁTICO

A forma mais simples de obter uma aproximação do FDF ideal é através do truncamento de sua resposta ao impulso. O truncamento também é interessante por fornecer a solução ótima para a integral de erro quadrática de  $E(e^{j\omega})$  (LAAKSO et al., 1996) (VÄLIMÄKI, 1995b). A integral de erro quadrática pode ser definida como:

$$E_2 = \frac{1}{\pi} \int_0^\pi |E(e^{j\omega})|^2 d\omega = \frac{1}{\pi} \int_0^\pi |H_{ap}(j\omega) - H(j\omega)|^2 d\omega. \quad (29)$$

Utilizando o teorema de Parseval em (29):

$$E_2 = \sum_{-\infty}^{\infty} |h_{ap}(n) - h(n)|^2. \quad (30)$$

Considerando o truncamento da resposta impulsiva do filtro ideal por um filtro de ordem  $N$  pode-se escrever:

$$h_{ap}(n) = \begin{cases} \text{sinc}(n - D) & M_i \leq n \leq M_i + N \\ 0 & \text{demais valores} \end{cases} \quad (31)$$

A variável  $M_i$  representa o índice do primeiro valor diferente de zero na resposta ao impulso do filtro, sendo que  $M_i$  deve ser igual a zero para um filtro causal. Da equação (31) pode-se deduzir que o erro de aproximação, expresso por (30), pode ser reescrito como:

$$E_2 = \sum_{n=-\infty}^{M_i-1} |h(n)|^2 + \sum_{n=M_i+N+1}^{\infty} |h(n)|^2. \quad (32)$$

Da expressão (32) pode-se concluir que: um filtro de ordem mais elevada possui um erro menor e que, o valor da parte inteira do *delay*  $D$  deve coincidir com o centro de gravidade do filtro, próximo do valor de pico de  $h_{ap}$ . O valor da parte inteira de  $D$  é dado por (33).

$$\lfloor D \rfloor = \begin{cases} \frac{N-1}{2} & \text{se } N \text{ ímpar} \\ \frac{N}{2} & \text{se } N \text{ par} \end{cases} \quad (33)$$

Onde  $\lfloor D \rfloor$  denota a parte inteira do *delay*  $D$ . Apesar da simplicidade do projeto do filtro utilizando o truncamento da função  $\text{sinc}(n - D)$ , os filtros obtidos desta forma geralmente não possuem resultados satisfatórios (DIAZ-CARMONA; DOLECEK, 2011). Isto é consequência do fenômeno de *Gibbs* (PROAKIS; MANOLAKIS, 1996), que causa um *ripple* na magnitude da resposta em frequência do filtro. O uso do janelamento, similar ao utilizado para amostragem de um sinal e redução de erros na FFT do sinal, pode minimizar os efeitos indesejados do fenômeno de *Gibbs*.

Na figura 12 é apresentada a magnitude da resposta em frequência de um filtro de ordem 9, projetado utilizando a integral do mínimo erro quadrático (ou simplesmente na sua denominação na língua inglesa *Least Square, LS*). Na figura 13 é apresentado o atraso de fase deste mesmo filtro. Nas duas figuras é possível observar o efeito do fenômeno de *Gibbs* sobre a resposta em frequência. Também é evidente das duas figuras que a aproximação do filtro é



mais próxima do ideal quanto menor é o valor do *delay* fracionário. Pode ser observado na figura 13 que o filtro adiciona um *delay* inteiro ao sinal de entrada. O valor deste *delay* é dado pela equação (33) para  $N = 9$ .

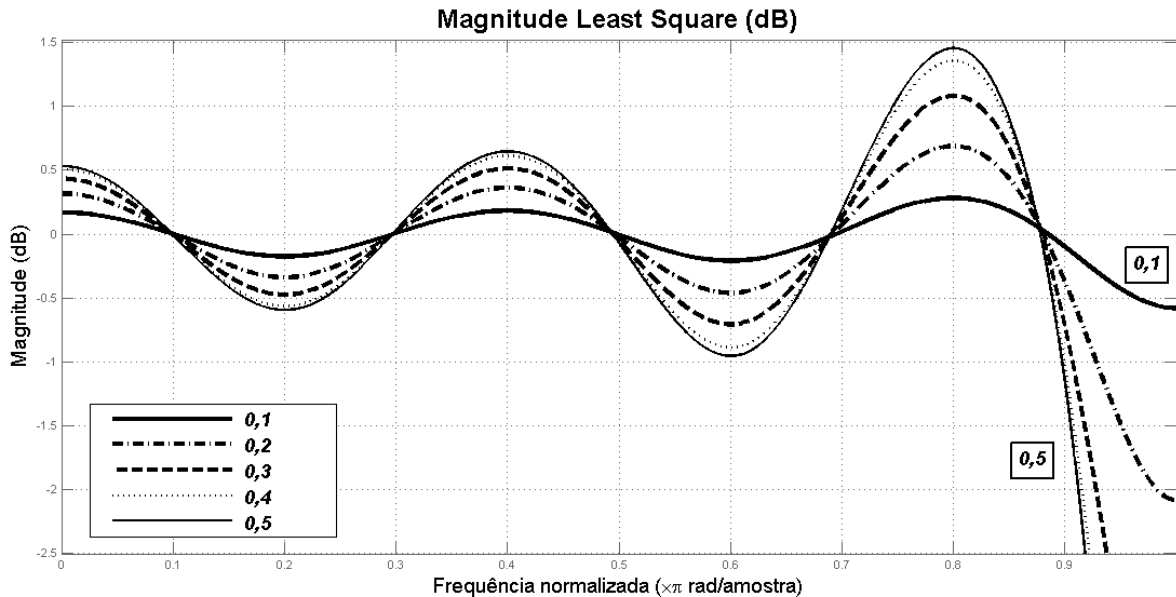


Figura 12: Magnitude de filtro projetado através do método *Least Square*, do *delay* 0,1 ao *delay* 0,5.

Fonte: Autoria própria

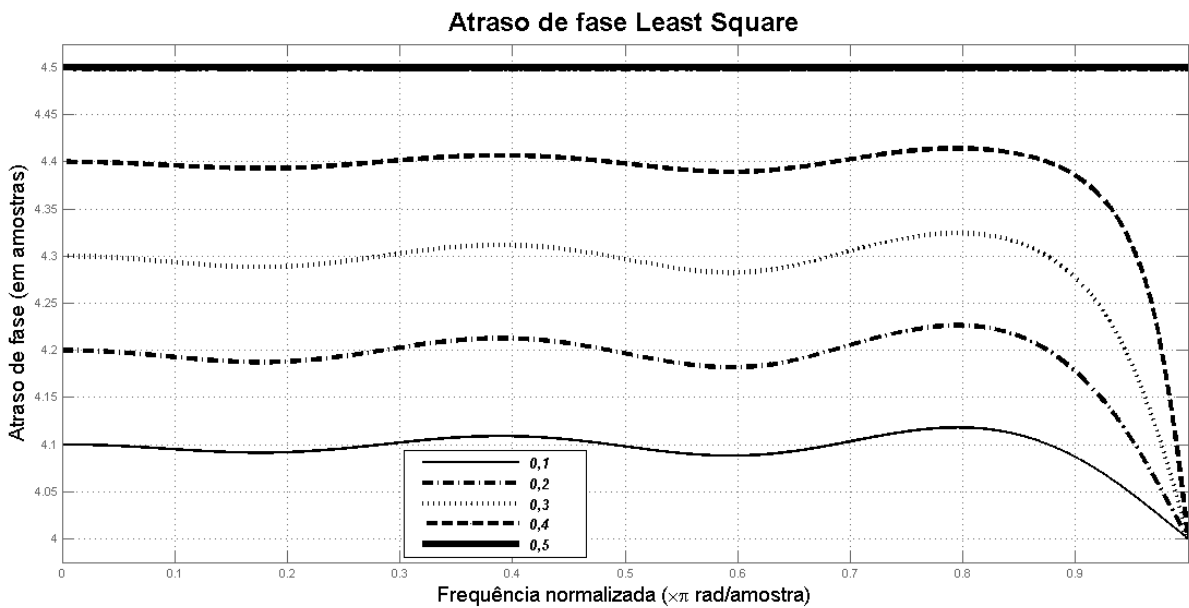


Figura 13: Atrazo de fase do filtro projetado através do método *Least Square*, do *delay* 0,1 ao *delay* 0,5.

Fonte: Autoria própria

### 3.1.2 PROJETO DO FILTRO PELA APROXIMAÇÃO GENERALIZADA DOS MÍNIMOS QUADRADOS

Se for definida uma banda de interesse  $[0, \alpha\pi]$ , e se for aplicada uma função peso  $W(\omega) \in \Re+$  para ponderação em frequência do erro dado por (28), pode-se reescrever (29) como:

$$E_2 = \frac{1}{\pi} \int_0^{\alpha\pi} W(\omega) |H_{ap}(e^{j\omega}) - H(e^{j\omega})|^2 d\omega. \quad (34)$$

Considere que o filtro de aproximado  $H_{ap}$  é dado por:

$$H_{ap}(e^{j\omega}) = \mathbf{h}^T \mathbf{e}. \quad (35)$$

onde,

$$\mathbf{h} = [h(0) \ h(1) \ \dots \ h(N)]^T \quad \mathbf{e} = [1 \ e^{-j\omega} \ \dots \ e^{-jN\omega}]^T. \quad (36)$$

Substituindo (35) em (34) é obtido:

$$\begin{aligned} E_2 &= \frac{1}{\pi} \int_0^{\alpha\pi} W(\omega) [\mathbf{h}^T \mathbf{e} - H(e^{j\omega})] [\mathbf{h}^T \mathbf{e} - H(e^{j\omega})]^* d\omega = \\ &= \frac{1}{\pi} \int_0^{\alpha\pi} W(\omega) [\mathbf{h}^T \mathbf{C} \mathbf{h} - 2\mathbf{h}^T \text{Re}\{H(e^{j\omega})\mathbf{e}^*\} + |H(e^{j\omega})|^2] d\omega \end{aligned} \quad (37)$$

Onde  $\mathbf{C}$  é dado por:

$$\mathbf{C} = \text{Re}\{\mathbf{e}\mathbf{e}^H\} = \begin{bmatrix} 1 & \cos(\omega) & \dots & \cos(N\omega) \\ \cos(\omega) & 1 & \dots & \cos((N-1)\omega) \\ \vdots & \dots & \dots & \vdots \\ \cos(N\omega) & \cos((N-1)\omega) & \dots & 1 \end{bmatrix} \quad (38)$$

Sendo que o sobrescrito  $H$  representa a operação Hermitiana. O erro então pode ser reescrito como:

$$E_2 = \mathbf{h}^T \mathbf{P} \mathbf{h} - 2\mathbf{h}^T \mathbf{p}_1 + p_0 \quad (39)$$

onde,

$$\begin{aligned}
\mathbf{P} &= \frac{1}{\pi} \int_0^{\alpha\pi} W(\omega) \mathbf{C} d\omega \\
\mathbf{p}_1 &= \frac{1}{\pi} \int_0^{\alpha\pi} W(\omega) [\operatorname{Re}H(e^{j\omega})\mathbf{e} - \operatorname{Im}H(e^{j\omega})\mathbf{s}] d\omega \\
p_0 &= \frac{1}{\pi} \int_0^{\alpha\pi} W(\omega) |H(e^{j\omega})|^2 d\omega \\
\mathbf{e} &= [1 \cos(\omega) \dots \cos(N\omega)]^T \\
\mathbf{s} &= [0 \operatorname{sen}(\omega) \dots \operatorname{sen}(N\omega)]^T
\end{aligned} \tag{40}$$

Para encontrar o valor mínimo de erro, a solução ótima, a equação (39) é derivada em relação à  $\mathbf{h}$  e igualada a zero, ou seja:

$$2\mathbf{P}\mathbf{h} - 2\mathbf{p}_1 = 0 \tag{41}$$

A solução da equação (41) é dada formalmente pela inversão da matriz  $\mathbf{P}$ , sendo obtido:

$$\mathbf{h} = \mathbf{P}^{-1}\mathbf{p}_1 \tag{42}$$

Pode-se observar que a solução ótima é obtida a partir da resolução das integrais de (40) e das  $(N + 1)$  equações lineares de (42). As integrais de (40) são significativamente simplificadas quando a função ideal é dada por  $e^{-jD\omega}$  (LAAKSO et al., 1996). No caso em que a função  $W(\omega) = 1$  a solução é dada de forma fechada, sendo que  $\mathbf{P}$  e  $\mathbf{p}_1$  podem ser expressados pelas equações (43) e (44) respectivamente.

$$P_{k,l} = \frac{1}{\pi} \int_0^{\alpha\pi} \cos((k-l)\omega) d\omega = \alpha \operatorname{sinc}(\alpha(k-l)) \quad k, l = 1, 2, \dots, N+1 \tag{43}$$

$$p_{1,k} = \frac{1}{\pi} \int_0^{\alpha\pi} \cos((k-D)\omega) d\omega = \alpha \operatorname{sinc}(\alpha(k-D)) \quad k = 1, 2, \dots, N+1 \tag{44}$$

A principal vantagem do projeto pela aproximação generalizada dos mínimos quadrados (ou na sigla inglesa *GLS*) é a flexibilidade em definir uma banda de interesse e em ponderar o erro do filtro no domínio da frequência. Ao definir uma banda de interesse e uma região em que o erro da aproximação não é importante, é possível obter um erro de aproximação menor do que no projeto de aproximação LS para um filtro de mesma ordem. Nas figuras 14 e 15 são apresentadas a magnitude e o atraso de fase, respectivamente, de filtros projetados utilizando o método GLS para  $\alpha = 0,5$ . Como pode ser observado nas duas figuras, há um

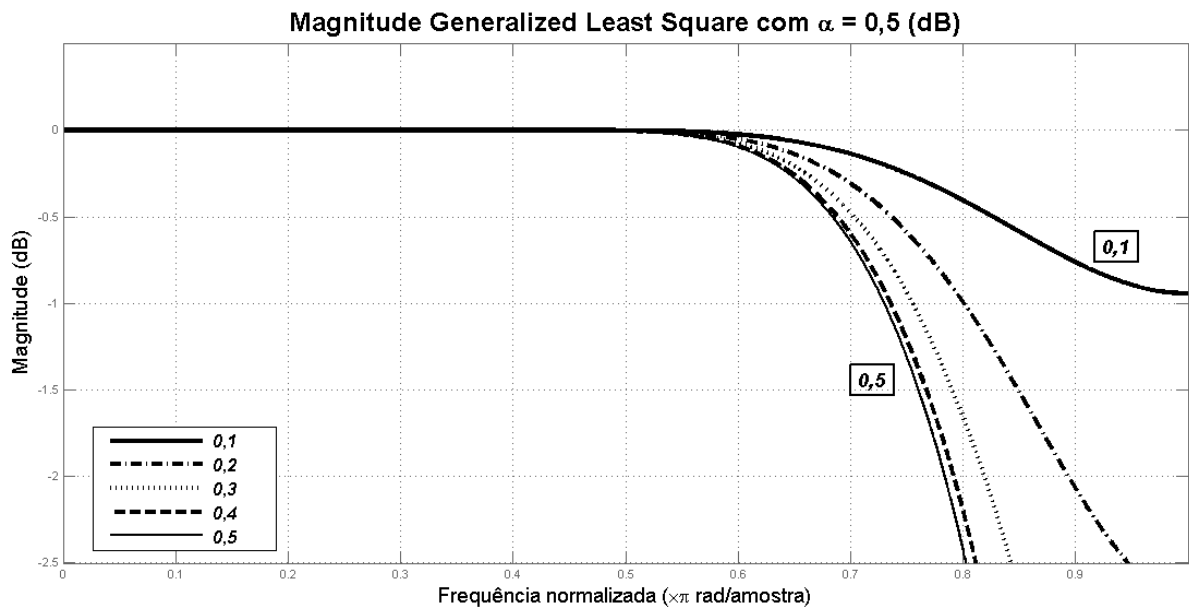


Figura 14: Magnitude de filtros projetados utilizando método GLS para  $\alpha = 0,5$  e *delay* entre 0,1 e 0,5.

Fonte: Autoria própria

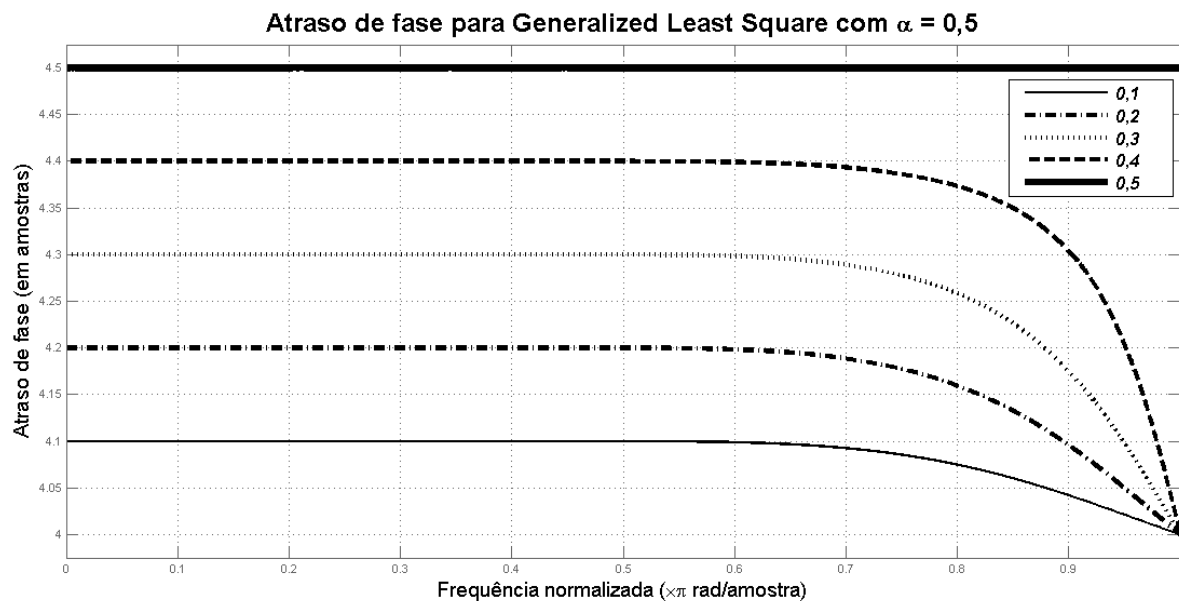


Figura 15: Atraso de fase de filtros projetados utilizando método GLS para  $\alpha = 0,5$  e *delay* entre 0,1 e 0,5.

Fonte: Autoria própria

erro de aproximação dos filtros muito menor na banda definida por  $[0; 0,5\pi]$  do que nas demais frequências.

### 3.1.3 PROJETO DO FILTRO UTILIZANDO MÉTODO MINIMAX

Diferentemente dos métodos *LS* e *GLS*, o método *minimax* não busca reduzir o valor do erro quadrático da aproximação do filtro, mas sim o valor de pico do erro de aproximação, o que é expresso pela equação (45).

$$E_{max} = \min \left\{ \max_{\omega \in [0, \alpha\pi]} \left\{ |E(e^{j\omega})| \right\} \right\} \quad (45)$$

A solução para o problema da equação (45) normalmente é obtido através de métodos iterativos (VÄLIMÄKI, 1995b). Entretanto, em (OETKEN, 1979) é proposto um método para filtros de ordem ímpar que não requer o uso de um método iterativo. A discussão a seguir é adaptada de (VÄLIMÄKI, 1995b). Oetken observou que, calculando os zeros na magnitude de um único filtro ótimo (um filtro de protótipo), é possível utilizar os zeros calculados para projetar outros filtros FIR com características similares. A função de erro para um filtro de ordem  $N$  possui então  $K = (N + 1)/2 - 1$  zeros, ou seja:

$$E(e^{j\omega}) = H_{ap}(e^{j\omega}) - H(e^{j\omega}) = 0, \quad \omega = \Omega_k, \quad k = 1, 2, \dots, K \quad (46)$$

o que implica em

$$\sum_{n=0}^N h(n)e^{-jn\Omega_k} = e^{-j\Omega_k \frac{N}{2}}, \quad k = 1, 2, \dots, K \quad (47)$$

Em (47) o valor  $N/2$  é o *delay* do filtro. A solução para os coeficientes do filtro pode ser encontrada se o valor do *delay*  $D$  for um valor próximo de  $N/2$ , o que é expresso pela equação (48).

$$\mathbf{E}_{\Omega} \mathbf{h} = \mathbf{e}_D \quad (48)$$

Sendo  $\mathbf{e}_D$  o vetor dado pela equação (50) e  $\mathbf{E}_{\Omega}$  a matriz de  $K$  linhas por  $(N + 1)$  colunas dada por (49).

$$\mathbf{E}_{\Omega} = \begin{bmatrix} 1 & e^{-j\Omega_1} & e^{-j2\Omega_1} & \dots & e^{-jN\Omega_1} \\ 1 & e^{-j\Omega_2} & e^{-j2\Omega_2} & \dots & e^{-jN\Omega_2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & e^{-j\Omega_K} & e^{-j2\Omega_K} & \dots & e^{-jN\Omega_K} \end{bmatrix} \quad (49)$$

$$\mathbf{e}_D = \left[ e^{-jD\Omega_1} \quad e^{-jD\Omega_2} \quad \dots \quad e^{-jD\Omega_K} \right]^T \quad (50)$$

Um conjunto igual à  $K$  equações complexas, com  $L = 2K$  variáveis desconhecidas, pode ser obtido da equação (48). As  $L$  variáveis desconhecidas podem ser determinadas da parte real e imaginária de  $E_\Omega$  e  $e_D$ , se  $\mathbf{E}_\Omega = \mathbf{C}_\Omega - j\mathbf{S}_\Omega$  e  $\mathbf{e}_D = \mathbf{c}_D - js_D$ , a equação (48) pode ser reescrita como:

$$\mathbf{P}_\Omega \mathbf{h} \equiv \mathbf{p}_\Omega \quad (51)$$

Onde,

$$\mathbf{P}_\Omega = \begin{bmatrix} \mathbf{C}_\Omega \\ \mathbf{S}_\Omega \end{bmatrix} \quad (52)$$

e,

$$\mathbf{p}_\Omega = \begin{bmatrix} c_D \\ s_D \end{bmatrix}. \quad (53)$$

Pode-se deduzir da equação (51) que o vetor de coeficientes do filtro FIR pode ser dado por:

$$\mathbf{h} = \mathbf{P}_\Omega^{-1} \mathbf{p}_\Omega \quad (54)$$

Nas figuras 16 e 17 são ilustrados a magnitude e o atraso de fase, respectivamente, de filtros de ordem 9 projetados utilizando o método Minimax para uma banda de  $0,5\pi$ . Pode-se observar pela resposta em frequência que o método Minimax é equiparável ao GLS com banda de  $0,5\pi$ .

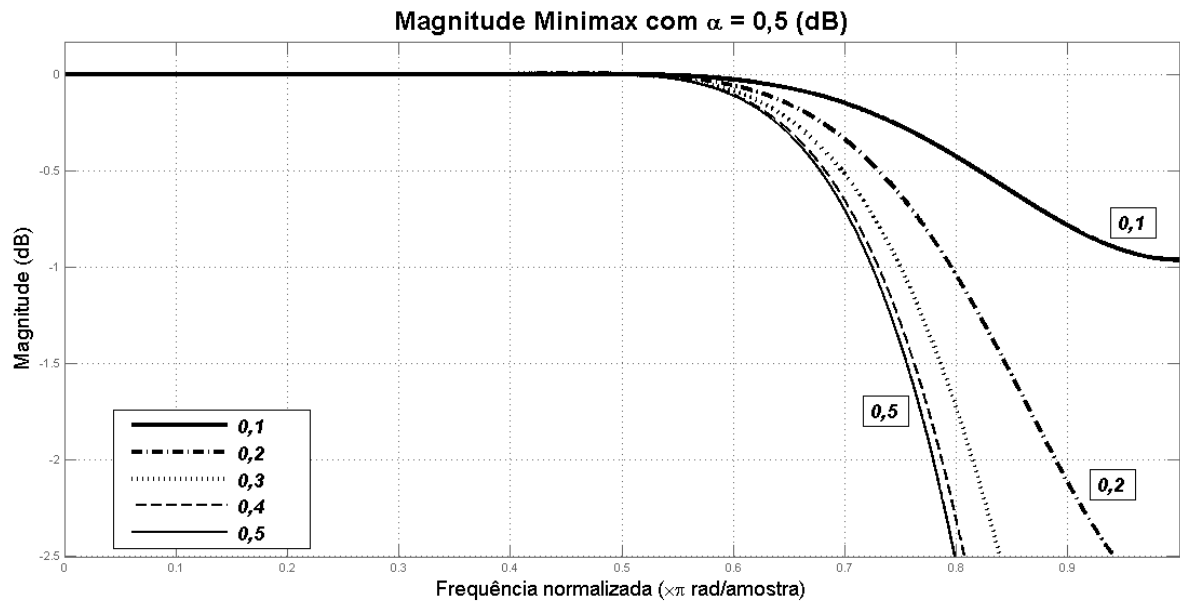


Figura 16: Magnitude de filtros projetados utilizando método Minimax para banda de  $0,5\pi$  e *delay* entre  $0,1$  e  $0,5$ .

Fonte: Autoria própria

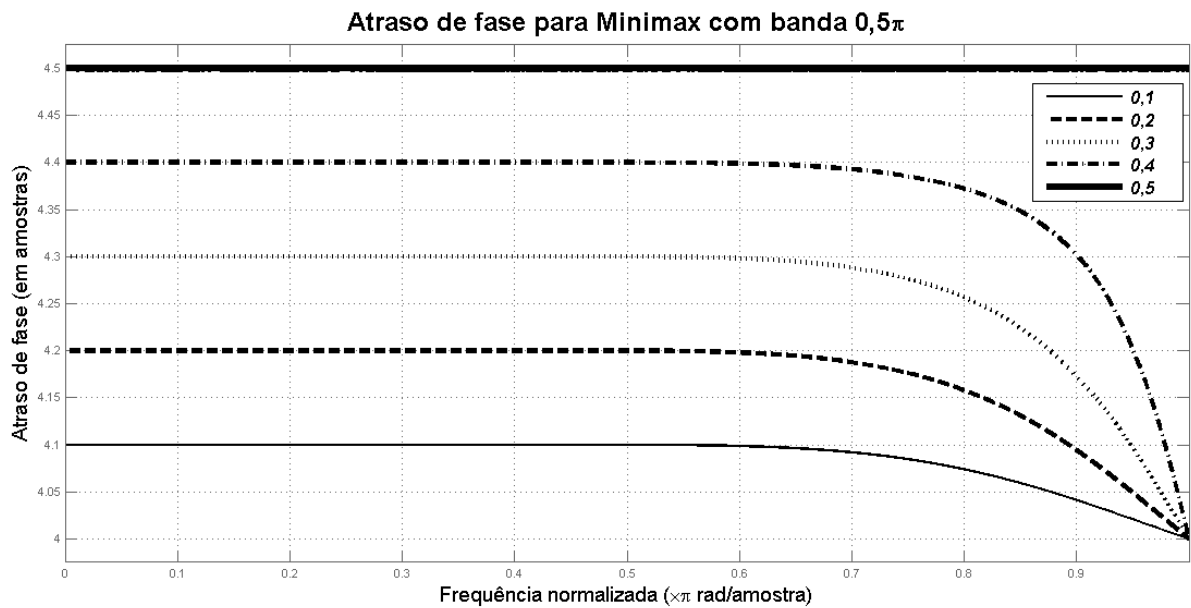


Figura 17: Atraso de fase de filtros projetados utilizando método Minimax para banda de  $0,5\pi$  e *delay* entre  $0,1$  e  $0,5$ .

Fonte: Autoria própria

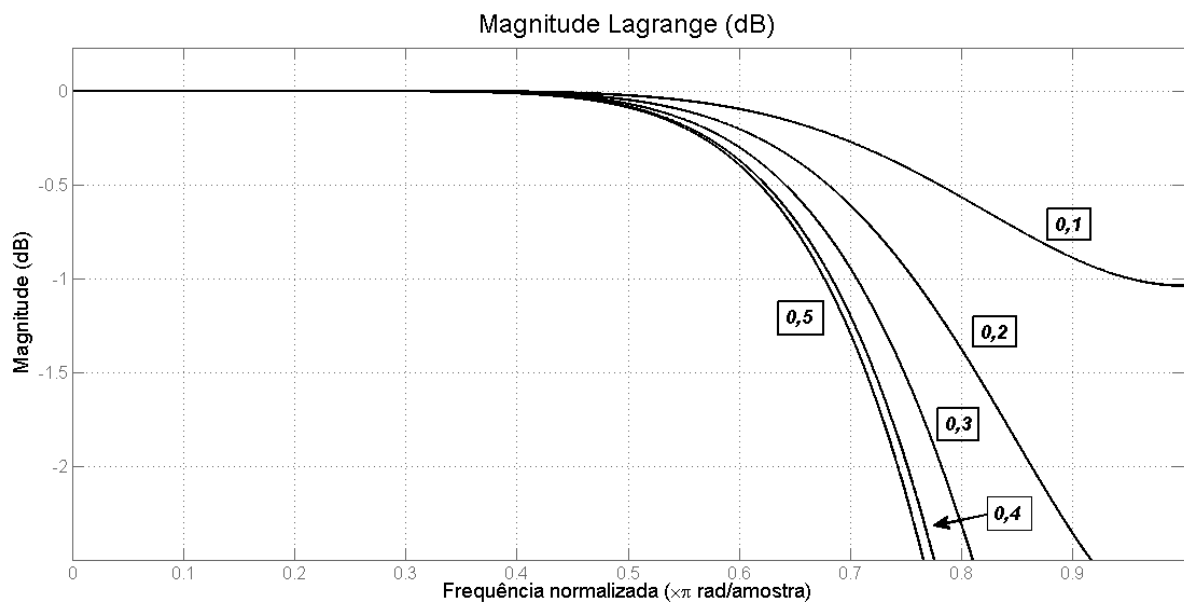
### 3.1.4 PROJETO DO FILTRO UTILIZANDO INTERPOLAÇÃO DE LAGRANGE

O método de projeto de filtro através de Lagrange é o homônimo da interpolação polinomial de Lagrange. Os coeficientes do filtro FIR são obtidos de uma equação de forma

fechada, dada pela equação (55), onde  $N$  é a ordem do filtro e  $D$  é o *delay* total do filtro (parte inteira mais fracionária).

$$h(n) = \prod_{k=0, k \neq n}^N \frac{D-k}{n-k} \quad n = 0, 1, 2, \dots, N. \quad (55)$$

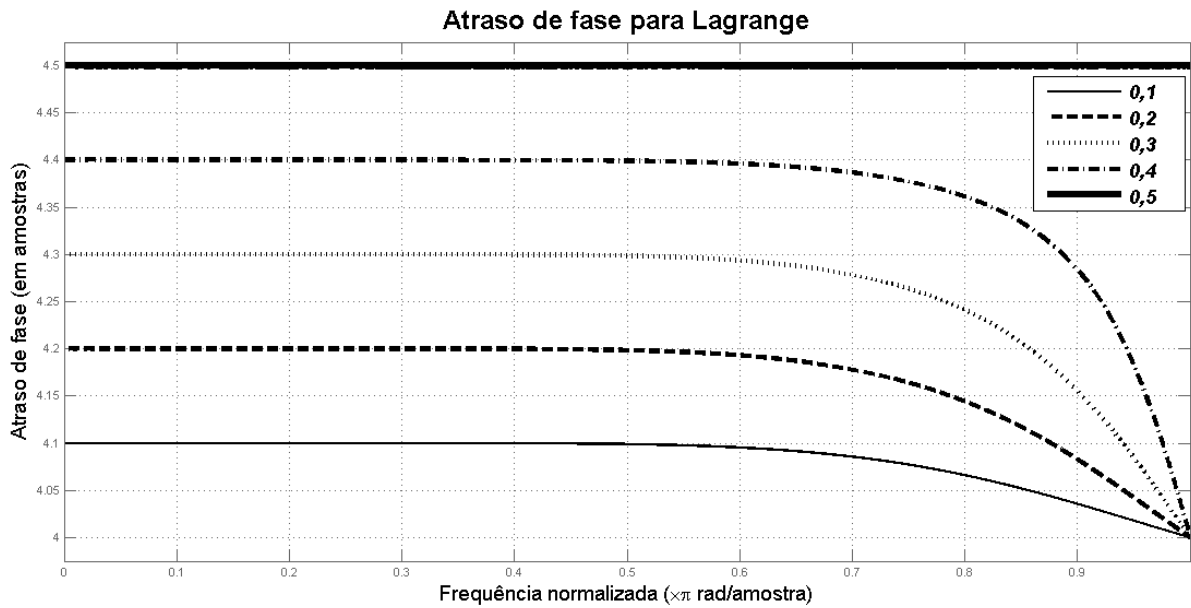
O método de Lagrange possui algumas vantagens, entre elas: simplicidade de projeto do filtro, a resposta do filtro em baixas frequências é completamente plana, o filtro obtido é sempre *passivo* (magnitude sempre menor ou igual a 1), característica importante em sistemas que utilizem o filtro em malhas de *feedback* (LAAKSO et al., 1996). A desvantagem do método de Lagrange está na ausência de controle sobre a resposta em frequência do filtro projetado e na banda obtida do filtro. Um estudo amplo do método de Lagrange é realizado em (VÄLIMÄKI, 1995b). Em (VÄLIMÄKI, 1995b) é sugerida a aplicação deste filtro para modelagem de ondas sonoras em tubos, enquanto em (ERUP et al., 1993) o filtro é sugerido como interpolador para sincronização de símbolos em modems. A figura 18 apresenta a magnitude de filtros de *delay* entre 0,1 e 0,5 utilizando o método de Lagrange para um filtro de ordem  $N = 9$ . Como pode ser observado nesta figura, a resposta do filtro é plana, completamente sem oscilações. Entretanto, comparando com os filtros obtidos através do método GLS e Minimax, o filtro obtido com o método de Lagrange possui uma banda menor. Na figura 19 é apresentado o atraso de fase dos filtros apresentados na figura 18.



**Figura 18: Magnitude de filtros projetados utilizando Lagrange para *delay* entre 0,1 e 0,5.**

**Fonte: Autoria própria**





**Figura 19:** Atraso de fase de filtros projetados utilizando Lagrange para *delay* entre 0,1 e 0,5.

**Fonte:** Autoria própria

### 3.1.5 SUMÁRIO DO PROJETO DE FILTROS DE DELAY FRACIONÁRIO

Foram apresentados nas subseções anteriores o projeto de FDF do tipo FIR através dos métodos LS, GLS, Minimax e Lagrange. Outros métodos de projeto são encontrados na literatura, sendo sugeridas as referências (LAAKSO et al., 1996), (ERUP et al., 1993) e (VÄLIMÄKI, 1995b). Na tabela 3 é apresentado um comparativo entre estes 4 métodos. São analisados os parâmetros de especificação do filtro, a forma de cálculo dos coeficientes e as características da resposta em frequência do filtro.

**Tabela 3: Tabela comparativa dos métodos de projeto de FDF.**

Método	Parâmetros	Cálculo dos coeficientes	Principais características
<i>Least Square</i> (LS)	Ordem $N$ do filtro, <i>Delay</i> $D$ .	Truncamento da função $\text{sinc}(n - D)$ .	Há um <i>ripple</i> na magnitude da resposta em frequência devido ao truncamento. Normalmente não é aplicável.
<i>Generalized Least Square</i> (GLS)	Ordem $N$ do filtro, <i>Delay</i> $D$ , banda passante.	Pode ser simplificado para o uso de equações de forma fechada.	Possui pequena oscilação na banda passante, mesmo quando comparado com outros filtros contendo a mesma especificação.
Minimax	Ordem $N$ do filtro, <i>Delay</i> $D$ , banda passante.	O cálculo dos coeficientes é realizado através da solução de uma equação matricial.	Resposta em frequência similar ao GLS, entretanto pode apresentar maior ondulação na banda passante se mau especificado.
Lagrange	Ordem $N$ do filtro, <i>Delay</i> $D$ .	Utiliza equações fechadas simples, que podem inclusive ser utilizadas em sistemas de tempo real.	A magnitude da resposta em frequência é plana nas baixas frequências, possuindo um erro pequeno com relação ao filtro ideal. Entretanto a banda do filtro é muito limitada.

Fonte: Autoria própria

### 3.2 ESTRUTURA DE FARROW

Na seção anterior foram apresentados vários métodos para o projeto do filtro de *delay* fracionário. Indiferente do método utilizado, o filtro obtido possui um *delay* fixo e idealmente constante. Em diversas aplicações destes filtros o *delay* necessário somente é conhecido durante a operação do sistema, ou então o valor do *delay* necessário sofre variações ao longo do tempo. Isto exige o reprojeto do filtro para cada valor de *delay*, o que pode ser complexo de ser realizado em aplicações de tempo real. Uma alternativa ao reprojeto do filtro é armazenar os coeficientes de um conjunto de  $Q + 1$  filtros, com diferentes *delays*, em memória. Esta abordagem entretanto irá quantizar o *delay* em  $Q + 1$  valores e irá requerer uma grande quantidade de memória para

armazenar os coeficientes.

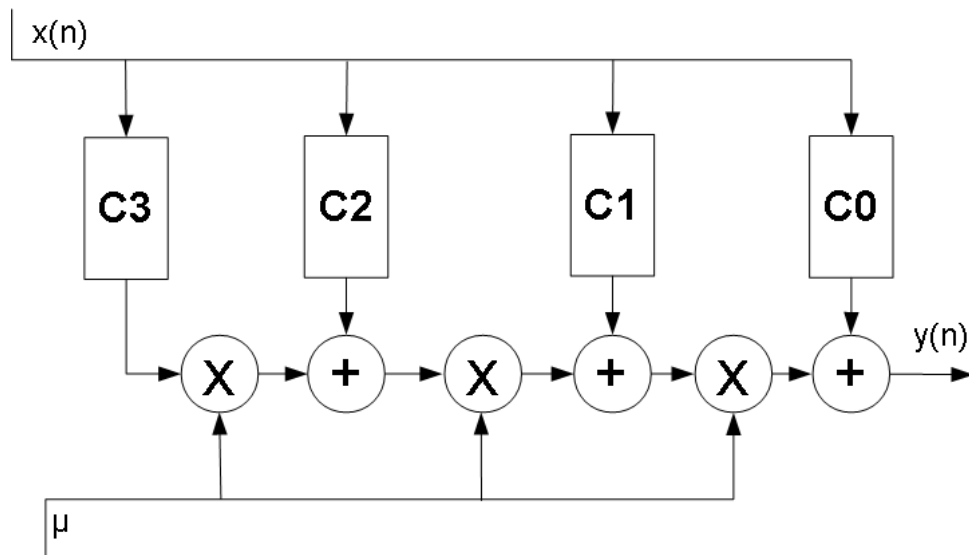
Farrow (1988) propôs uma solução para esta questão com um filtro FIR, em que cada coeficiente deste filtro é uma função do *delay* fracionário  $\mu$ . Este filtro é dado como um polinômio em  $\mu$  de ordem  $M_O$ , em que cada um dos  $M_O + 1$  coeficientes representam um filtro FIR de ordem  $N$ . Desta forma pode-se representar os  $M_O + 1$  coeficientes do filtro  $h_m(n)$  proposto como:

$$h_m(n) = \sum_{m=0}^{M_O} C_m(n) \mu^m, n = 0, 1, 2, \dots, N. \quad (56)$$

Onde  $C_m(n)$  são os  $M_O + 1$  filtros FIR de ordem  $N$ . A equação (56) sugere o uso da regra de *Horner* (PROAKIS; MANOLAKIS, 1996) para o cálculo do sinal de saída do filtro. O uso desta regra para um polinômio qualquer em  $x$ , de coeficientes  $a_n$ , é demonstrado na equação (57).

$$y(x) = a_0 + a_1x + a_2x^2 + a_3x^3 = a_0 + x(a_1 + x(a_2 + a_3x)) \quad (57)$$

Da equação (56), e da utilização da regra de Horner, pode-se deduzir o diagrama de blocos da figura 20, que representa a conhecida estrutura de *Farrow*.



**Figura 20: Estrutura de Farrow**

**Fonte: Autoria própria**

A principal vantagem da estrutura de Farrow é o uso de coeficientes fixos para todos os valores de *delay* fracionário que estão entre  $(-1/2; 1/2)$ . Para o projeto dos coeficientes

da estrutura de Farrow são sugeridos vários métodos na literatura. Nesta dissertação serão abordados dois deles: o  $L_2$  Laakso (LAAKSO et al., 1996) (VESMA, 1999), e a estrutura de *Farrow Lagrange* (VÄLIMÄKI, 1995a). Outros métodos de projeto são apresentados em (VESMA, 1999) e (JOHANSSON; LOWENBORG, 2003). Estes métodos foram escolhidos devido a flexibilidade do método de Laakso e a similaridade da resposta em frequência do método de Farrow Lagrange com o filtro de Lagrange.

No método  $L_2$  Laakso o projeto dos coeficientes da estrutura de Farrow é dividido em duas etapas. Na primeira etapa um conjunto de  $Q + 1$  filtros de *delay* fracionário, todos de ordem  $N$ , são projetados dentro de uma faixa de *delays* dada por  $[\mu_{min}; \mu_{max}]$ . A saída da primeira etapa é um conjunto de coeficientes  $h_{\mu,q}(n)$ , onde  $q = 0, 1, 2, \dots, Q$ ;  $n = 0, 1, 2, \dots, N$ , e  $\mu$  representa o *delay* do filtro correspondente. Na segunda etapa é obtido um polinômio, de ordem  $M_O$  em  $\mu$ , para os coeficientes com o mesmo valor de  $n$ , utilizando um ajuste de curva pelo método dos mínimos quadrados. Os valores obtidos desta aproximação tornam-se então os coeficientes dos  $M_O + 1$  filtros FIR da estrutura de Farrow. No método de Laakso qualquer método de aproximação dos filtros de *delay* fracionário pode ser utilizado, e a aproximação do polinômio em  $\mu$  pode ser facilmente controlada pela definição dos valores limítrofes do *delay* fracionário.

A estrutura de Farrow Lagrange é derivada da interpolação de Lagrange e foi proposta em (VÄLIMÄKI, 1995a). A dedução do cálculo dos coeficientes é apresentada abaixo, tendo sido adaptada de (VÄLIMÄKI, 1995a). Considere a resposta do filtro da equação (56) representada no domínio  $z$ , então:

$$H_m(z) = \sum_{k=0}^{M_O} C_{m,k}(z) D^k, \quad (58)$$

sendo  $C_{m,k}(z)$  polinomiais em  $z$  e  $D$  o valor do *delay* do filtro. Para valores inteiros de  $D$  a resposta do filtro é dada por:

$$H_m(z) = z^{-D}, \quad D = 0, 1, 2, \dots, M. \quad (59)$$

Considerando os  $M_O + 1$  valores inteiros de  $D$ , pode-se obter um sistema de equações de  $M_O + 1$  equações a partir da equação (60), obtida das equações (58) e (59).

$$\sum_{k=0}^{M_O} C_{m,k}(z) D^k = z^{-D}, \quad D = 0, 1, 2, \dots, M. \quad (60)$$

Este sistema de equações pode ser expresso matricialmente, como:

$$\mathbf{V}\mathbf{c} = \mathbf{z} \quad (61)$$

sendo  $\mathbf{V}$  dado pelo matriz de Vandermonde na equação (62).

$$\mathbf{V} = \begin{bmatrix} 0^0 & 0^1 & 0^2 & \dots & 0^M \\ 1^0 & 1^1 & 1^2 & \dots & 1^M \\ 2^0 & 2^1 & 2^2 & \dots & 2^M \\ \vdots & \vdots & \vdots & \dots & \vdots \\ M^0 & M^1 & M^2 & \dots & M^M \end{bmatrix} \quad (62)$$

E  $\mathbf{c}$  e  $\mathbf{z}$  são definidos como:

$$\mathbf{c} = [C_0(z) \ C_1(z) \ C_2(z) \ \dots \ C_{M_O}(z)]^T \quad (63)$$

$$\mathbf{z} = [1 \ z^{-1} \ z^{-2} \ \dots \ z^{-M_O}]^T \quad (64)$$

A solução da equação (61) pode ser obtida isolando-se o vetor  $\mathbf{c}$ , sendo então que os coeficientes são dados pelo cálculo da matriz inversa de Vandermonde  $\mathbf{P} = \mathbf{V}^{-1}$ . As linhas de  $\mathbf{P}$  são os coeficientes de cada um dos  $M_O + 1$  filtros FIR da estrutura de Farrow, ou seja,  $C_m(z)$  será:

$$C_m(z) = \sum_{k=0}^{M_O} p_m(k)z^{-k} \quad m = 1, 2, \dots, M, \quad (65)$$

onde  $p_m$  são obtidos das  $m$  linhas da matriz  $P$ .

A estrutura de Farrow Lagrange apresenta respostas em frequência similares as do filtro de Lagrange, entretanto o número de adições e multiplicações necessários para o mesmo *delay* (considerando a necessidade de os coeficientes serem recalculados) é menor (VÄLIMÄKI, 1995a). Deve-se observar que o número de coeficientes resultantes da estrutura é dado por  $M_O \cdot (M_O - 1)$ , sendo que  $C_0(z)$  sempre será igual à 1. A estrutura de Farrow Lagrange possui uma versão modificada que permite a construção de um filtro de segunda ordem extremamente eficiente. A eficiência é verificada pelo fato de que somente os valores 1 e 1/2 estão entre os coeficientes do filtro, o que torna simples a otimização através de somadores e operadores de deslocamento (VÄLIMÄKI, 1995a). Este ganho de eficiência entretanto somente é verificado para os filtros de segunda e terceira ordem.

A estrutura de Farrow Lagrange apresenta resposta em frequência similar ao filtro de Lagrange, porém com a vantagem do *delay* configurável proporcionada pela estrutura de Farrow. Em contrapartida, a estrutura  $L_2$  Laakso permite flexibilizar o projeto da estrutura de Farrow. Nesta estrutura é possível selecionar o filtro, e conseqüentemente a resposta em frequência, mais adequada para a aplicação. A estrutura também traz a flexibilidade na faixa de *delay* abrangida pelo filtro.

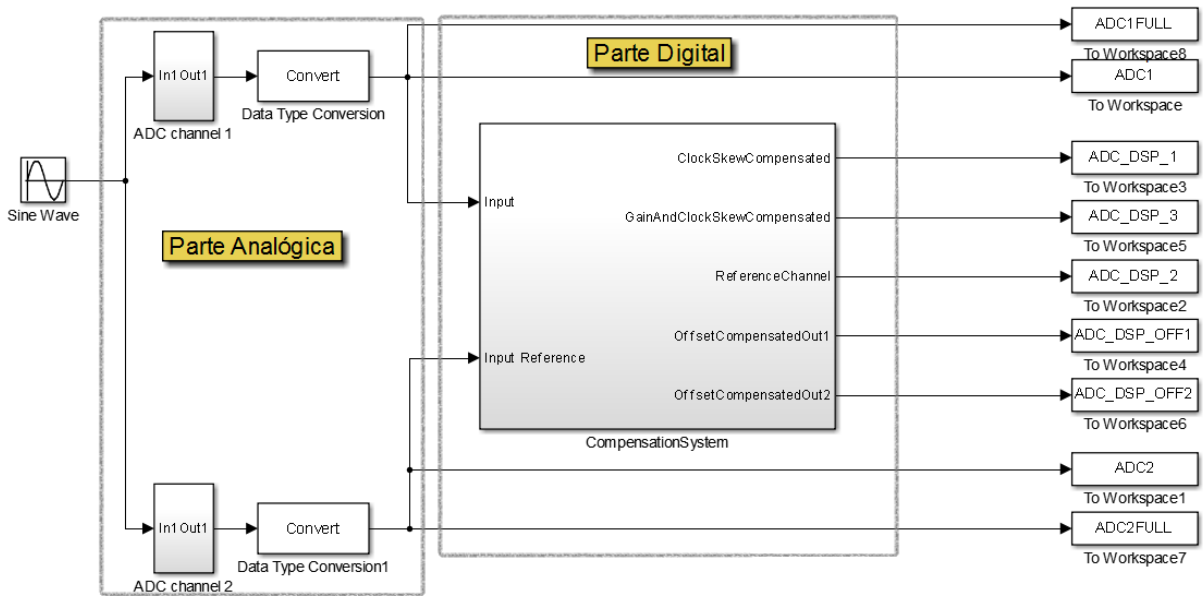
## 4 MODELAGEM DO SISTEMA E METODOLOGIA APLICADA

Neste capítulo será apresentada a modelagem do TIADC e de seu sistema de compensação de desvios. Também será apresentada a metodologia para implementação e validação em FPGA dos sistemas digitais de compensação.

Toda a modelagem do TIADC, assim como dos compensadores de desvios, é realizada através de diagramas de bloco no *software Simulink* (MATLAB; SIMULINK, 2014). A modelagem do sistema foi concebida para permitir a simulação da operação estática (sem alteração de parâmetros ao longo da simulação) de um TIADC de dois canais, contendo os efeitos dos desvios de *offset*, ganho e *clock skew*. Os efeitos do *jitter* também são emulados pelo sistema. Cada uma destas não idealidades pode ser controlada através de parametrizações realizadas no modelo.

Além da modelagem do TIADC e suas não idealidades, foram modelados sistemas digitais para compensação destas não idealidades. A modelagem dos compensadores visa não somente a sua simulação comportamental, mas também a implementação destes sistemas em FPGA. Futuramente estes modelos serão inseridos em um projeto de circuito integrado de um TIADC. Sendo então este modelo uma abstração do TIADC, pode-se separá-lo em duas partes: *analógica* e *digital*. A parte *analógica* compreende o modelo dos circuitos de aquisição de sinal pelo TIADC, o processo de conversão analógico-digital em si. A parte *digital* é reservada aos compensadores de desvio. Esta última parte é implementada e validada em FPGA, enquanto a primeira parte somente é simulada através de *software*. A modelagem de ambas as partes é tratada em maiores detalhes nas seções subsequentes. O diagrama em blocos da figura 21 permite visualizar esta divisão do modelo em duas partes. Nesta mesma figura é possível observar o sinal senoidal de entrada (um único bloco à esquerda) e os blocos de captura de dados (conjunto de blocos à direita).

É realizado também um estudo comparativo entre filtros de *delay* fracionário para compensação do *clock skew*. Estes filtros foram avaliados utilizando aritmética de ponto flutuante de dupla precisão e aritmética de ponto fixo, para a implementação em FPGA.



**Figura 21: Visão geral da modelagem do TIADC**

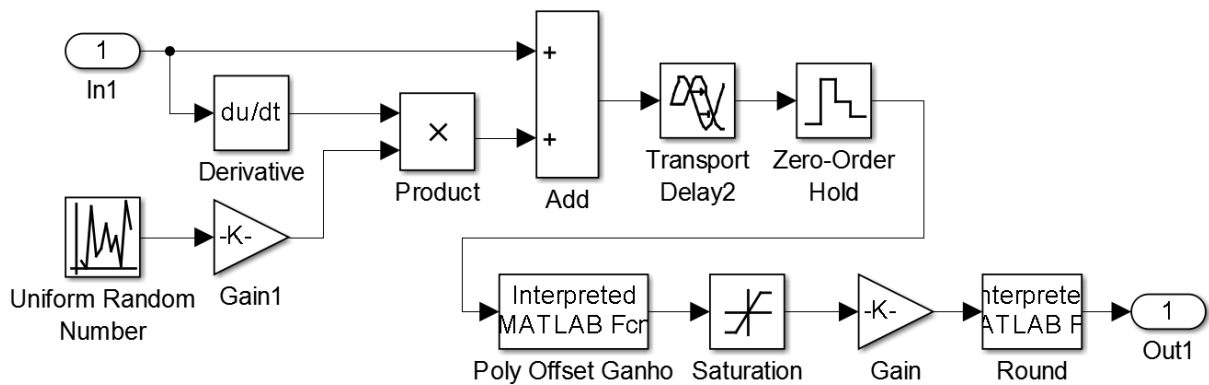
**Fonte: Autoria própria**

#### 4.1 MODELAGEM DO TIADC

Na parte *analógica* do TIADC, ilustrada na figura 21, existem dois blocos denominados *ADC channel 1* e *ADC channel 2*, que são os modelos dos ADCs de canal 1 e 2, respectivamente. Na saída destes blocos há ainda um conversor de tipo de dados, para converter o tipo de dado do sinal de entrada do modelo no tipo de dado adequado para a resolução do ADC. Ao longo de todo este trabalho a resolução do ADC considerada será de 8 bits. Cada bloco de ADC internamente descreve matematicamente as principais características de um ADC real, assim como suas fontes de erros. Na figura 22 é apresentado o diagrama de blocos de um dos ADCs; lembrando que a modelagem para 2 ADCs é a mesma.

Observando-se o diagrama da esquerda para a direita, a entrada de sinal ocorre no bloco *In1*. A primeira operação que ocorre no sinal é a adição do ruído de *jitter*. O *jitter* é simulado através de um gerador pseudo-randômico de números, ponderado pelo ganho *Gain1* e pela derivada do sinal de entrada. No bloco *Add* o ruído de *jitter* é adicionado ao sinal de entrada. Na sequência o bloco *Transport Delay* gera um atraso controlado no sinal de entrada. Este atraso controlado permite a defasagem de 180 graus do sinal de entrada necessária para um dos ADCs (para que a aquisição ocorra de forma intercalada) e permite a adição de desvios no instante de aquisição, simulando os efeitos do *clock skew*. A operação de *sample-and-hold* do ADC é simulada através do bloco *Zero-Order Hold*, que opera na frequência configurável  $F_{ADC}$ .





**Figura 22: Modelo interno do ADC**

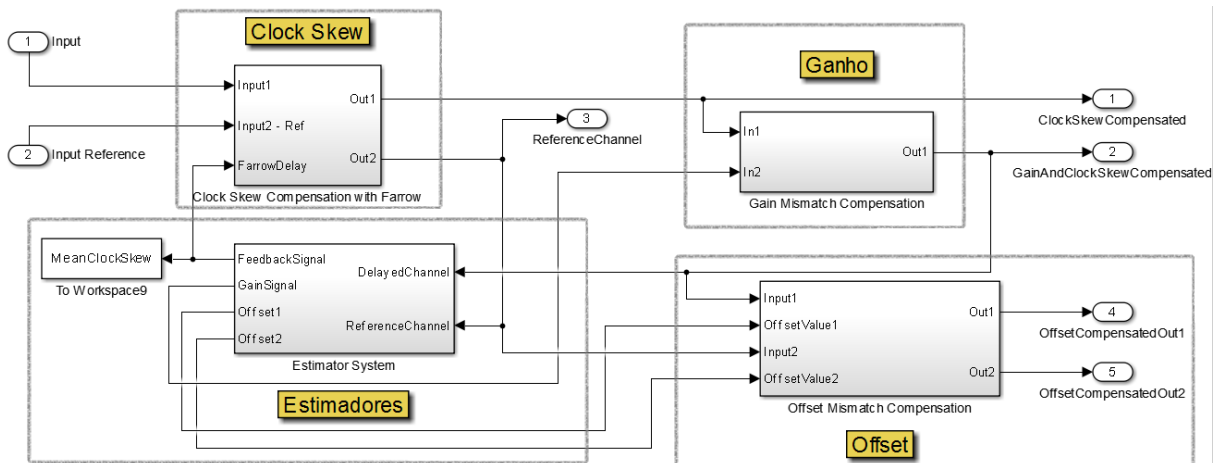
**Fonte: Autoria própria**

Após a discretização no tempo do sinal de entrada *contínuo* no tempo, os desvios de *offset* e *ganho* são emulados no bloco *Poly Offset Ganho* através de uma função polinomial. Na sequência, a saturação do ADC é emulada através do bloco *Saturation*, e o nível de *tensão* simulado na entrada é codificado e quantizado pelos blocos *Gain* e *Round*. Estes dois últimos blocos multiplicam a saída do sinal amostrado por uma constante, alinhando o valor máximo do fundo de escala da codificação com a tensão máxima de entrada, e realizam o arredondamento deste valor, quantizando a saída do ADC. O tipo de dado da saída deste bloco é o ponto flutuante de dupla precisão. O mesmo é posteriormente convertido pelo conversor de tipo de dado, ilustrado na figura 21. O modelo permite a configuração dos seguintes parâmetros do ADC: resolução, intensidade de ruído do *jitter*, *offset*, ganho, atraso de propagação no ADC (o que permite emular o *clock skew*) e tensão de fundo de escala. Estes parâmetros são configurados através de *script* em Matlab (MATLAB, 2014).

## 4.2 MODELAGEM DAS ESTRUTURAS DE COMPENSAÇÃO

As estruturas de compensação de *offset*, ganho e *clock skew* são modeladas na parte *digital* do TIADC. A compensação destes efeitos é subdividida em duas etapas: *estimação* dos desvios e *reconstrução* dos sinais. As estruturas de compensação modeladas operam em *background*, não requerendo portanto a injeção de um sinal conhecido na entrada. Entretanto, nesta forma de operação é necessário o uso de *estimadores* de desvios do TIADC. A modelagem dos compensadores e estimadores realizada é apresentada na figura 23.

A compensação de cada um dos efeitos é realizada individualmente, na sequência: *clock skew*, ganho e *offset*. Os estimadores de cada um dos desvios são apresentados por um



**Figura 23: Estrutura geral dos compensadores de desvio.**

**Fonte: Autoria própria**

único bloco, que internamente é subdividido com base nestes três parâmetros de desvio. A sequência foi selecionada com o objetivo de minimizar a utilização de elementos lógicos de uma implementação em FPGA. Isto porque o *word length* (o tamanho da palavra binária utilizada para a representação numérica dos sinais) da saída de cada compensador é maior do que o de entrada. Por esta razão optou-se por primeiro compensar o *clock skew*, que utiliza filtros de *delay* fracionário com vários multiplicadores e somadores, para na sequência compensar o ganho, que requer essencialmente um multiplicador, e o *offset*, que requer dois somadores somente.

Da figura 23 pode-se verificar ainda que o sinal de saída dos compensadores de ganho e *clock skew* são utilizados para realimentação através dos estimadores, enquanto no caso do *offset* a realimentação não é necessária. Isto porque os estimadores de ganho e *clock skew* realizam a estimativa de erro destes efeitos, onde o objetivo é minimizá-los. Diferentemente destes estimadores, o estimador de *offset* obtém os valores de *offset* de cada canal de forma independente. O detalhamento da modelagem de cada compensador e de seu respectivo estimador é dado nas subseções seguintes.

#### 4.2.1 COMPENSAÇÃO DO CLOCK SKEW

A compensação do *clock skew* é realizada utilizando a estrutura de Farrow, como proposto em (JOHANSSON; LOWENBORG, 2002). Entretanto, durante o projeto da estrutura de Farrow, as seguintes questões precisam ser avaliadas:

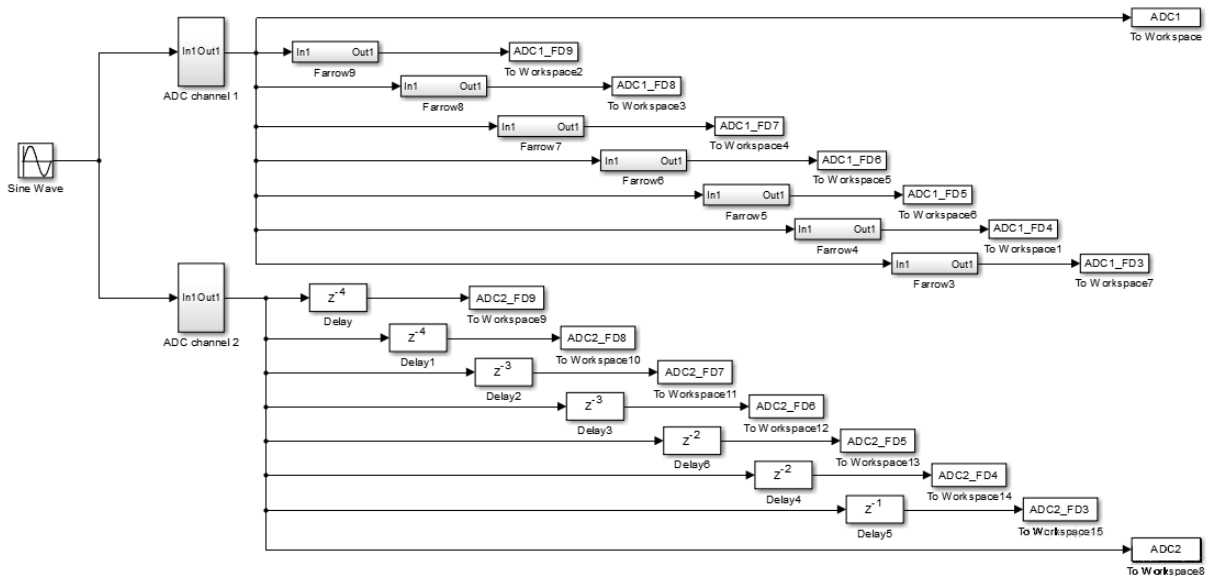
- Qual a ordem mínima dos filtros para atender uma especificação de banda mínima?

- Qual método de projeto da estrutura de Farrow, e de projeto de filtro, terá melhor relação custo/benefício?
- Qual o *word length* dos coeficientes em uma implementação utilizando aritmética de ponto fixo?

Para responder estas questões foi realizado um estudo dos diferentes projetos de filtros discutidos no capítulo 3. Para este estudo foi considerado que o TIADC é constituído de dois ADCs com resolução de 8 bits, cada ADC opera à frequência de 1 GHz, com *clock skew* máximo de 10% do intervalo de aquisição do ADC. A definição do valor máximo de *clock skew* segue o valor utilizado para testes de estimador de desvio apresentado em (ELBORNSSON et al., 2005). Este valor é coerente com os valores de desvio máximo encontrados na literatura, como por exemplo em (JAMAL et al., 2002), (RAZAVI, 2013) e (ZHAO et al., 2013). A compensação do *clock skew* deve permitir frequência máxima do sinal de entrada do TIADC de no mínimo 200MHz (banda mínima do TIADC). Após a reconstrução dos sinais do TIADC pelos filtros de compensação, o SNR e o SFDR são utilizados para avaliar o desempenho dos diferentes filtros.

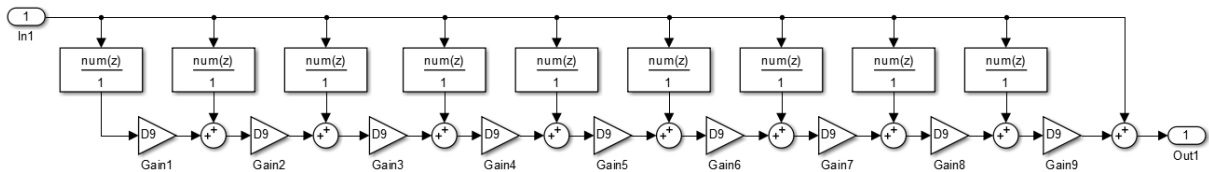
Para avaliar a menor ordem necessária dos filtros é utilizado um modelo simplificado do TIADC, contendo somente a parte analógica e os filtros de compensação do *clock skew*. Nesta avaliação foi utilizada a estrutura de Farrow Lagrange, devido principalmente a magnitude de sua resposta em frequência ser plana até os limites de banda do filtro. Esta escolha simplifica a análise do desempenho do filtro, assim como a escolha adequada da mínima ordem necessária do filtro. Outros métodos para obtenção da estrutura de Farrow podem apresentar oscilações na magnitude de sua respectiva resposta em frequência. Estas oscilações, que podem ocorrer ao longo de todo o espectro, tornariam complexa esta avaliação inicial. Para esta primeira avaliação foram projetados 7 estruturas de Farrow Lagrange, da terceira a nona ordem. O modelo completo é ilustrado na figura 24. O *clock skew* é emulado através de um desvio no instante de aquisição do ADC2. Este desvio consiste na adição de um atraso no instante de aquisição em relação ao instante ideal. Este atraso, que é menor do que o intervalo de aquisição do ADC, deverá ser compensado pela estrutura de Farrow na saída do ADC1. O sinal do ADC2 é considerado a *referência* para compensação do *clock skew*, e o ADC1 é o sinal que precisa ser compensado. Deve-se observar que os filtros de *delay* fracionário adicionam um *delay* inteiro ao sinal filtrado, sendo o seu valor dependente da ordem do filtro. A parte inteira do *delay* adicionado pelo filtro do ADC1 precisa ser então compensada no ADC2 através de elementos de *delay* inteiro, como é ilustrado na figura 24. O gráfico do SFDR é apresentado na seção 5.1.

Como exemplo do modelo da estrutura de Farrow Lagrange, é apresentado na figura 25 o modelo da estrutura de ordem 9. O multiplicador de *delay* é inserido através de uma variável



**Figura 24:** Modelo utilizado para teste da ordem do filtro utilizando a estrutura de Farrow Lagrange. Os blocos do ADC superior da figura são as estruturas de Farrow e os blocos do ADC inferior são os delays inteiros.

Fonte: Autoria própria



**Figura 25:** Modelo de teste da estrutura de Farrow Lagrange

Fonte: Autoria própria

em *script* do Matlab. As demais estruturas são similares a esta, com a diferença de possuírem uma quantidade menor de filtros FIR e multiplicadores de *delay*.

Uma das características da estrutura de Farrow Lagrange é a sua resposta em frequência praticamente idêntica ao filtro de Lagrange. Porém, neste método, ao definir a ordem dos filtros FIR da estrutura de Farrow, também é definida a ordem do polinômio em  $\mu$ . No capítulo 3 foram abordados dois métodos para o projeto da estrutura de Farrow: o de Farrow Lagrange e o  $L_2$  Laakso. O método  $L_2$  Laakso não possui a mesma limitação da estrutura de Farrow Lagrange, por esta razão foi avaliado o uso desta estrutura com filtros de Lagrange. A ordem do filtro é a mesma que foi selecionada no primeiro teste, mas foi testada a ordem do polinômio de 1 (assumindo a possibilidade de uma aproximação linear) à 5. Os gráficos do SNR e SFDR deste teste são apresentados na seção 5.1. Além da análise da diferença da ordem polinomial, também foi avaliado o *word length* necessário dos coeficientes para cada estrutura, assim como

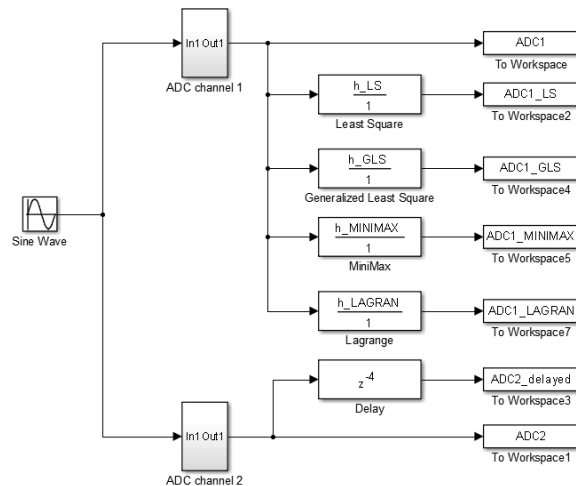
os recursos utilizados em uma implementação em FPGA. A estratégia utilizada para definição do *word length* constituiu em:

- configurar um *word length* comum à todos os coeficientes;
- simular o modelo do TIADC em diferentes frequências;
- avaliar o SNR, e principalmente o SFDR resultante;
- se a degradação do SFDR for menor do que 2 dB o *word length* é reduzido de 1 bit e o processo é repetido, caso contrário o último *word length* utilizado é considerado o valor mínimo necessário.

O SFDR foi utilizado como parâmetro por apresentar maior sensibilidade as mudanças na estrutura do filtro do que o SNR, que sofre alterações muito pequenas em amplitude quando comparado com o SFDR. Estas duas estruturas foram implementadas em FPGA, com o objetivo de avaliar o custo, em termos de unidades lógicas, de cada estrutura. A metodologia de implementação e testes é discutida na seção 4.3 deste capítulo. Deve-se observar que a estrutura  $L_2$  Laakso permite o uso de qualquer método de projeto de filtros de *delay* fracionário. Desta forma é interessante avaliar o desempenho de diferentes métodos de projeto. Esta avaliação foi realizada para os métodos de projeto apresentados no capítulo 3 com o uso do modelo da figura 26. Dos quatro métodos de projeto apresentados, dois permitem o controle da banda de interesse do filtro. Na avaliação realizada a banda de interesse configurada foi de  $0,5\pi$ , o equivalente à 250MHz.

Os resultados obtidos destes testes serão discutidos na seção 5.1. Um dos métodos de projeto de filtros foi selecionado nestes testes. Todas as demais etapas deste estudo utilizam exclusivamente este método de projeto de filtros de *delay* fracionário.

A próxima etapa para implementação deste filtro foi obter o menor *word length* possível. Nesta etapa também foi avaliada a limitação da faixa do *delay* da estrutura de Farrow, com o objetivo de reduzir a ordem do polinômio da estrutura de Farrow em  $\mu$ . Em todos os testes realizados anteriormente a faixa do *delay* era limitada à  $1/2$  do intervalo de aquisição. A metodologia para determinação do *word length* foi diferente neste caso. Com base no *word length* obtido para o filtro de Lagrange é possível determinar o valor máximo de erro da magnitude do filtro com relação ao filtro ideal, dentro da banda de interesse, para diferentes delays. Conhecido o erro máximo da aproximação, a determinação do *word length* é realizada iterativamente através de um *script* em Matlab. Duas estratégias, complementares, foram utilizadas para determinação do *word length*. A primeira estratégia utiliza a ideia de



**Figura 26: Modelo utilizado para comparação do desempenho dos filtros *LS*, *GLS*, *Minimax* e *Lagrange*.**

**Fonte: Autoria própria**

*word length* igual para todos os coeficientes, como ocorreu na determinação do *word length* da estrutura anterior. Nesta estratégia é utilizado um *script* com a seguinte sequência de passos:

- um valor inicial de *word length* é dado para os coeficientes da estrutura;
- é gerada a resposta em frequência desta estrutura e é obtido o maior erro de aproximação da magnitude do filtro com relação ao ideal, dentro da banda de interesse;
- se o erro de aproximação for menor do que o especificado, 1 bit do *word length* é reduzido, e a etapa anterior repetida;
- caso contrário o valor de *word length* anterior é tratado como valor limite do filtro.

A segunda estratégia é um complemento da primeira. Esta tem por princípio reduzir o *word length* dos coeficientes dos filtros FIR da estrutura de Farrow de maior ordem polinomial em  $\mu$ . A justificativa para esta estratégia consiste na seguinte avaliação sobre a estrutura de Farrow. Considere a equação (56), em que os coeficientes  $C_m(n)$  do filtro  $h_m(n)$  são dados como polinômios em  $\mu$ . A quantização dos coeficientes do filtro FIR de  $C_m(n)$  pode ser interpretada como a adição de um erro  $E_m(n)$  a estes coeficientes. Sendo assim, a equação (56) pode ser reescrita como:

$$h_m(n) = \sum_{m=0}^{M_O} \{C_m(n) + E_m(n)\} \mu^m, n = 0, 1, 2, \dots, N. \quad (66)$$

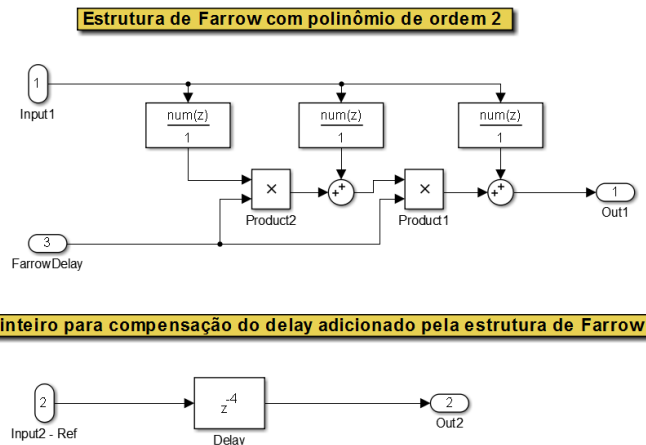
Retirando do somatório os primeiros termos da equação (66) pode-se escrever:

$$h_m(n) = [C_0(n) + E_0(n)] + [C_1(n) + E_1(n)]\mu + [C_2(n) + E_2(n)]\mu^2 + \sum_{m=3}^{M_O} \{C_m(n) + E_m(n)\}\mu^m, \quad n = 0, 1, 2, \dots, N. \quad (67)$$

Lembrando que os valores para o *delay* fracionário  $\mu$  são menores do que 1, torna-se intuitivo da equação (67) que o coeficiente de menor ordem polinomial,  $C_0(n)$ , é o que possui maior peso na saída da estrutura de Farrow. Consequentemente, um erro  $E_0(n)$  maior neste coeficiente levará a um maior erro na saída da estrutura. Os coeficientes  $C_1(n)$  e  $C_2(n)$  são ponderados por  $\mu$  e  $\mu^2$ , respectivamente, possuindo um peso menor na saída da estrutura. Desta observação pode-se verificar que, os erros  $E_1(n)$  e  $E_2(n)$  também possuirão menor peso no erro total. E o peso dos coeficientes e do erro será menor quanto maior for o valor do expoente de  $\mu$ . Assim, a segunda estratégia consiste em:

- Do valor de *word length* obtido com a primeira estratégia é subtraído um bit dos coeficientes do expoente de maior valor;
- é gerada a resposta em frequência da estrutura de Farrow e é obtido o maior erro de aproximação da magnitude com relação ao filtro ideal;
- se o erro de aproximação for menor do que o especificado, 1 bit do *word length* é reduzido dos coeficientes com o valor do expoente atual menos 1, e o passo anterior é repetido;
- caso o erro de aproximação venha a ser maior do que o especificado, ou o expoente venha a ser zero, trata a configuração de *word length* anterior como menor valor possível.

Os resultados obtidos com o uso destas duas estratégias são discutidos na seção 5.1. Como mencionado anteriormente nesta subseção, além do *word length*, também foi verificada a alteração da faixa do *delay* da estrutura de Farrow, que nos primeiros testes era de 1/2. Utilizando o método  $L_2$  Laakso foram verificados os efeitos de limitar a faixa do *delay* ao valor 0, 1, considerado o valor máximo de *clock skew* do TIADC. Comparações entre implementações utilizando diferentes estratégias para definição do *word length*, diferentes limitações de faixas do *delay* fracionário para a estrutura de Farrow e diferentes ordens de polinômios em  $\mu$  são apresentadas na seção 5.1. O modelo do compensador de *clock skew* com a estrutura de Farrow, agora com a entrada do valor de *delay* fracionário  $\mu$ , é apresentado na figura 27. Na parte superior da figura é apresentado o modelo para a estrutura de Farrow com polinômio de segunda



**Figura 27: Modelagem do compensador de *clock skew* com estrutura de Farrow.**

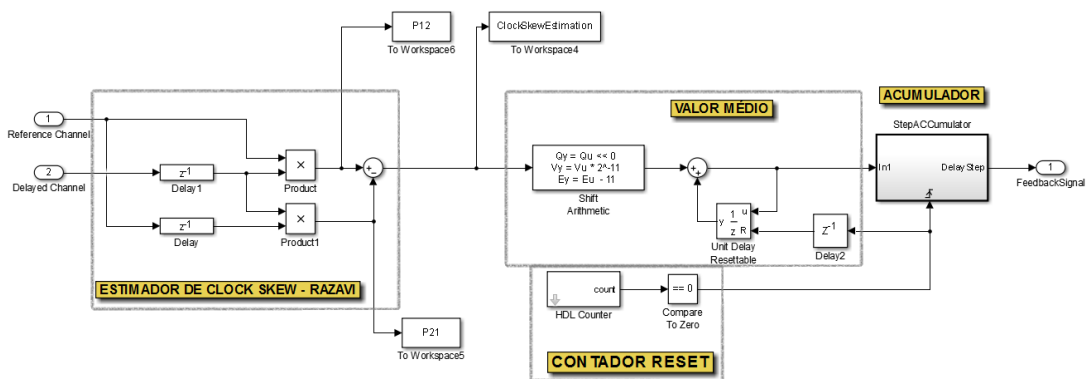
**Fonte: Autoria própria**

ordem, na parte inferior o *delay* inteiro para compensação do *delay* do filtro é modelado. A estrutura apresenta três filtros FIR (os blocos retangulares maiores), dois multiplicadores e dois somadores.

Além da reconstrução do sinal para compensação do *clock skew*, também é necessária a modelagem de um estimador de desvio. O estimador implementado, com algumas adaptações, é o proposto em (RAZAVI, 2013). Razavi propôs um estimador que gera um sinal de erro a partir do valor médio da diferença da autocorrelação entre os dois canais do TIADC. Este estimador foi estudado utilizando sinais de entrada senoidais e sinais de entrada randômicos. A estrutura principal do estimador requer dois multiplicadores e um somador, sendo que depois é requerido o cálculo de valor médio do erro e a multiplicação deste valor por um fator de escalonamento. O valor obtido do fator de escalonamento é diretamente utilizado pela estrutura de Farrow para compensação do *clock skew*. A modelagem completa do estimador é apresentada na figura 28.

Na figura 28 pode-se identificar o conjunto de blocos à esquerda como os dois multiplicadores e o somador da estrutura sugerida por Razavi. A direita é identificado um conjunto de blocos marcado na região do *VALOR MÉDIO*, que é um acumulador que pode ser reinicializado e que contém um divisor na entrada, implementado através de um deslocamento à direita dos bits do sinal de entrada deste bloco. Os blocos marcados na região *CONTADOR RESET* realizam a contagem do número de amostras que devem ser acumuladas, e geram um sinal para o *reset* do acumulador de *VALOR MÉDIO* e para o *trigger* do bloco *ACUMULADOR* ao final de cada contagem. O valor médio de erro calculado em *VALOR MÉDIO* é acumulado e escalonado no bloco acumulador, sendo a saída deste bloco diretamente conectada à estrutura de Farrow. Ou seja, o estimador retorna o valor do *delay* fracionário que deve ser adicionado ao





**Figura 28: Modelo do estimador de *clock skew* proposto por Razavi.**

**Fonte: Autoria própria**

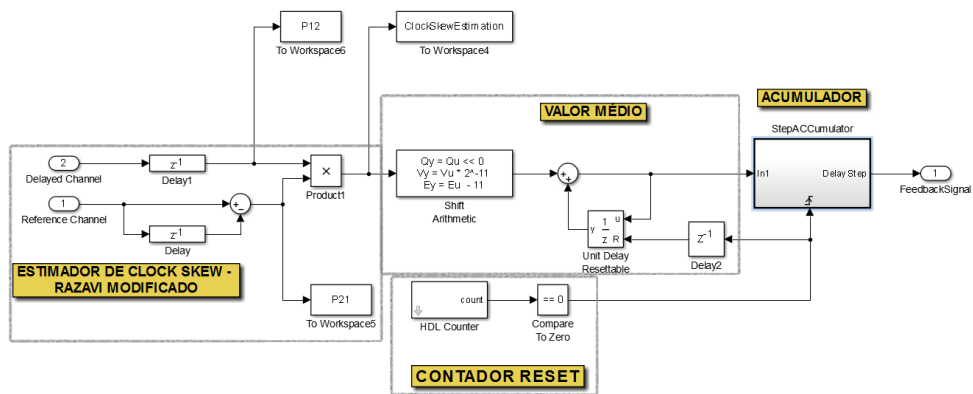
signal do ADC1 para a compensação do *clock skew*.

É possível simplificar o estimador proposto por Razavi se for observada a equação representada pelos multiplicadores e somadores do conjunto de blocos do estimador. A mesma é dada por:

$$e(n) = ADC1(n-1) \cdot ADC2(n) + ADC1(n-1) \cdot ADC2(n-1), \quad n = 0, 1, 2, \dots, \infty \quad (68)$$

onde  $e(n)$  é a saída do estimador,  $ADC1(n)$  é o sinal do ADC que precisa ser compensado e  $ADC2(n)$  o sinal do ADC de referência. A equação (68) pode ser simplificada para a equação (69), permitindo a implementação do mesmo estimador com a redução de um multiplicador da estrutura. A figura 29 apresenta o modelo da estrutura modificada.

$$e(n) = ADC1(n-1) \cdot [ADC2(n) - ADC2(n-1)]. \quad (69)$$



**Figura 29: Modelo do estimador de *clock skew* proposto por Razavi com a simplificação do estimador.**

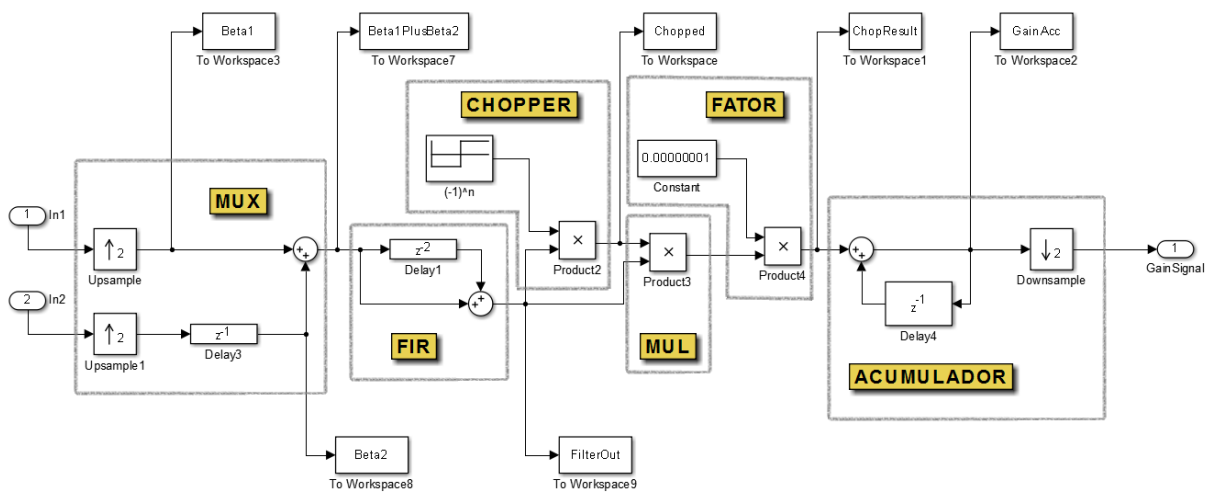
**Fonte: Autoria própria**

#### 4.2.2 COMPENSAÇÃO DO GANHO

O estimador implementado para o ganho é o método proposto em (JAMAL et al., 2002). Jamal propôs um método de estimação em que, o sinal de saída do TIADC é multiplicado por um sinal periódico alternado de sequência  $\{1; -1\}$  e frequência  $F_{TIADC}/2$ . Na sequência este sinal gerado é multiplicado pelo sinal de saída do TIADC, o que resulta em um sinal DC (contínuo) proporcional ao desvio de ganho. Com o uso de um acumulador todas as demais componentes deste sinal são removidas, exceto a componente DC. A saída do acumulador é multiplicada por um fator e depois utilizada no multiplicador de reconstrução do sinal de saída do TIADC. O multiplicador de reconstrução é utilizado em somente um canal do TIADC, sendo o outro canal utilizado como *referência*. O estimador opera então continuamente dentro de uma malha de realimentação, fazendo o sistema aproximar a saída de erro de ganho do valor zero. A figura 30 apresenta o modelo do estimador de ganho.

Na figura 30 são marcados vários blocos contendo uma indicação de funcionalidade no sistema. As entradas de sinais *In1* e *In2* são as saídas dos dois ADCs do TIADC, sendo que o sinal de *In1* é alterado pelo multiplicador de reconstrução de ganho e *In2* é a referência. Os blocos marcados na região do *MUX*, na figura 30, realizam a multiplexação dos sinais dos dois ADCs, retornando um único sinal com o dobro da taxa de aquisição de cada ADC individual. Na região marcada por *CHOPPER* é realizada a multiplicação pelo sinal periódico alternado, e na região *MUL* é realizada a multiplicação do sinal do TIADC pelo sinal de saída do *CHOPPER*.

Pode-se observar que o sinal do TIADC é processado primeiro pelos blocos da região *FIR*, que são um filtro FIR dado por  $1 + z^{-2}$ . Este filtro produz uma saída nula para sinais na frequência  $F_{TIADC}/4$ . No estimador proposto por (JAMAL et al., 2002) sinais na frequência



**Figura 30: Modelo do estimador de ganho**

**Fonte: Autoria própria**

$F_{TIADC}/4$  geram um nível DC na saída mesmo quando não há desvio de ganho no TIADC. Esta limitação do método proposto por Jamal exige a filtragem dos sinais nesta frequência. Na sequência, o sinal de saída de *MUL* é multiplicado por um fator de escala pelos blocos da região *FATOR*, para então ser acumulado na região marcada pelo *ACUMULADOR*. Deve-se observar que nesta região há um bloco denominado *downsample*, que divide por dois a frequência do sinal do acumulador para ser utilizado no multiplicador de reconstrução do ganho.

O multiplicador de reconstrução de ganho está no bloco interno de *GANHO*, na figura 23. Como pode ser observado da figura 23, o sinal de entrada deste bloco é a saída do bloco de compensação de *clock skew*. Isto requer que a *word length* deste bloco, tanto a parte inteira quanto a fracionária, atenda o *word length* mínimo requerido pelo bloco anterior. Caso contrário o sinal de saída da compensação do *clock skew* será distorcido pelo compensador de ganho. A *word length* mínima necessária para o ganho, considerando os compensadores implementados, é de 16 bits, 8 bits para a parte inteira e 8 bits para a parte fracionária. Gráficos do SNR e do SFDR do TIADC antes e depois da compensação de ganho são apresentados na seção 5.2. A utilização de unidades lógicas da FPGA pelo sistema de compensação de ganho é apresentada na seção 5.4.

#### 4.2.3 COMPENSAÇÃO DO OFFSET

Para a compensação do *offset* é implementado um dos métodos mais simples de estimativa, o uso do valor médio do sinal de entrada de cada ADC. Este método possui algumas limitações, sendo por isto sugerida na literatura a utilização de *choppers* randômicos de sinal

na entrada do TIADC (EKLUND; GUSTAFSSON, 2000). Em (VOGEL, 2005) é realizada uma discussão sobre a utilização deste método. Um dos aspectos que requer atenção neste método de estimativa é a quantidade de amostras que devem ser consideradas para o cálculo da média. O valor de 4096 amostras utilizado neste trabalho foi determinado empiricamente, através de simulações. A figura 31 apresenta o diagrama em blocos do estimador de *offset*. Para o cálculo do valor médio de cada canal foi utilizado um acumulador por canal, como é destacado na figura 31. Cada acumulador é constituído de um bloco somador e de um elemento de memória que pode ser *resetado*. Na figura 31 é destacado ainda um *contador de reset*. Este contador é um *down counter* que recarrega o valor inicial de contagem automaticamente ao fim de cada contagem (quando chega em zero). Pode-se observar que na saída deste contador há um comparador em zero, que retorna um sinal para o *reset* dos acumuladores e para o subsistema *Triggered Subsystem*.

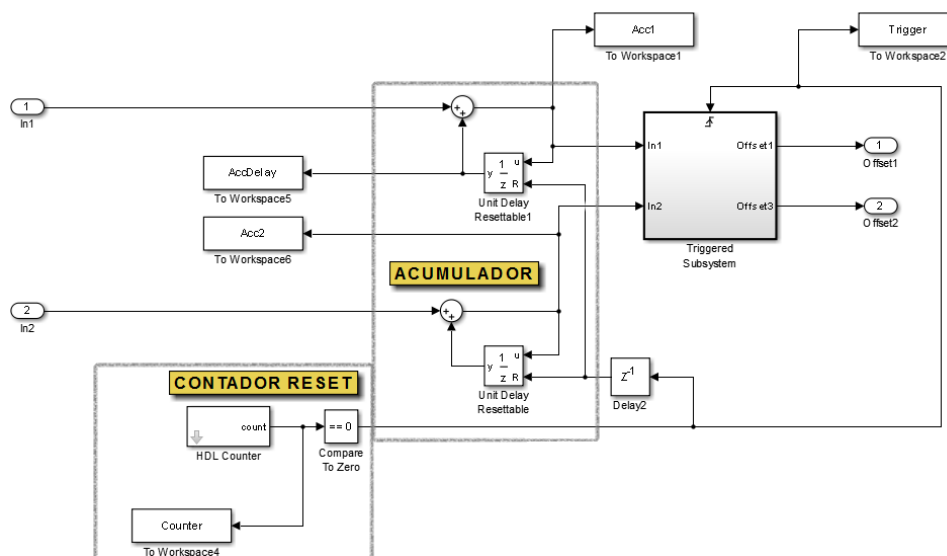


Figura 31: Modelo do estimador de *offset*.

Fonte: Autoria própria

O subsistema *Triggered Subsystem* somente está ativo após a ocorrência do sinal de *reset*, mantendo os valores de saída do bloco inalterados após a ocorrência deste sinal. Internamente este subsistema realiza a divisão dos valores presentes nos acumuladores. A divisão é efetuada através de um deslocamento aritmético de 12 bits para a direita, o que equivale a divisão por 4096. A vantagem do deslocamento aritmético é que esta operação utiliza uma quantidade menor de unidades lógicas, para uma implementação em FPGA, do que a operação completa de divisão.

O valor obtido da estimativa de *offset* é subtraído de cada canal para compensação

através de dois somadores, presentes no bloco de *offset* marcado na figura 23. Gráficos do SNR e do SFDR do TIADC antes e depois da compensação de *offset* são apresentados na seção 5.3. A utilização de unidades lógicas da FPGA pelo sistema de compensação de *offset* é apresentada na seção 5.4.

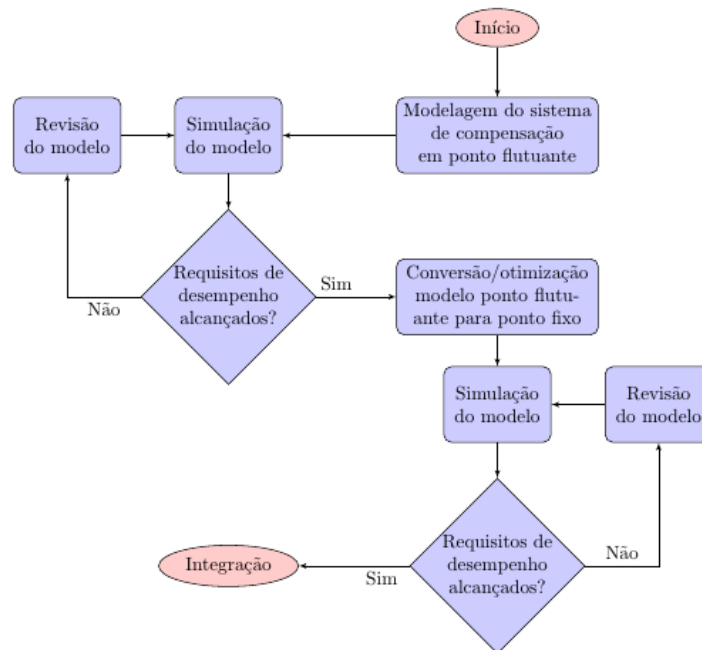
### 4.3 METODOLOGIA DE DESENVOLVIMENTO E TESTE

#### 4.3.1 METODOLOGIA DE DESENVOLVIMENTO

Nas seções anteriores foram tratadas a modelagem das partes analógica e digital do TIADC, sendo esta última parte constituída dos compensadores e estimadores de desvio. Após construir os modelos comportamentais da compensação dos desvios, estes foram simulados em *software* para verificação de funcionalidade e avaliação de desempenho. Esta etapa de maior abstração precede todas as demais neste projeto de implementação em FPGA dos compensadores de desvio. O fluxo do desenvolvimento deste projeto seguiu uma metodologia *top-down* (MARIANO, 2008). Seguindo esta metodologia, foi criado, simulado e avaliado um modelo abstrato da estrutura e comportamento dos compensadores. Concluído este processo, passou-se para a transcrição deste modelo abstrato em linguagem de descrição de *hardware*. E, finalmente, foi realizada a implementação deste sistema para validação do projeto.

Do apresentado nas seções anteriores pode-se observar a clara divisão do sistema de compensação em três subsistemas de compensação, sendo eles: *offset*, ganho e *clock skew*. A modelagem de cada subsistema é realizada segundo o fluxograma apresentado na figura 32. A primeira etapa no projeto de cada subsistema consiste na sua modelagem, que foi realizada com o *software* Simulink. Este primeiro modelo é construído utilizando aritmética de ponto flutuante de dupla precisão. Após construído, o modelo é simulado e os dados obtidos são analisados. Como pode ser observado do fluxograma, após a etapa de simulação e análise, uma decisão sobre o projeto deve ser tomada. Se o desempenho requerido do compensador for obtido, segue-se para a próxima etapa do projeto, como apontado no fluxograma. Caso contrário, o modelo é revisto e o processo de simulação e análise é repetido.

A etapa seguinte consiste na conversão e otimização da estrutura modelada para operação utilizando aritmética de ponto fixo. Deve-se lembrar que o objetivo final do projeto é a implementação dos compensadores em FPGA. A implementação em *hardware* de operações aritméticas em ponto flutuante possui um elevado custo, em termos de recursos da FPGA (ANDERSON et al., 2008). Por esta razão, sistemas de processamento digital de sinais, quando implementados em FPGA, utilizam preferencialmente aritmética de ponto fixo. A representação



**Figura 32: Fluxograma do processo de desenvolvimento dos subsistemas de compensação.**

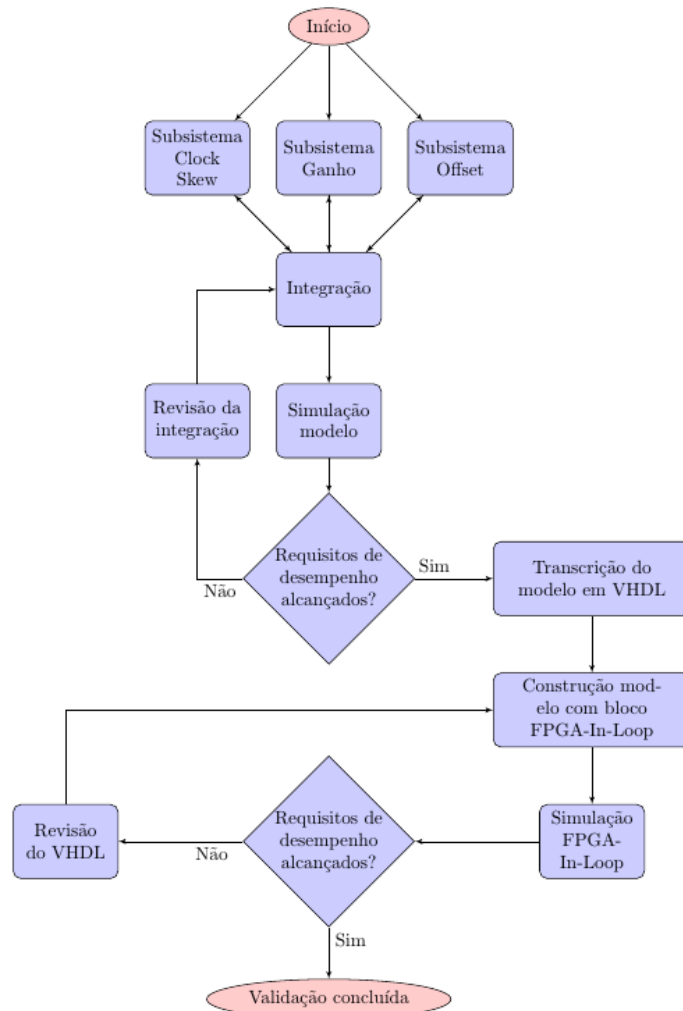
**Fonte: Autoria própria**

numérica em ponto fixo entretanto introduz não linearidades ao sistema, modificando o seu comportamento. Após a conversão do modelo de ponto flutuante para ponto fixo, o mesmo é simulado para verificar a degradação no desempenho do sistema. Desta forma, é possível tratar os efeitos da utilização de aritmética de ponto fixo durante a construção do modelo abstrato do sistema. Oportunidades para otimizar o sistema, com base no *word length* utilizado nas estruturas, também podem ser detectadas nesta fase do projeto.

Seguindo o fluxograma da figura 32 é observado que, após a conversão do sistema, mais uma simulação e análise são realizadas. Após estas duas etapas, é novamente necessária a tomada de uma decisão. Se o novo modelo em ponto fixo do sistema obter o desempenho desejado, o subsistema estará pronto para a próxima etapa: a sua integração ao sistema. Caso contrário, o subsistema deve ser revisto e o processo repetido.

A etapa de integração envolve a construção de um modelo em que todos os subsistemas coexistam, com o mínimo de interferência em suas funcionalidades. O fluxo de desenvolvimento do sistema é apresentado na figura 33. Este fluxograma descreve o desenvolvimento do sistema completo, sendo a primeira etapa o desenvolvimento dos subsistemas. Seguindo o fluxograma da figura 33, observa-se que após integrado o sistema é simulado, e novamente o seu desempenho avaliado. Se o resultado obtido for negativo, a integração é revista. Eventualmente, durante a integração é detectada a necessidade de rever

um subsistema. Esta revisão é indicada por flexas de retorno da etapa de integração para os subsistemas. Se a integração for bem sucedida, a etapa seguinte consiste na transcrição do modelo comportamental para uma linguagem de descrição de *hardware*.



**Figura 33: Fluxograma do processo de desenvolvimento do sistema completo de compensação.**

**Fonte: Autoria própria**

A linguagem escolhida para a transcrição é a VHDL (*Very high speed integrated circuit Hardware Description Language*). Esta etapa foi automatizada através da ferramenta HDLCoder (MATLAB; HDLCODER, 2014), do *software* Matlab. A ferramenta permite a conversão automática de blocos do Simulink em VHDL, inclusive de subsistemas inteiros. O código VHDL gerado segue o padrão de interface da indústria, utilizando o tipo de dado *standard logic* (PEDRONI, 2010) da linguagem VHDL. Além das entradas e saídas do modelo do subsistema convertido, também são adicionados os sinais de *clock*, *reset* e *enable* no código gerado. Estas entradas são necessárias para o controle e operação do sistema digital descrito.

Na sequência do fluxograma da figura 33, é apresentada a etapa de construção de modelo para simulação *FPGA In Loop* (FIL). *FPGA In Loop* consiste em uma metodologia de verificação e validação de projetos de sistemas digitais de controle e processamento de sinais. Maior detalhamento sobre a construção do modelo FIL e a próxima etapa de simulação FIL será dado na subseção seguinte. Com o modelo de interface entre o computador e a FPGA é realizado o último teste, em que o sistema digital é executado na FPGA e a parte analógica do modelo é emulada em *software*.

Da análise deste último teste é tomada uma decisão no fluxograma da figura 33. Se o sistema implementado em FPGA alcançou o desempenho requerido da aplicação, é considerado que o sistema digital implementado está *validado*. Caso contrário, será necessário rever o processo de transcrição do modelo abstrato do sistema em VHDL. Deve-se observar que esta última etapa valida o sistema implementado para atuação no modelo do sistema físico descrito. Caso existam erros ou imperfeições nos modelos comportamentais dos elementos externos à FPGA, pode ocorrer de o sistema digital implementado sofrer degradação de desempenho ou perda de funcionalidade. Entretanto, nesta situação, antes de ser revisto o modelo do sistema digital é preciso rever o modelo dos sistemas externos. Somente após esta revisão torna-se interessante realizar modificações nos sistemas implementados na FPGA e a repetição de todo este processo.

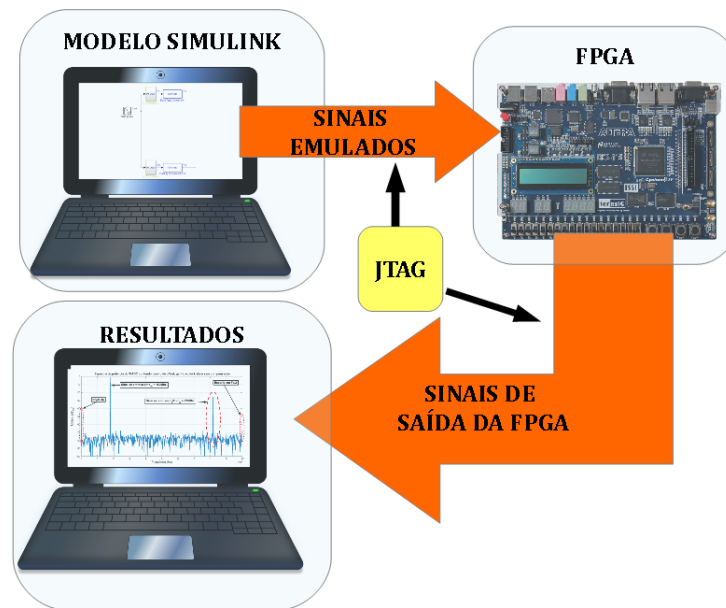
Os resultados obtidos da FIL dos sistemas de compensação, e o uso adicional de recursos da FPGA para a sua realização, são discutidos na seção 5.5.

#### 4.3.2 METODOLOGIA DE TESTE

Como explanado na subseção anterior, os testes dos compensadores em FPGA foram realizados utilizando a metodologia *FPGA In Loop*. Nesta metodologia o sistema digital desenvolvido para controle e processamento é implementado em *hardware*, mais especificamente em uma FPGA. Porém, todos os sistemas externos à FPGA, como circuitos eletrônicos, sensores e atuadores, são emulados por *software*. O conceito de FIL é utilizado por exemplo em (KARIMI et al., 2007) para o desenvolvimento de sistemas de controle de fonte chaveadas.

Para utilizar a FIL no Simulink foi utilizada a ferramenta HDLVerifier (MATLAB; HDLVERIFIER, 2014), do *software* Matlab. A ferramenta permite automatizar a implementação de vários processos para a realização da simulação com a FIL. Deve-se observar que a FIL exige o uso de uma interface de comunicação entre o sistema implementado em FPGA e o computador. Esta interface deve enviar os sinais emulados em *software* para as entradas da





**Figura 34: Ilustração do processo de teste FPGA In Loop.**

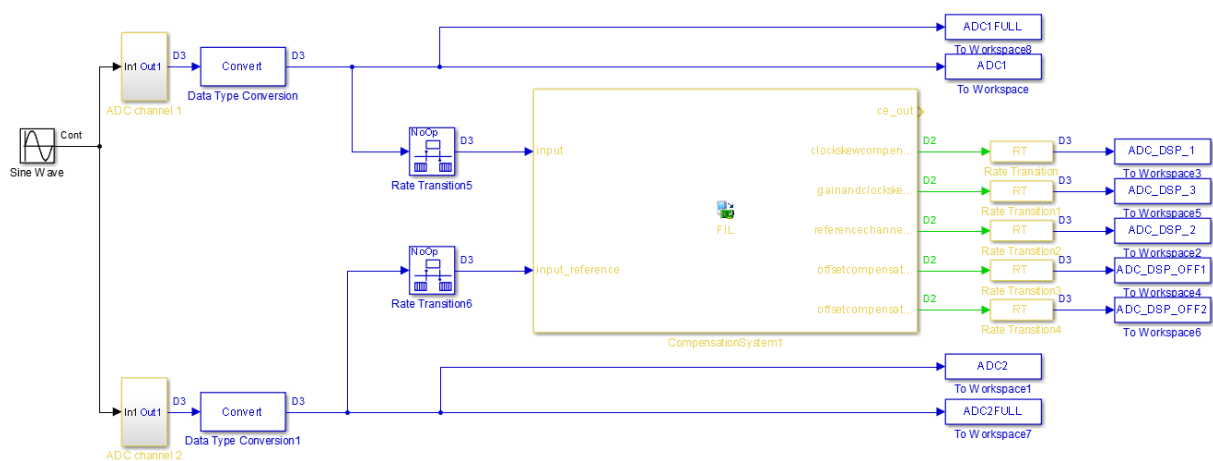
**Fonte: Autoria própria**

FPGA, e enviar os sinais de saída desta novamente para o computador. Este processo deve ser sincronizado com a emulação de sinais do *software*, sendo que o controle adequado deste sincronismo é realizado por esta interface. O processo é ilustrado pela figura 34, onde a parte *analógica* do TIADC é emulada em *software*, e a parte *digital* do sistema de compensação é realizada pela FPGA. Desta forma o sistema de compensação dos desvios do TIADC é validado no próprio *hardware*.

Antes de a FIL ser utilizada no Simulink, é necessária a construção de um modelo contendo a interface com o projeto em FPGA. Assim como é necessário adicionar ao projeto da FPGA a interface de comunicação com o computador. Estas duas etapas são automatizadas pela ferramenta HDLVerifier. A utilização da ferramenta consiste no processo que é discutido abaixo.

Primeiramente, é necessário definir uma FPGA, em conjunto com uma placa de desenvolvimento, ou uma placa customizada, e uma interface de comunicação entre esta e o computador. Para este trabalho foi utilizado o kit de desenvolvimento *DE2-115*, da empresa Terasic, que contém uma FPGA do fabricante Altera, da família *Cyclone IV E (EP4CE115F29C7)*. A interface de comunicação selecionada foi a interface JTAG (*Joint Test Action Group*, especificação IEEE Std. 1149.1-2001), disponível no kit para conexão com o computador pela porta USB (*Universal Serial Bus*).

Definida a placa e a interface, a próxima etapa consiste em especificar para a



**Figura 35: Diagrama de interface da simulação FPGA In Loop.**

**Fonte: Autoria própria**

ferramenta HDLVerifier quais os arquivos de código VHDL descrevem o sistema que será testado e validado. Qualquer projeto em VHDL pode ser utilizado em conjunto com a ferramenta, somente é requerida a identificação de sinais de *clock*, *reset* e *enable*, assim como a utilização do tipo de dado STD LOGIC. A identificação do arquivo principal de descrição do *hardware* também é requerida. Com a definição destes arquivos o HDLVerifier consegue identificar automaticamente as entradas e saídas do sistema. A ferramenta trata como responsabilidade do usuário a definição do tipo de dado que estas entradas e saídas representam para o Simulink, permitindo que o usuário realize a configuração adequada. Não havendo nenhum erro nestas etapas, o HDLVerifier constrói um projeto no *software* do fabricante da FPGA, neste caso no Quartus II (Quartus II Web Edition, 2013). Neste projeto são adicionados o código VHDL selecionado pelo usuário e um código VHDL de controle e interface entre a FPGA e o computador através da JTAG. Este código de controle e interface é automaticamente customizado pela ferramenta. Todo o processo de síntese do projeto e geração de arquivos de gravação é realizado pela ferramenta através de uma integração entre o Matlab e o Quartus.

Ao finalizar esta etapa com sucesso, o Simulink fornece ao usuário um novo bloco, exclusivo para a realização da FIL com o sistema sintetizado. Este bloco deve ser utilizado no modelo Simulink que irá emular os sistemas externos à FPGA, pois é este bloco que fornecerá todo o controle e interface de dados do Simulink com a FPGA. Na figura 35 é apresentado o bloco gerado pela ferramenta e o modelo construído para a utilização da FIL.

Concluída esta etapa, é gravado o sistema gerado de forma automática pelo Simulink na FPGA, e é realizada a simulação com FIL. Em resumo, todo o processamento do sistema digital é realizado efetivamente na FPGA, de forma similar ao que irá ocorrer na aplicação final

do sistema. A interface entre o projeto em FPGA e o Simulink permite a visualização dos sinais processados pela FPGA no computador, facilitando o armazenamento dos dados de testes e a sua posterior análise. Neste projeto por exemplo, foi possível realizar as mesmas avaliações de SNR, SINAD, SFDR e ENOB utilizadas nas simulações somente em *software* para o sistema implementado em FPGA.

## 5 DISCUSSÃO DOS RESULTADOS

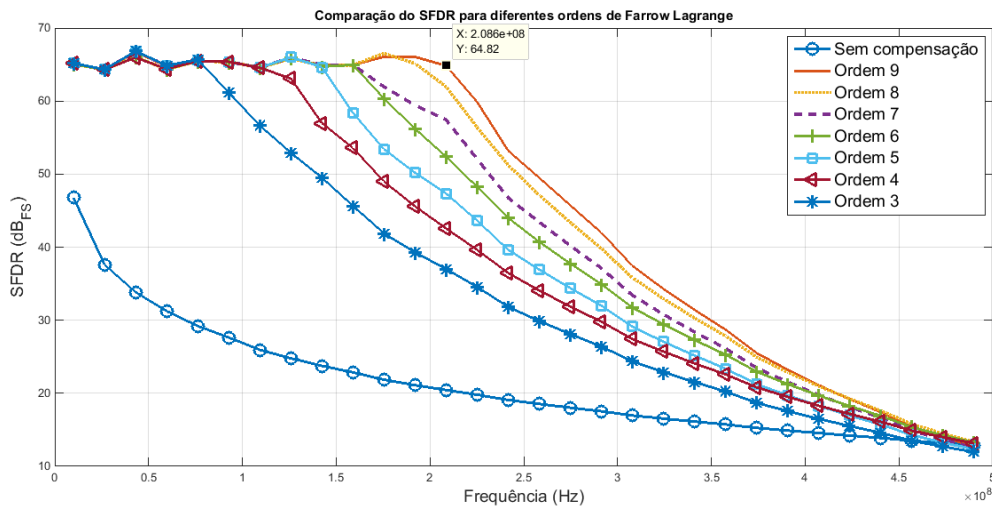
Serão apresentados e discutidos neste capítulo os resultados obtidos com os modelos e sistemas implementados no capítulo 4. Da mesma forma como foi subdividido no capítulo anterior, os desvios de offset, ganho e clock skew possuirão uma seção própria neste capítulo. Na seção 5.4 será realizada breve discussão e apresentação de resultados do sistema como um todo, contendo todos os compensadores. Questões relativas a implementação de *hardware*, como uso dos recursos da FPGA, serão apresentados em maiores detalhes nesta seção.

### 5.1 CLOCK SKEW

Nesta seção serão apresentados os resultados parciais obtidos das análises descritas na subseção 4.2.1. Os resultados obtidos com a estrutura de *clock skew* utilizada no sistema de compensação serão apresentados ao final desta seção. Os resultados apresentados nesta seção seguirão a sequência em que é descrito o estudo da estrutura de Farrow na subseção 4.2.1. A primeira questão diz respeito à determinação da ordem dos filtros, que foi analisada com o auxílio do gráfico da figura 36. A figura 36 apresenta o SFDR obtido para diferentes ordens da estrutura de Farrow Lagrange, considerando um *clock skew* de 10% do intervalo de aquisição do TIADC, valor máximo do desvio do *clock skew*, como discutido na subseção 4.2.1. Do gráfico obtido é evidente que a estrutura de ordem 9 é a mais indicada para atender o requisito de banda mínima de 200MHz.

Deste resultado pode-se comparar a estrutura de Farrow Lagrange com a estrutura  $L_2$  Laakso, como descrito na subseção 4.2.1. A comparação do SNR e do SFDR pode ser visualizada nas figuras 37 e 38, respectivamente. Pode-se observar destas figuras que os resultados obtidos com a estrutura  $L_2$  Laakso, da segunda a quinta ordem, são muito próximos aos obtidos com a estrutura de Farrow Lagrange. A estrutura  $L_2$  Laakso de primeira ordem (linear) mostra-se inadequada para o propósito da compensação do *clock skew*.

Um detalhe importante para ser observado é a diferença entre a segunda e a terceira ordem da estrutura  $L_2$  Laakso. Se for observado o gráfico do SFDR entre 150MHz e 250MHz,



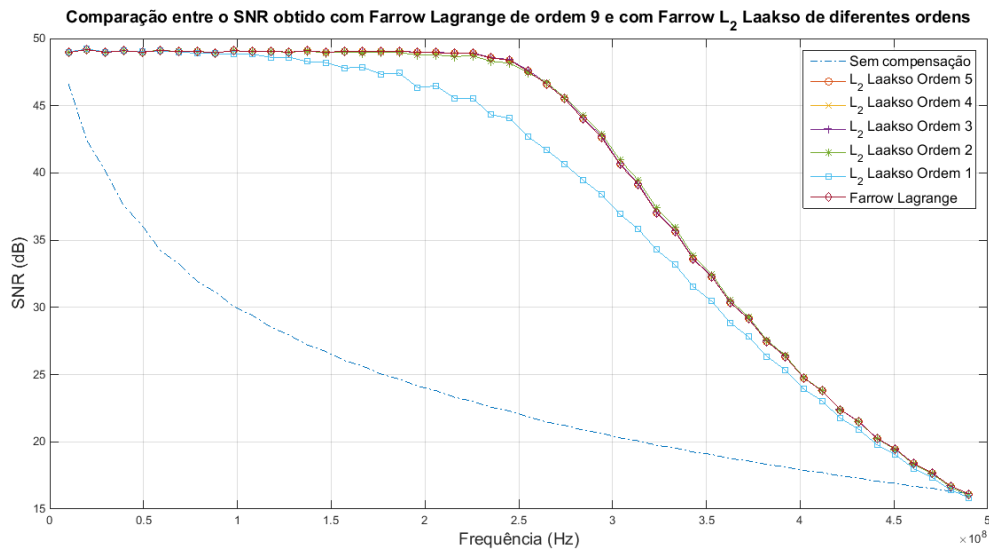
**Figura 36: Gráfico da comparação de ordem da estrutura de Farrow Lagrange.**

**Fonte: Autoria própria**

na figura 38, é possível observar uma degradação na resposta da estrutura  $L_2$  Laakso de segunda ordem. Entretanto, no gráfico do SNR, na figura 37, não é possível obter a mesma conclusão, pois a degradação observada é de pequena amplitude. Situação similar foi observada ao serem realizadas outras comparações entre estruturas e filtros. Por esta razão o SNR é preterido em algumas das análises e comparações entre os filtros e estruturas, pois diferenças significativas de desempenho entre os filtros podem não ser observadas no SNR. A razão para esta diferença deve-se ao sinal de teste, um único tom senoidal. O sinal espúrio gerado por este tom é essencialmente outro tom em uma única frequência no espectro. Como no cálculo do SNR é utilizado o valor médio do RSS de todos os demais componentes espectrais (ver subseção 2.1.3), o peso do tom espúrio no valor final do SNR é reduzido.

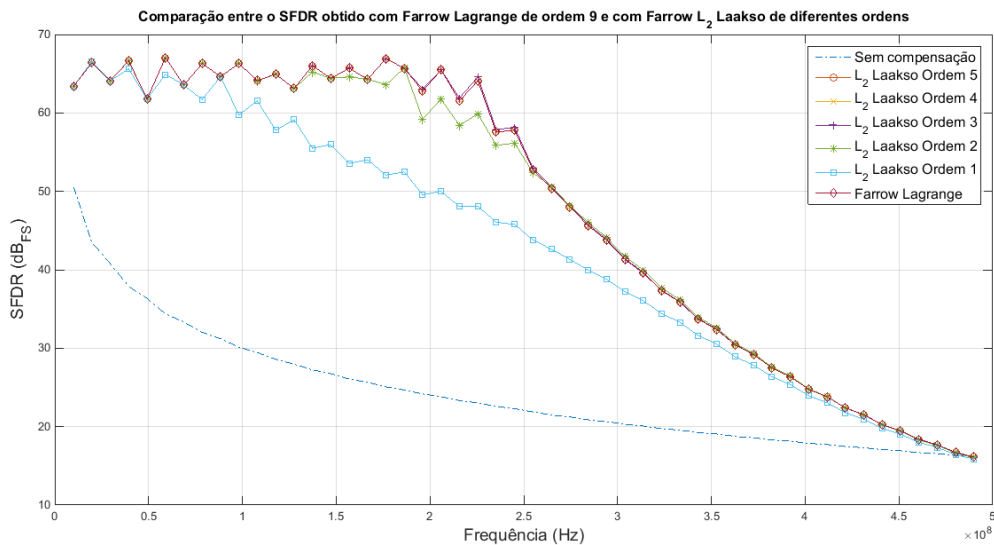
A vantagem da estrutura  $L_2$  Laakso de terceira ordem é o número reduzido de filtros FIR necessários para implementá-la, comparando-a com a estrutura de Farrow Lagrange. Esta vantagem também pode ser observada na implementação em ponto fixo destas duas estruturas, como é apresentado no SFDR da figura 39. Neste gráfico é apresentada a comparação da estrutura de Farrow Lagrange de ordem 9 e a  $L_2$  Laakso com polinômio de terceira ordem implementadas em ponto fixo. Na legenda a descrição  $WL(X,Y)$  significa que o *word length* total é de  $X$  bits, contendo parte fracionária de  $Y$  bits.

Na sequência é apresentado na figura 40 o gráfico do SFDR obtido com os diferentes filtros de *delay* fracionário discutidos no capítulo 3. O gráfico é obtido da simulação do modelo apresentado na figura 26, na subseção 4.2.1, para um *clock skew* de 10% do intervalo de



**Figura 37:** Gráfico do SNR para comparação da estrutura de Farrow Lagrange com  $L_2$  Laakso.

Fonte: Autoria própria

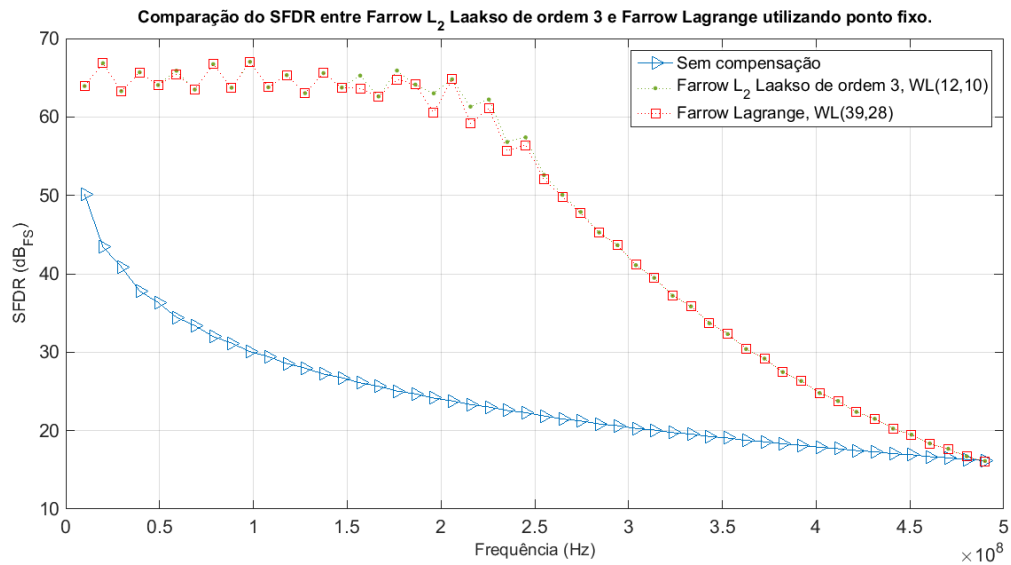


**Figura 38:** Gráfico do SFDR para comparação da estrutura de Farrow Lagrange com  $L_2$  Laakso.

Fonte: Autoria própria

aquisição.

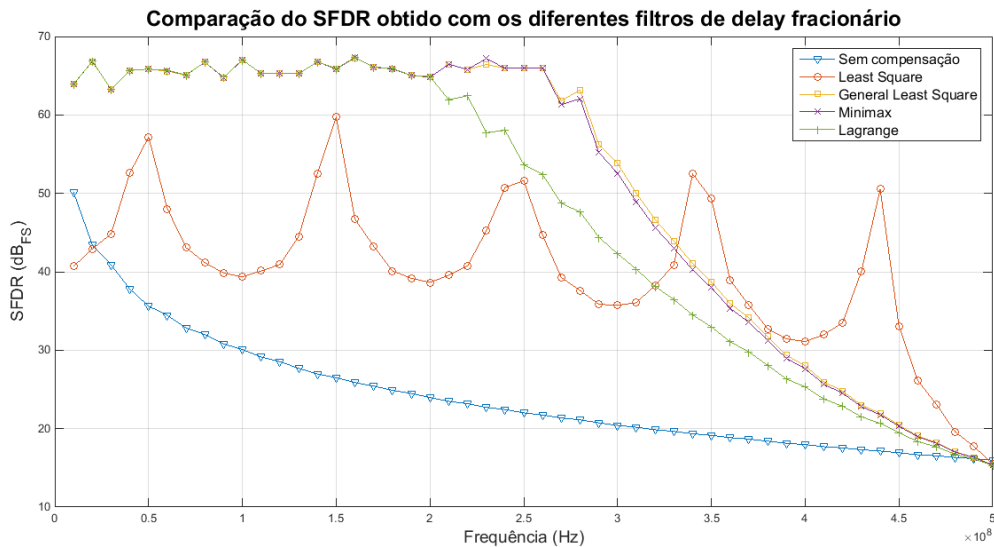
Do gráfico é evidente a desvantagem do uso do filtro *LS* e o ganho obtido com os filtros *GLS* e *Minimax* sobre o filtro de *Lagrange*. Os filtros *GLS* e *Minimax* apresentaram um desempenho similar, havendo pequeno ganho do *GLS* sobre o *Minimax* para frequências acima de 250MHz, fora da banda principal de interesse entretanto. O desempenho similar entre



**Figura 39:** Gráfico do SFDR para comparação da estrutura de Farrow Lagrange com  $L_2$  Laakso em ponto fixo.

Fonte: Autoria própria

estes dois filtros permitiu a seleção com base em outro aspecto: a simplicidade de projeto. Por permitir o uso de expressões fechadas, seguindo as observações discutidas na subseção 3.1.2, o filtro GLS foi selecionado.

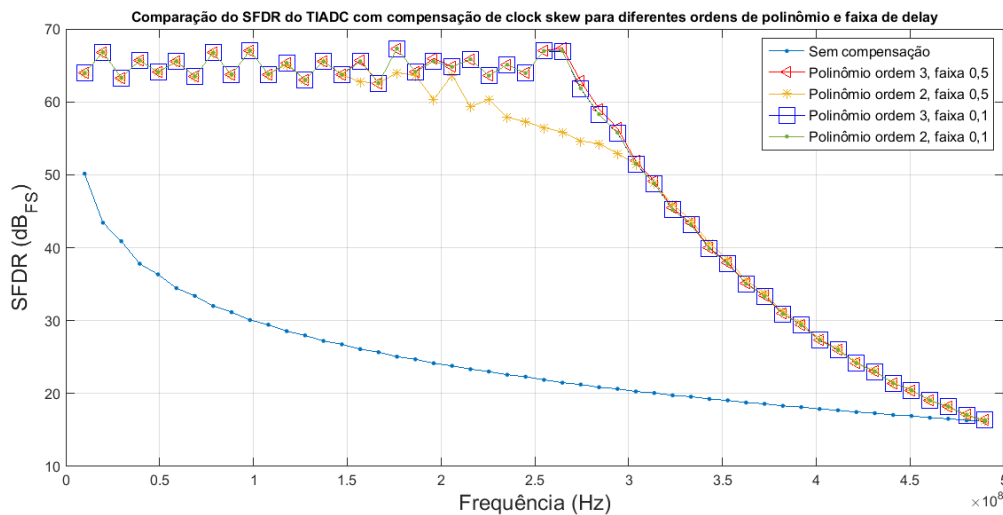


**Figura 40:** Gráfico do SFDR para comparação de diferentes filtros de delay fracionário.

Fonte: Autoria própria

Como discutido na subseção 4.2.1, definido o filtro GLS, as próximas etapas constituem em definir o *word length* da estrutura, a ordem do polinômio e a faixa de *delay*

da estrutura de Farrow. Para avaliar o impacto da faixa do *delay* foi realizada uma simulação com estruturas  $L_2$  Laakso de segunda e terceira ordem, com faixa de *delay* de  $[0; 1/2]$  e  $[0; 0, 1]$ . O SFDR obtido do TIADC após compensação por cada uma destas estruturas é apresentado na figura 41. Pode-se observar da figura 41 uma pequena diferença entre as duas estruturas com polinômio de terceira ordem e diferentes faixas de *delay*. Entretanto, é observada uma clara diferença de desempenho para as duas estruturas com polinômio de segunda ordem. Para a estrutura com faixa de *delay* de até  $1/2$ , o SFDR entre 200MHz e 300MHz é até 10dB menor do que para a estrutura com faixa de *delay*  $[0; 0, 1]$ . O desempenho das duas estruturas com polinômio de terceira ordem é similar ao desempenho da estrutura com polinômio de segunda ordem e faixa de *delay* de até 0,1. Esta similaridade sugere avaliar a possibilidade de utilizar um polinômio de segunda ordem, o que significa utilizar menos um filtro FIR na estrutura de Farrow.

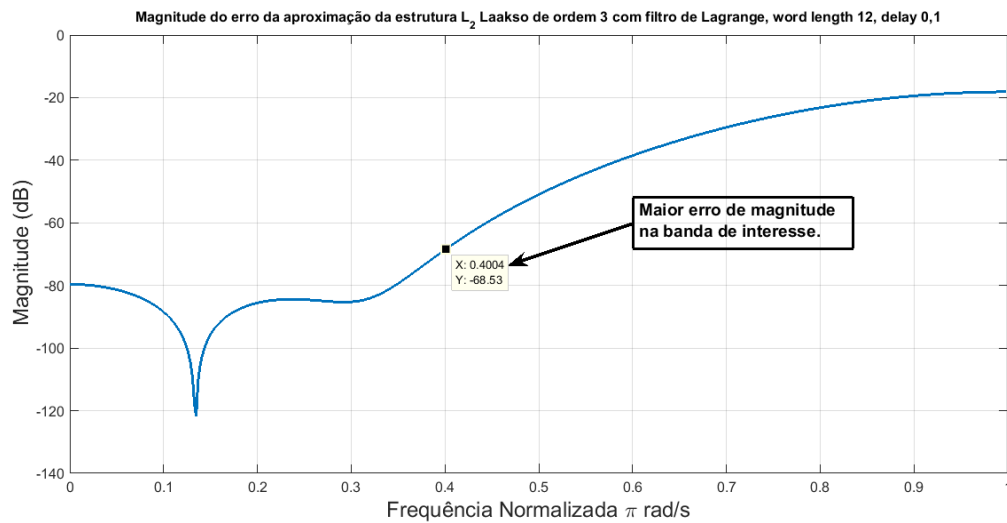


**Figura 41:** Gráfico do SFDR para comparação da ordem e da faixa de *delay* da estrutura  $L_2$  Laakso com filtro GLS.

**Fonte:** Autoria própria

A determinação do *word length* é realizada utilizando-se das duas estratégias complementares descritas na subseção 4.2.1. Para utilizá-las é necessário definir um valor de erro máximo de aproximação para a magnitude da resposta em frequência da estrutura de Farrow. A definição é obtida do erro obtido com a estrutura  $L_2$  Laakso com polinômio de terceira ordem e filtro de Lagrange utilizado para comparação com a estrutura de Farrow Lagrange. Na figura 42 é apresentada a curva de erro da magnitude desta estrutura. O valor de erro é selecionado na frequência de 200MHz (o equivalente a frequência normalizada de  $0,4 \pi rad/s$ ), valor de frequência limite da banda de interesse do filtro. O valor que foi definido para o erro é de -68 dB.





**Figura 42:** Gráfico da magnitude do erro de aproximação da estrutura de Farrow  $L_2$  Laakso com filtro de Lagrange.

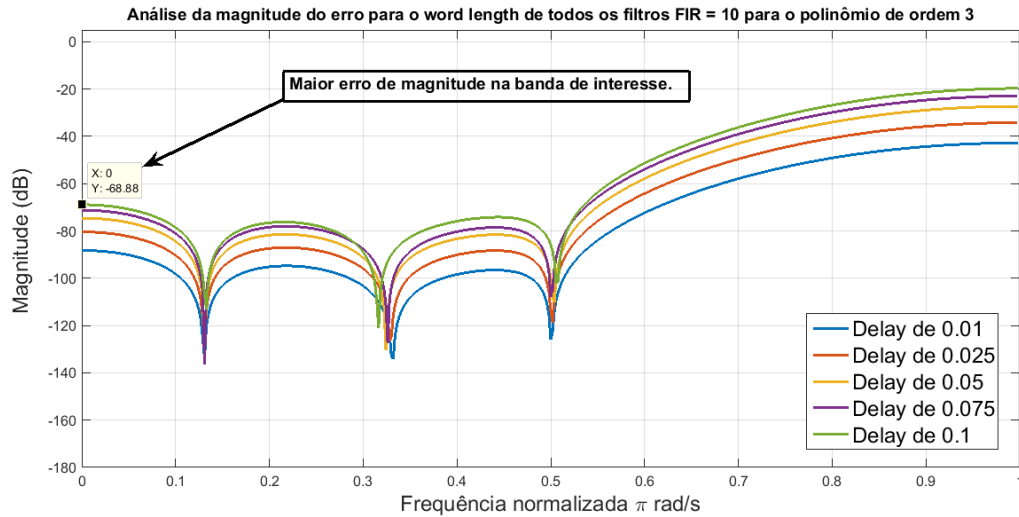
**Fonte:** Autoria própria

Desta forma foi avaliado o *word length* para as estruturas de segunda e terceira ordem, para a estratégia em que os coeficientes de todos os filtros FIR possuem o mesmo *word length*, e para a estratégia em que estes podem ser diferentes. Um exemplo do gráfico de erro obtido para a estrutura de terceira ordem e *word length* igual para os coeficientes de todos os filtros FIR é apresentado na figura 43. Na figura 43 pode-se observar o erro de magnitude para diferentes valores de *delay* fracionário, considerando o *word length* de 10 bits. O erro de magnitude é menor para os menores valores de *delay* fracionário. Dentro da região da banda de interesse a maior magnitude de erro é de -68,88 dB, menor do que o encontrado no gráfico da figura 42.

Para poder comparar as quatro estruturas de Farrow são definidos parâmetros, sendo eles:

- SFDR mínimo para a banda de 200MHz;
- SFDR mínimo para a faixa entre 200MHz e 250MHz;
- SNR mínimo para a banda de 200MHz;
- SNR mínimo para a faixa entre 200MHz e 250MHz;
- LUTs (*Look Up Table*) requeridos para implementação;
- registradores requeridos para implementação;

- frequência máxima de operação da estrutura (limitada pelo acúmulo dos atrasos de propagação dos elementos da FPGA).



**Figura 43:** Gráfico da magnitude do erro de aproximação da estrutura de Farrow  $L_2$  Laakso com filtro GLS e *word length* dos coeficientes iguais.

Fonte: Autoria própria

A comparação entre as estruturas pode ser verificada na tabela 4. Na descrição de *word length* da tabela 4 a sequência dentro do parênteses informa o *word length* dos filtros FIR da estrutura de Farrow. O primeiro valor é o filtro FIR do coeficiente de grau mais elevado na estrutura, e o último valor o filtro do coeficiente  $C_0$ . A coluna *Ordem* refere-se a ordem do polinômio da estrutura de Farrow. Dos dados apresentados na tabela 4 pode-se concluir que o desempenho na compensação do *clock skew* de todas as estruturas é similar. Entretanto, a utilização de recursos da FPGA (LUTs e Reg.) e a frequência máxima de operação ( $F_{máx}$ ) possuem diferenças significativas. Pode-se observar que as estruturas com polinômio de terceira ordem utilizam mais LUTs da FPGA e operam em frequências menores do que as estruturas de segunda ordem. Por possuir uma diferença de um filtro FIR, a estrutura de segunda ordem requer um número menor de elementos, justificando a menor utilização de LUTs. O filtro FIR adicional na estrutura de terceira ordem também requer mais um multiplicador e somador no caminho crítico da estrutura de Farrow, o que justifica a sua menor frequência de operação.

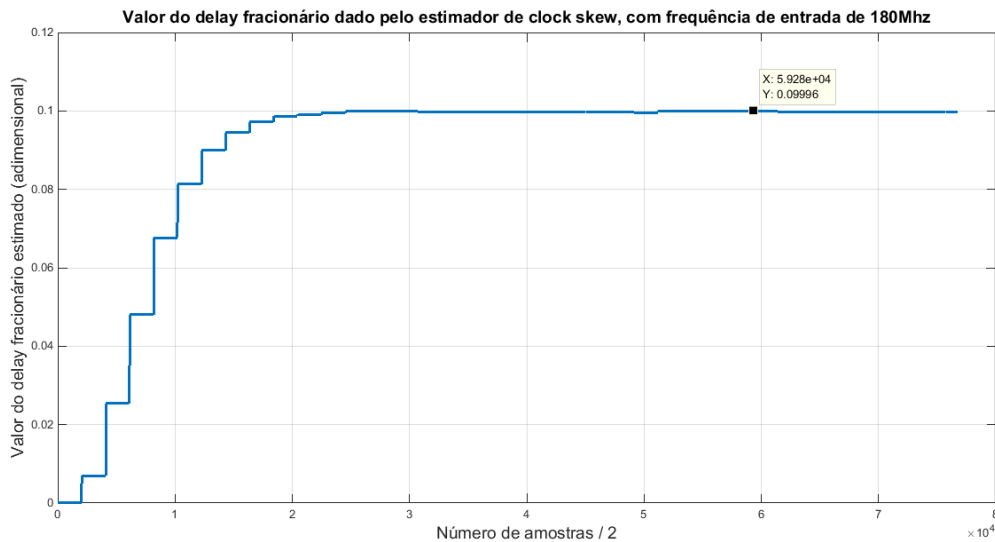
Pode-se verificar também que o uso de *word length* diferentes entre os filtros FIR da estrutura permitiu a redução na utilização de LUTs da FPGA, entretanto reduziu também a frequência máxima de operação da estrutura. Para a implementação da estrutura completa de compensação de *clock skew*, com o estimador incluído, foi selecionada a estrutura com polinômio de segunda ordem e *word length* diferente para os filtros. Os parâmetros do filtro

**Tabela 4: Tabela de comparação entre as estruturas de Farrow com filtro GLS.**

Ordem	Word length	SFDR	SFDR	SNR	SNR	LUTs	Reg.	$F_{m\acute{a}x}$ (MHz)
		200MHz	250MHz	200MHz	250MHz			
		( $dB_{FS}$ )	( $dB_{FS}$ )	(dB)	(dB)			
3	(10, 10, 10, 10)	62,5621	63,5879	48,8826	48,9605	2598	104	16,96
2	(10, 10, 10)	62,5635	63,5881	48,8641	48,9606	1552	104	21,15
3	(9, 9, 10, 10)	62,5622	63,5882	48,8718	48,9601	2507	104	12,91
2	(9, 10, 10)	62,5637	63,588	48,864	48,96	1487	104	20,48

**Fonte: Autoria prpria**

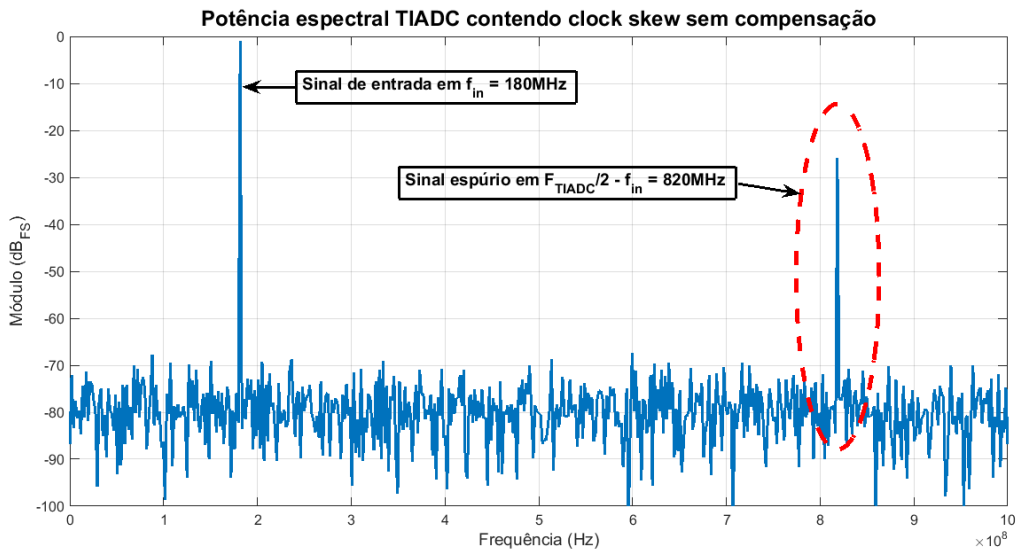
selecionado esto na ltima linha da tabela 4. Esta estrutura foi selecionada por apresentar o menor custo, em termos de unidades lgicas, de *hardware*. A reduo percentual no uso de unidades lgicas da estrutura selecionada em relao a estrutura com polinmio de terceira ordem e *word length* igual  de aproximadamente 42%. Para as duas implementaes com polinmio de segunda ordem a diferena  muito menor, sendo que a estrutura com *word length* diferentes utiliza 4% menos LUTs do que a estrutura com *word length* iguais. Deve-se observar que todas as estruturas de Farrow implementadas fornecem um valor de sada do filtro em um nico ciclo de *clock*, no havendo latncia devido a *buffers* de memria ou ao uso de *pipeline*.

**Figura 44: Curva do valor estimado de *delay* para compensao do *clock skew*.**

**Fonte: Autoria prpria**

Assim como no sistema de compensao de ganho, o sistema de compensao de *clock skew* opera com a realimentao de um sinal de erro. Este sistema requer um intervalo de tempo para a estabilizao do valor estimado de desvio do sinal de entrada. O tempo para estimativa do *delay* necessrio para a estrutura de Farrow pode ser visualizado na figura 44. Para esta

simulação são apresentados o espectro do TIADC antes da compensação e após a compensação de *clock skew*, com o valor do *delay* estável, nas figuras 45 e 46, respectivamente. Como destacado na figura 45, o desvio de *clock skew* causa um sinal espúrio em 820MHz para um sinal senoidal de entrada em 180MHz. Este mesmo sinal espúrio é atenuado pela compensação de *clock skew*, como pode ser observado na figura 46, possuindo potência menor ou igual a potência do *jitter* nesta frequência.



**Figura 45: Espectro do TIADC com *clock skew* e sem compensação.**

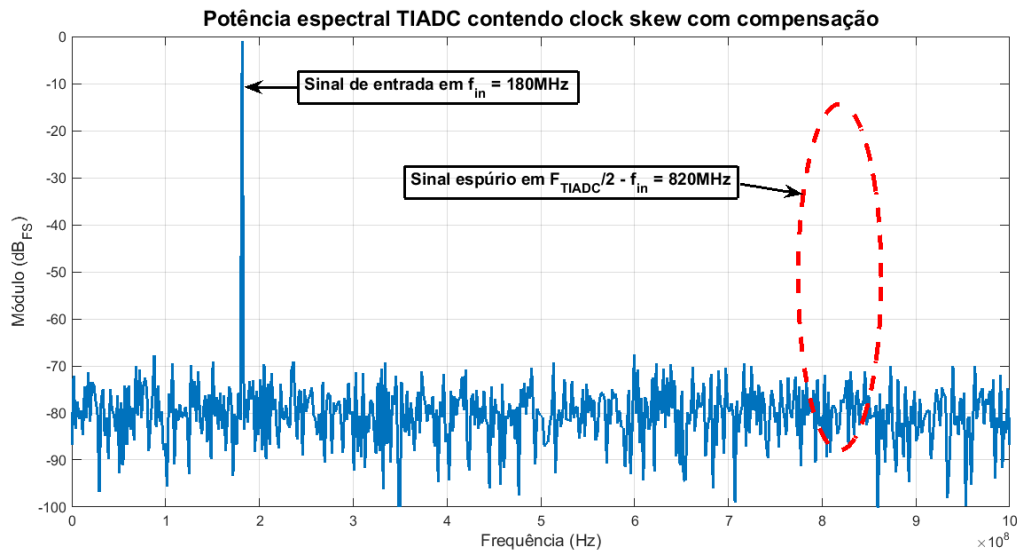
**Fonte: Autoria própria**

O desempenho do TIADC antes e depois da compensação de *clock skew* neste teste pode ser avaliado pelas métricas de desempenho definidas na seção 2.1.3, como é apresentado na tabela 5.

**Tabela 5: Tabela de comparação do desempenho do TIADC antes e depois da compensação de *clock skew* em 180MHz.**

Compensação	SNR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
Inativa	24,87	24,87	24,89	3,84
Ativa	49,01	48,98	66,69	7,84

**Fonte: Autoria própria**



**Figura 46: Espectro do TIADC com *clock skew* e com compensação.**

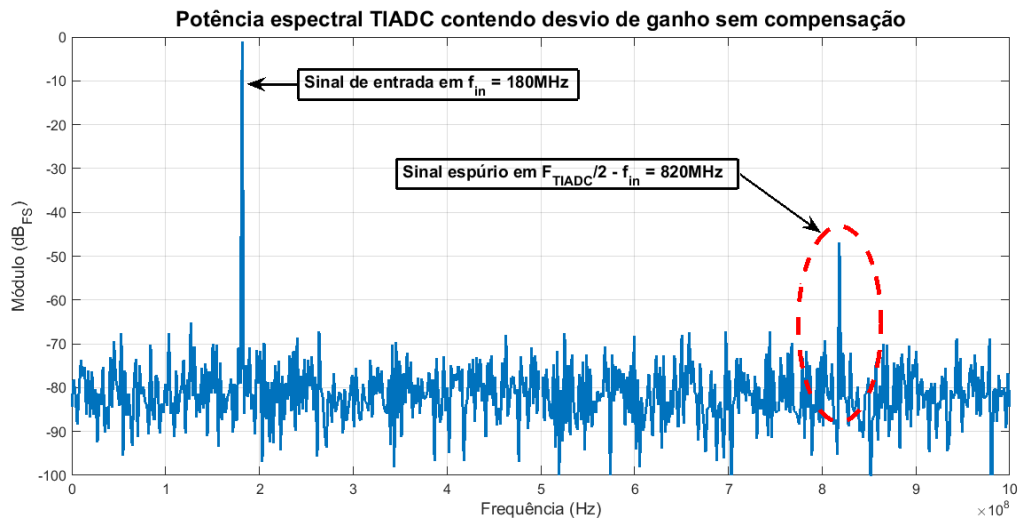
**Fonte: Autoria própria**

## 5.2 GANHO

O efeito no espectro do desvio de ganho para o TIADC de dois canais é um sinal espúrio em  $F_{TIADC}/2 - f_{in}$ , como pode ser visualizado no espectro do TIADC apresentado na figura 47. Este espectro foi obtido com a simulação de um desvio de 1% do ganho entre os dois canais do TIADC, onde o ADC1 possui ganho de 0,99 e o ADC2 ganho igual a 1,0. O sinal de entrada  $f_{in}$  possui frequência de 180MHz e amplitude de 90% do fundo de escala. O SNR deste TIADC é de 44,14 dB e o SFDR é de 45,88 dB. O valor de desvio de ganho foi selecionado como um exemplo do impacto de pequenos desvios no desempenho do TIADC. Valores de desvio de ganho entre 2 e 3 por cento são sugeridos para simulação em (FERRÉ et al., 2008) e (VOGEL, 2005).

O espectro deste TIADC após a compensação do desvio de ganho é apresentado na figura 48. Após a compensação o SNR do TIADC é de 48,96 dB e o SFDR é de 64,13 dB, um ganho de 4,82 dB no SNR e de 18,25 dB no SFDR. A tabela 6 apresenta o desempenho do TIADC utilizando as métricas apresentadas na subseção 2.1.3.

Deve-se observar que o sistema de compensação de ganho opera com a realimentação de um estimador de erro, requerendo portanto um período para estabilização e convergência do valor do multiplicador de compensação do ganho. Este período deve ser pequeno o suficiente para garantir a estabilidade do compensador de ganho. Um período muito longo em relação à variação do desvio de ganho torna o compensador ineficiente, podendo inclusive tornar o



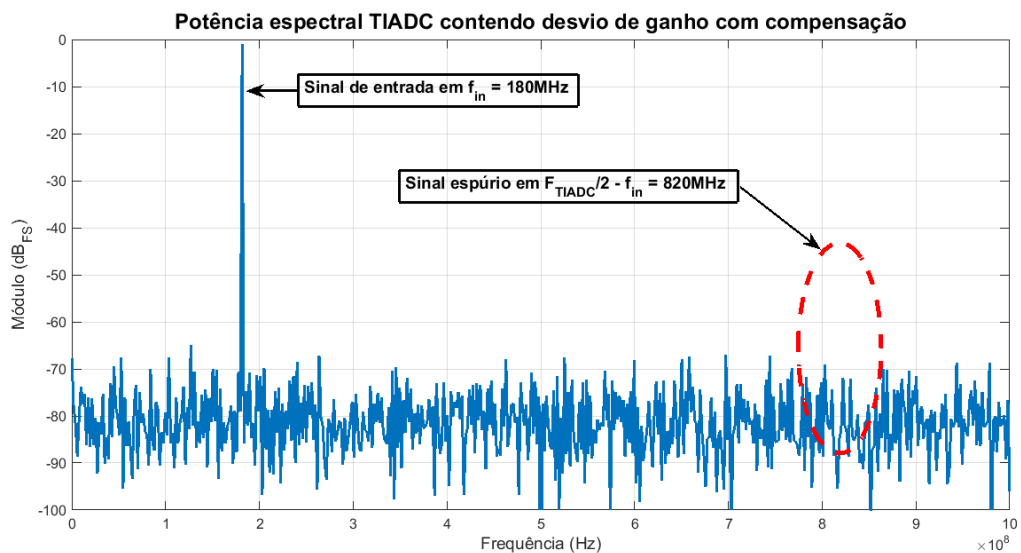
**Figura 47: Espectro TIADC com desvio de ganho e sem compensação.**

**Fonte: Autoria própria**

**Tabela 6: Tabela de comparação do desempenho do TIADC antes e depois da compensação de ganho em 180MHz.**

Compensação	SNR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
Inativa	44,14	48,96	45,88	7,04
Ativa	48,96	48,93	64,13	7,84

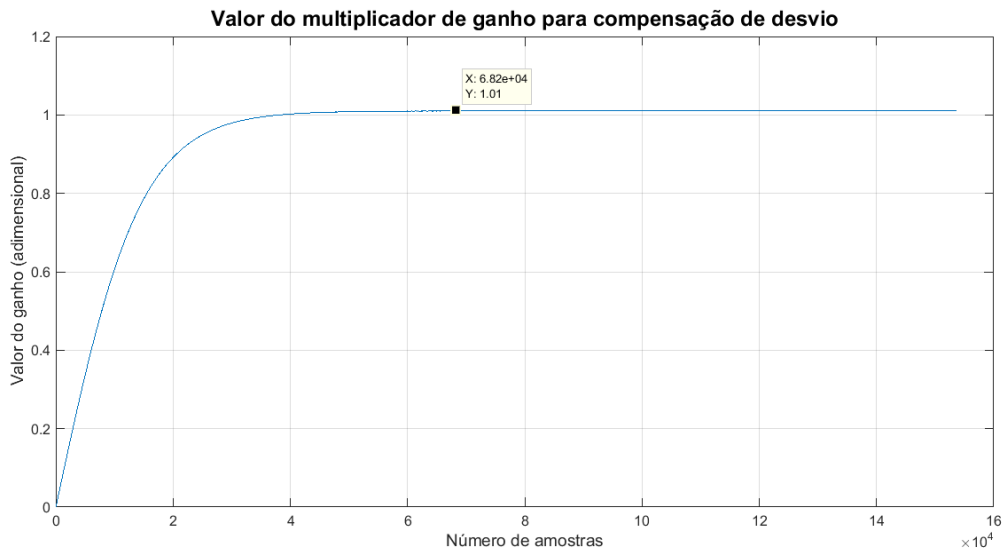
**Fonte: Autoria própria**



**Figura 48: Espectro TIADC com desvio de ganho e com compensação.**

**Fonte: Autoria própria**

sistema instável. O período de estabilização pode ser observado na figura 49. Da figura pode-se observar que foram necessárias aproximadamente 68 mil amostras para estabilização do valor do multiplicador de ganho, considerando que a sua condição inicial era o valor zero. Para o TIADC com 2GHz de frequência de amostragem, isto significa que o desvio de ganho inicial do TIADC pode ser compensado em aproximadamente 34 microsegundos de operação.

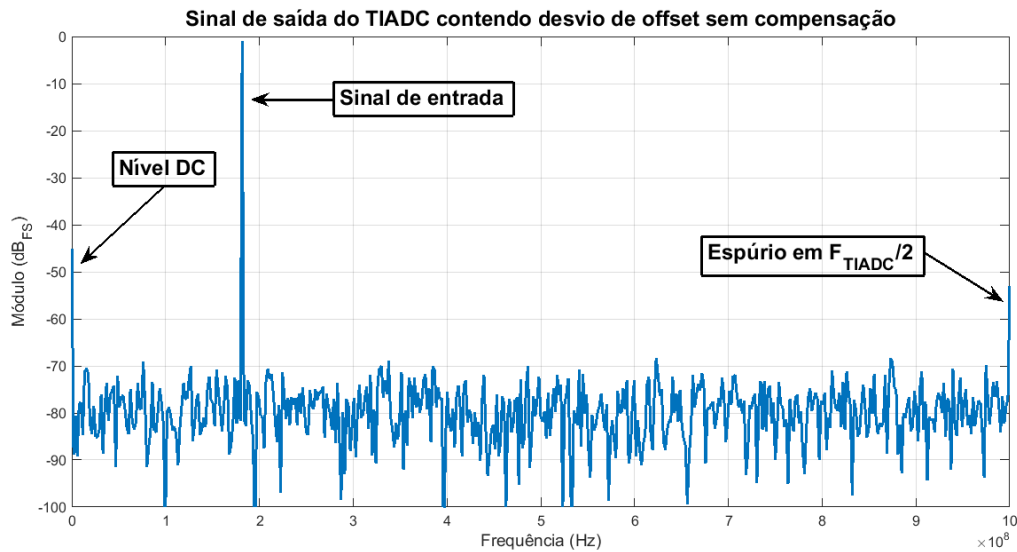


**Figura 49: Curva do multiplicador de ganho ao longo do tempo.**

**Fonte: Autoria própria**

### 5.3 OFFSET

Como apresentado na subseção 2.3.1, o desvio de *offset* entre os conversores do TIADC produz sinais espúrios em frequências múltiplas de  $F_{TIADC}/2$ . Para avaliar o desempenho dos compensadores de *offset*, descritos na subseção 4.2.3, foi realizada uma simulação em que o *offset* do ADC1 é parametrizado para 0,5 LSB (*Least Significant Bit*) e o *offset* do ADC2 para 0,25 LSB. O efeito destes desvios pode ser observado no espectro obtido da simulação do TIADC, considerando um sinal senoidal de entrada  $f_{in}$ , com frequência de 180MHz e amplitude de 90% do fundo de escala, apresentado na figura 50. Os valores de desvio de *offset* para o TIADC podem ser relativamente elevados, como por exemplo 5 LSB de *offset*, sugerido para simulações em (FERRÉ et al., 2008). Entretanto, deve-se observar que a sensibilidade do TIADC em relação aos desvios é elevada, aspecto que já foi observado na primeira proposição da estrutura em (BLACK JR.; HODGES, 1980). Os valores de *offset* que foram selecionados para esta simulação visam ilustrar o impacto de pequenos *offsets*, menores que 1 LSB, e a capacidade do sistema de compensação de atenuá-los.



**Figura 50: Espectro TIADC com desvio de offset e sem compensação.**

**Fonte: Autoria própria**

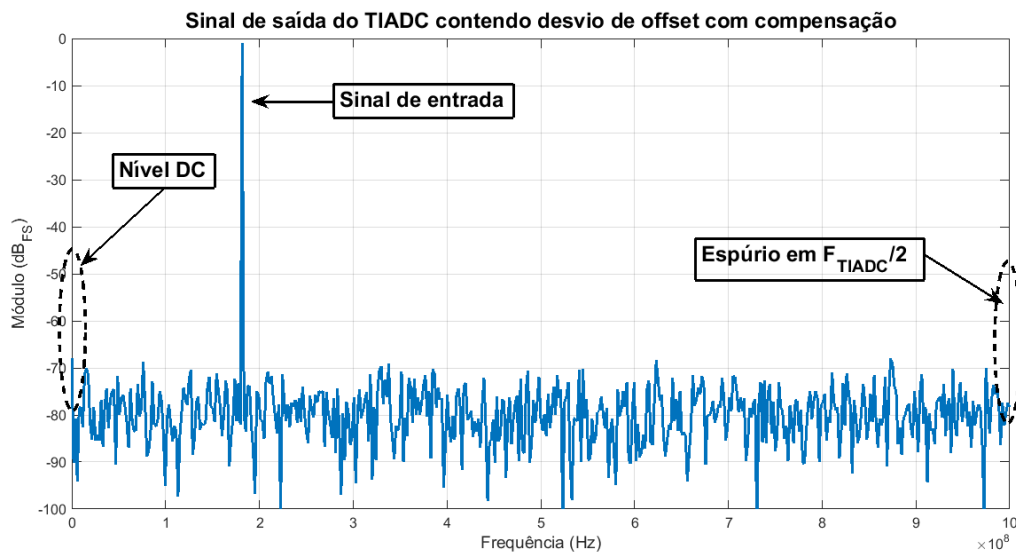
Na figura 51 pode ser visualizada a atenuação do sinal espúrio devido a compensação do *offset*. Pode-se observar também que o sinal de nível DC também foi atenuado. Esta foi uma opção de projeto do compensador de *offset*. A alternativa para esta situação seria a compensação da diferença do *offset* entre os canais, assumindo um valor único, de um dos ADCs, ou o valor médio dos dois ADCs para o *offset* do TIADC. A comparação no ganho de desempenho do TIADC com a compensação do *offset* pode ser realizada verificando-se a diferença nas métricas de desempenho apresentadas na subseção 2.1.3. As mesmas são apresentadas na tabela 7. Antes da compensação o SFDR é 52,01 dB, e após a compensação o SFDR é igual a 67,16 dB, um ganho de ao menos 15 dB.

**Tabela 7: Tabela de comparação do desempenho do TIADC antes e depois da compensação de *offset* em 180MHz.**

Compensação	SNR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
Inativa	47,25	47,21	52,01	7,55
Ativa	49,02	48,96	67,16	7,84

**Fonte: Autoria própria**





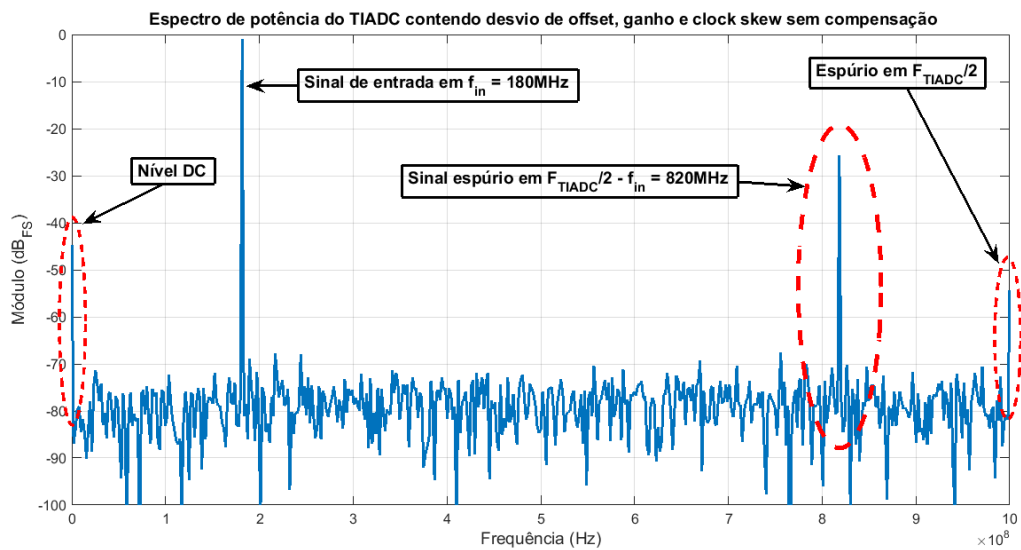
**Figura 51: Espectro TIADC com desvio de *offset* e com compensação.**

**Fonte: Autoria própria**

#### 5.4 SISTEMA COMPLETO

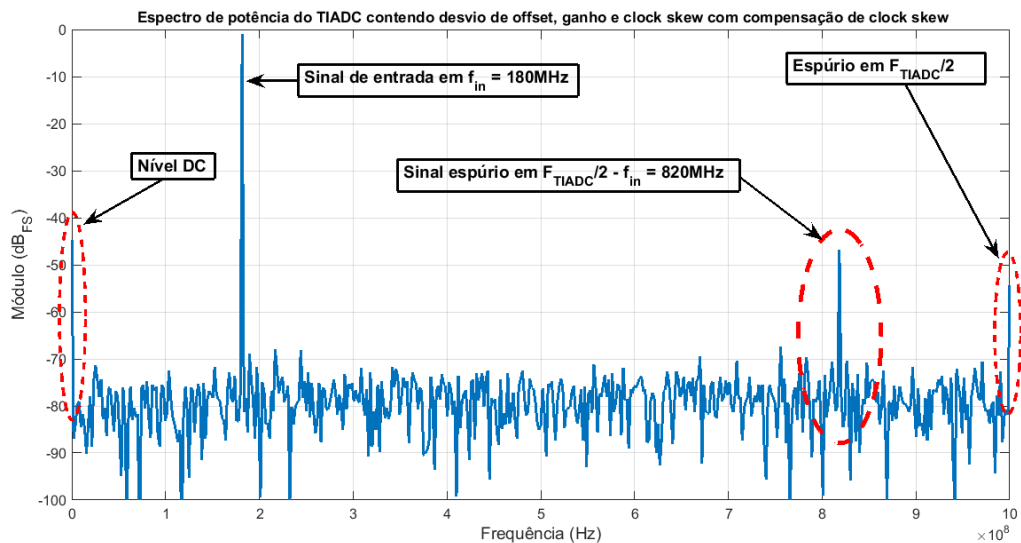
Nas seções anteriores foram apresentados os resultados individuais de cada um dos compensadores de desvio, considerando a atuação isolada de cada desvio. Entretanto, em um TIADC os desvios ocorrem simultaneamente, sendo necessário que todos os sistemas de compensação e estimação dos desvios operem em conjunto, no mesmo sistema. A principal dificuldade encontra-se nos estimadores, que individualmente devem ser capazes de estimar somente um dos desvios com um mínimo de erro devido aos demais. A robustez do sistema pode ser verificada pelo espectro do TIADC nas figuras 52, 53, 54 e 55. A figura 52 apresenta o espectro do TIADC contendo todos os desvios discutidos simultaneamente. Nas demais figuras as compensações dos desvios são realizadas na ordem: *clock skew*, figura 53; ganho, figura 54; e *offset* na figura 55. Deve ser observada a diferença na amplitude do sinal espúrio em 820MHz na figura 53, com relação a figura 52. O sinal espúrio que aparece na figura 53 é devido ao desvio de ganho somente, sendo que o sinal espúrio do *clock skew* é atenuado ao mesmo nível de potência do *jitter*. Os valores de desvio utilizados nesta simulação são os mesmos utilizados na compensação individual dos desvios nas seções anteriores. A tabela 8 apresenta o desempenho do TIADC para toda a sequência de compensação que é realizada.

A utilização de recursos e a frequência máxima de cada estimador e compensador são apresentados na tabela 9, assim como são apresentados estes dados para o sistema completo. Nesta mesma tabela estão separados os dois modelos de estimador de *clock skew*,



**Figura 52: Espectro do TIADC com todos os desvios e sem compensação.**

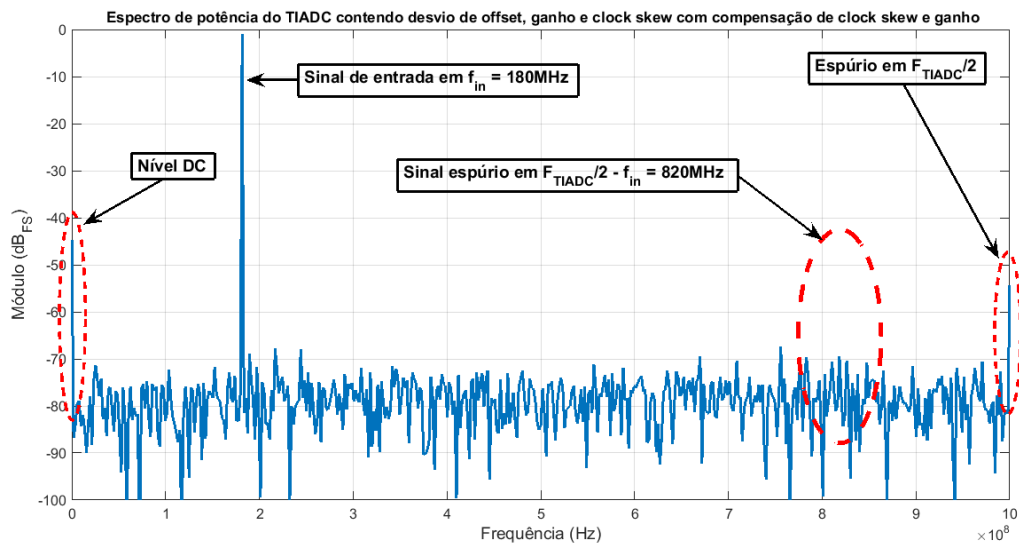
Fonte: Autoria própria



**Figura 53: Espectro do TIADC com todos os desvios e compensação do *clock skew*.**

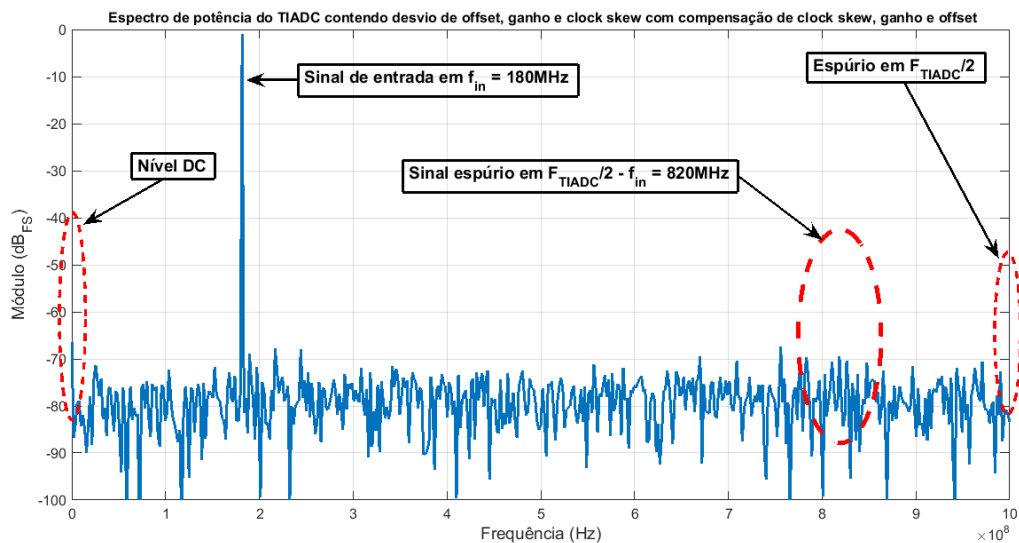
Fonte: Autoria própria

o originalmente proposto em (RAZAVI, 2013) e a sua versão simplificada. O objetivo em apresentar a utilização de recursos para cada parte do sistema é o de permitir a visualização de quais elementos requerem maior quantidade de recursos da FPGA e quais elementos apresentam melhor desempenho. O sistema completo não foi sintetizado de forma incremental, portanto poderá haver otimizações pelo *software* Quartus durante a síntese do projeto. Por esta razão o simples somatório dos recursos utilizados por cada elemento do sistema não é igual a quantidade



**Figura 54:** Espectro do TIADC com todos os desvios e compensação do *clock skew* e do ganho.

Fonte: Autoria própria



**Figura 55:** Espectro do TIADC com todos os desvios e compensação de todos os desvios.

Fonte: Autoria própria

total de recursos do sistema de compensação. Um ponto interessante é a diferença na quantidade de LUTs utilizadas nos dois estimadores de *clock skew*. A versão simplificada do estimador utiliza aproximadamente 66% da quantidade de LUTs utilizada pela versão original.

Outros aspectos que podem ser observados é o desempenho do estimador e compensador de *offset*. A utilização de poucos recursos da FPGA e a elevada frequência de

**Tabela 8: Tabela de comparação do desempenho do TIADC antes e depois do sistema de compensação em 180MHz.**

Compensação	SNR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
Nenhuma	24,79	24,79	24,81	3,82
<i>Clock skew</i> somente	43,73	43,72	45,94	6,97
<i>Clock skew</i> e ganho	47,69	47,66	53,43	7,63
Todos os desvios	49,03	49,00	66,51	7,85

**Fonte: Autoria própria**

operação são justificados pela simplicidade da estrutura. Situação diferente é observada nos estimadores de ganho e *clock skew*, em especial o estimador de ganho, que é o estimador que requer mais recursos da FPGA. Esta diferença existe devido a quantidade de operações, ou seja, a quantidade de elementos, necessários para realizar a estimativa de ganho. É evidente que o subsistema que limita a frequência máxima de operação do sistema é o de compensação de *clock skew*. Este resultado é justificado pela quantidade de multiplicadores e somadores presentes no caminho crítico da estrutura de Farrow que foi implementada.

**Tabela 9: Tabela de utilização de recursos e desempenho dos estimadores e compensadores de desvio.**

Estrutura	LUTs	Registradores	$F_{m\acute{a}x}$ (MHz)
Estimador Offset	164	79	89,27
Estimador Ganho	631	173	46,92
Estimador Clock Skew Razavi Original	540	134	45,38
Estimador Clock Skew Razavi Simplificado	356	134	42,56
Compensador Offset	49	48	69,54
Compensador Ganho	1107	59	42,63
Compensador Clock Skew	1349	104	20,92
Sistema completo	3605	532	22,58

**Fonte: Autoria própria**

## 5.5 SIMULAÇÃO COM FPGA IN LOOP

Na seção 4.3 foi apresentado o conceito da metodologia de teste com *FPGA In Loop*, utilizada para validação dos sistemas de compensação. Naturalmente, o projeto do sistema completo para validação, contendo a interface de comunicação com o computador, requer uma quantidade maior de recursos da FPGA. A utilização de recursos pela ferramenta para validar o sistema de compensação completo é apresentada na tabela 10.

Foi realizado um teste de validação com os mesmos parâmetros da simulação realizada na sessão 5.4, como um exemplo da funcionalidade do sistema implementado em FPGA. Os

**Tabela 10: Tabela de utilização de recursos para o projeto utilizando *FPGA In Loop*.**

Recurso da FPGA	Quantidade	Percentual utilizado do dispositivo (%)	Total disponível na FPGA
LUTs	5,788	5	114.480
Registradores	2302	2	114.480
Memória ( <i>bits</i> )	113.296	3	3.981.312
PLL ( <i>Phase Locked Loop</i> )	1	25	4

**Fonte: Autoria própria**

resultados deste teste são apresentados na tabela 11, contendo as métricas de desempenho do TIADC antes e depois do sistema de compensação de desvios. O tempo para a realização deste teste foi de aproximadamente 8 minutos. Este intervalo de tempo para o teste com *FPGA In Loop* é justificado pelo uso da interface JTAG, que é uma interface de comunicação lenta, sendo esta o principal gargalo no tempo do teste.

Comparando-se a tabela 11 com a tabela 8, da seção 5.4, pode-se observar que o sistema de compensação implementado em FPGA obteve os mesmos resultados do sistema simulado em Simulink somente. Outros testes foram realizados com sinais de diferentes frequências, obtendo-se resultados similares. Estes resultados validaram a implementação em FPGA dos compensadores de desvio do TIADC.

**Tabela 11: Tabela de comparação do desempenho do TIADC antes e depois do sistema de compensação em 180MHz. Teste com *FPGA In Loop*.**

Compensação	SNR ( <i>dB</i> )	SINAD ( <i>dB</i> )	SFDR ( <i>dB</i> )	ENOB ( <i>bits</i> )
Nenhuma	24,79	24,79	24,81	3,82
<i>Clock skew</i> somente	43,73	43,72	45,94	6,97
<i>Clock skew</i> e ganho	47,69	47,67	53,44	7,63
Todos os desvios	49,04	49,00	66,51	7,85

**Fonte: Autoria própria**

## 6 CONCLUSÕES E TRABALHOS FUTUROS

O conceito de conversores inteligentes apresentado em (ROERMUND et al., 2005) possui o TIADC como um excelente exemplo. A sensibilidade desta estrutura em relação aos seus desvios torna o uso de técnicas tradicionais de pareamento de circuitos analógicos ineficientes. Por esta razão, sistemas digitais de compensação dos desvios do TIADC tornaram-se populares e amplamente pesquisados. Entretanto, mesmo após três décadas de pesquisa, os desvios do TIADC ainda são um desafio, em especial o desvio de *clock skew*.

Como observado em (MARIANO, 2008), dispositivos envolvendo intensa interação entre componentes analógicos e digitais clamam por metodologias de desenvolvimento *top-down*. Estas metodologias requerem o uso de modelos comportamentais com diferentes níveis de abstração. O foco deste trabalho foi a construção de modelos comportamentais do TIADC e de seu sistema de compensação. Assim como a implementação e validação deste sistema de compensação em FPGA, utilizando os modelos comportamentais construídos anteriormente.

Neste trabalho os modelos comportamentais foram construídos utilizando o *software* Simulink. Com o modelo da parte analógica do TIADC foi possível emular cada um de seus principais desvios, sendo eles: *offset*, ganho e *clock skew*. Na parte digital do TIADC, os modelos permitiram analisar e avaliar sistemas de compensação e estimação para cada desvio. O uso destes modelos permitiu o estudo comparativo entre diferentes filtros para a compensação de *clock skew*. Este estudo teve como principal objetivo a avaliação do desempenho e da relação custo benefício de cada filtro. O estudo demonstrou que o filtro GLS fornece melhor desempenho e relação custo benefício em relação aos demais filtros estudados, considerando a ordem do filtro e a sua complexidade de projeto. O mesmo estudo também demonstrou as vantagens, e a flexibilidade, do projeto da estrutura de Farrow através do método  $L_2$  Laakso. Com este método de projeto foi possível otimizar o subsistema de compensação de *clock skew*, pois a estrutura pôde ser projetada utilizando qualquer filtro de *delay* fracionário, utilizando a segunda ordem polinomial em  $\mu$ , com uma faixa restrita de *delay*.

O uso de uma metodologia *top-down* permitiu a otimização dos sistemas de

compensação desde a construção dos modelos comportamentais, como foi o caso da determinação do *word length* da estrutura de Farrow. Mais do que a determinação do *word length*, foi possível otimizar as estruturas de compensação no nível mais elevado de abstração. Dois exemplos em que houve otimização do sistema na construção do modelo comportamental são a discussão da ordem polinomial da estrutura de Farrow, e a redução de um multiplicador do estimador de *clock skew*. A redução da ordem polinomial em  $\mu$  da estrutura de Farrow, de terceira para segunda ordem, permitiu uma redução de aproximadamente 40% na utilização de recursos da FPGA. A eliminação de um multiplicador no estimador de *clock skew*, permitiu a obtenção de um estimador que utiliza aproximadamente 66% dos recursos utilizados do estimador original. Além do subsistema compensador de *clock skew*, foram modelados os subsistemas para compensação de *offset* e ganho. O desempenho e funcionalidade destes sistemas puderam ser avaliados em diferentes situações, utilizando-se de simulações dos modelos com diferentes parametrizações.

Além da modelagem dos sistemas de compensação em Simulink, foi obtida uma descrição em *hardware* destes compensadores, utilizando VHDL. Nesta etapa do trabalho, a ferramenta HDLCoder, do Simulink, permitiu automatizar o processo de transcrever os modelos comportamentais dos compensadores em VHDL. Deve-se observar que, apesar de funcionalmente corretos, os compensadores de desvio obtidos desta forma não possuem desempenho ótimo. Este aspecto torna-se claro quando é analisada a tabela 9, da subseção 5.4. Na tabela pode-se observar que os compensadores e estimadores operam em frequências relativamente baixas, quando comparadas com as frequências máximas de operação da FPGA. Este resultado sugere que o código VHDL gerado possa ser otimizado para a obtenção de sistemas com desempenho ótimo. Entretanto, o código VHDL gerado pela ferramenta permite a verificação e validação funcional completa dos sistemas de compensação.

A validação dos sistemas foi realizada através de sua implementação e teste em uma FPGA, em que a simulação *FPGA In Loop* mostrou-se essencial neste processo. A FIL permitiu que todo o sistema de compensação digital pudesse ser validado em *hardware*, sem exigir para isto a estrutura completa de um TIADC físico. Do ponto de vista do sistema digital, a simulação com FIL realiza as mesmas operações que seriam realizadas com um sistema de um TIADC completo. Diferentemente de uma simulação em *software*, os circuitos digitais são gravados na FPGA, e esta é submetida aos mesmos sinais de aplicação final. As simulações com FIL permitiram validar os compensadores de desvio, sendo que os mesmos obtiveram o mesmo desempenho obtido dos modelos comportamentais.

O desempenho do sistema completo de compensação dos desvios do TIADC permitiu

ganhos significativos do SNR, SFDR e ENOB do TIADC. No exemplo apresentado na subseção 5.4, o SNR e o SFDR apresentaram um ganho de  $24,24\text{ dB}$  e  $41,7\text{ dB}$ , respectivamente. O ENOB, que neste mesmo exemplo era inicialmente de  $3,82\text{ bits}$ , alcançou o valor de  $7,85\text{ bits}$ . Estes resultados mais uma vez ilustram a vantagem da adição de sistemas digitais de compensação de desvios para TIADC. Outro aspecto que deve ser observado é utilização de recursos da FPGA para a implementação do sistema completo. A quantidade de LUTs e registradores utilizadas é muito pequena, de aproximadamente 3% do dispositivo, quando comparada com a capacidade da FPGA utilizada. Este resultado sugere que: os sistemas implementados em FPGA neste trabalho são viáveis de serem implementados em circuitos integrados.

O *layout* de um circuito integrado digital utilizando estas estruturas é sugerido como um potencial trabalho futuro. Deve-se observar que, as estruturas de estimação dos desvios de ganho e *clock skew* tratadas neste trabalho, foram implementadas somente em *software* por seus respectivos autores. Dados sobre a área que seria ocupada por estas estruturas e o seu consumo de potência em um circuito integrado são pontos que devem ser analisados futuramente.

Além do *layout* de um circuito integrado com os sistemas implementados, um estudo comparativo com outras abordagens para o projeto da estrutura de Farrow deve ser realizado futuramente. Estruturas propostas em (VESMA; SARAMAKI, 1996), (BABIC, 2005) e (JOHANSSON et al., 2006) são as mais recomendadas. Mais do que métodos para o cálculo dos coeficientes, estas estruturas são modificações da estrutura proposta originalmente por Farrow. Como pôde ser observado ao longo deste trabalho, um dos principais problemas enfrentados durante a implementação da estrutura de Farrow é o projeto destas estruturas utilizando aritmética de ponto fixo. A otimização do *wordlength* dos coeficientes dos filtros FIR da estrutura de Farrow através de diferentes algoritmos também deverá ser tratada em um trabalho futuro. O uso de algoritmos que substituem os multiplicadores da estrutura de Farrow com coeficientes fixos por somadores, como a *Multiple Constant Multiplication* (MCM) utilizada em (JOHANSSON et al., 2006), também deverá ser abordada.

Outras linhas de trabalho futuro seriam a realização de testes utilizando sinais de entrada com tons de múltiplas frequências, e o teste destas estruturas com um TIADC de dois canais físico. A extensão dos métodos aqui utilizados para TIADCs com mais do que dois canais também é sugerida.



## REFERÊNCIAS

- ANDERSON, D.; PADGETT, W. T.; MOURA, J. **Fixed-Point Signal Processing**. [S.l.]: Morgan and Claypool Publishers, 2008.
- BABIC, D. Reconstruction of Non-uniformly Sampled Signal after Time-Interleaved ADCS using Transposed Farrow Structure. In: **Computer as a Tool, 2005. EUROCON 2005. The International Conference on**. [S.l.: s.n.], 2005. v. 2, p. 1622–1625.
- BLACK JR., W. C.; HODGES, D. A. Time interleaved converter arrays. **IEEE J. Solid-State Circuits**, SC-15, n. 6, p. 1022–1029, December 1980. Correspondence.
- CONROY, C. S. G.; CLINE, D. W.; GRAY, P. R. An 8-b 85 MS/s parallel pipelined A/D converter in 1- $\mu$  m CMOS. **IEEE J. Solid-State Circuits**, v. 28, p. 447–454.
- DIAZ-CARMONA, J.; DOLECEK, G. J. Applications of MATLAB in Science and Engineering. In: \_\_\_\_\_. InTech, 2011. cap. Fractional Delay Digital Filters. ISBN 9789533077086. Disponível em: <<http://www.intechopen.com/books/applications-of-matlab-in-science-and-engineering/fractional-delay-digital-filters>>.
- DYER, K. et al. An analog background calibration technique for time-interleaved analog-to-digital converters. **Solid-State Circuits, IEEE Journal of**, v. 33, n. 12, p. 1912–1919, Dec 1998. ISSN 0018-9200.
- EKLUND, J.-E.; GUSTAFSSON, F. Digital offset compensation of time-interleaved ADC using random chopper sampling. In: **Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on**. [S.l.: s.n.], 2000. v. 3, p. 447–450 vol.3.
- EL-CHAMMAS, M.; MURMANN, B. A 12-GS/s 81-mW 5-bit Time-Interleaved Flash ADC With Background Timing Skew Calibration. **Solid-State Circuits, IEEE Journal of**, v. 46, n. 4, p. 838–847, April 2011. ISSN 0018-9200.
- ELBORNSSON, J.; GUSTAFSSON, F.; EKLUND, J.-E. Blind equalization of time errors in a time-interleaved ADC system. **Signal Processing, IEEE Transactions on**, v. 53, n. 4, p. 1413–1424, April 2005. ISSN 1053-587X.
- ERUP, L.; GARDNER, F. M.; HARRIS, R. Interpolation in digital modems. II. Implementation and performance. **Communications, IEEE Transactions on**, v. 41, n. 6, p. 998–1008, Jun 1993. ISSN 0090-6778.
- FARROW, C. W. A continuously variable digital delay element. In: - **Circuits and Systems, 1988., IEEE International Symposium on**. [S.l.: s.n.], 1988.
- FERRÉ, G. et al. Orthogonal Correction Implementation for Time-Interleaved Analog-To-Digital Converters: Realtime application. In: **In Proceedings of the 16th EURASIP European Conference on Signal Processing, August 2008.G**. [S.l.: s.n.], 2008.

FU, D. et al. A digital background calibration technique for time-interleaved analog-to-digital converters. **Solid-State Circuits, IEEE Journal of**, v. 33, n. 12, p. 1904–1911, Dec 1998. ISSN 0018-9200.

HUANG, C.-C.; WANG, C.-Y.; WU, J.-T. A CMOS 6-Bit 16-GS/s Time-Interleaved ADC Using Digital Background Calibration Techniques. **Solid-State Circuits, IEEE Journal of**, v. 46, n. 4, p. 848–858, April 2011. ISSN 0018-9200.

JAMAL, S. et al. A 10b 120MSample/s time-interleaved analog-to-digital converter with digital background calibration. In: **Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International**. [S.l.: s.n.], 2002. v. 1, p. 172–457 vol.1.

JAMAL, S. M. et al. Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter. **IEEE Trans. on Circuits and Systems**, v. 51-I, n. 1, p. 130–139, 2004.

JOHANSSON, H. et al. Adjustable Fractional-Delay FIR Filters Using the Farrow Structure and Multirate Techniques. In: **Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on**. [S.l.: s.n.], 2006. p. 1055–1058.

JOHANSSON, H.; LOWENBORG, P. Reconstruction of nonuniformly sampled bandlimited signals by means of digital fractional delay filters. **Signal Processing, IEEE Transactions on**, v. 50, n. 11, p. 2757–2767, Nov 2002. ISSN 1053-587X.

JOHANSSON, H.; LOWENBORG, P. On the design of adjustable fractional delay FIR filters. **Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on**, v. 50, n. 4, p. 164–169, Apr 2003. ISSN 1057-7130.

JOHANSSON, H.; LOWENBORG, P.; VENGATTARAMANE, K. Reconstruction of two-periodic nonuniformly sampled signals using polynomial impulse response time-varying fir filters. In: **Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on**. [S.l.: s.n.], 2006. p. 4 pp.–2996.

KARIMI, S. et al. A Design Methodology for Power Electronics Digital Control based on an FPGA in the Loop Prototyping. In: **Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on**. [S.l.: s.n.], 2007. p. 701–704.

KESTER, W. **Data conversion handbook**. Elsevier : Newnes, 2005. Disponível em: <[http://www.analog.com/library/analogdialogue/archives/39-06/data\\_conversion\\_handbook.html](http://www.analog.com/library/analogdialogue/archives/39-06/data_conversion_handbook.html)>.

KUROSAWA, N. et al. Explicit Analysis of Channel Mismatch Effects in Time-Interleaved ADC Systems. **Fundamental Theory and Applications, IEEE Transactions on Circuits and Systems I**, v. 48, n. 3, March 2001.

LAAKSO, T. I. et al. Splitting the unit delay - tools for fractional delay filter design. **IEEE Signal Process. Mag.**, v. 13, n. 1, p. 30–60, January 1996.

LOONEY, M. Advanced digital post-processing techniques enhance performance in time-interleaved ADC systems. **Analog Dialogue**, v. 37, n. 8, 2003.

MARIANO, A. A.

**Mixed Simulations and Design of a Wideband Continuous-Time Bandpass Delta-Sigma Converter Dedicated to Software Defined Radio Applications** — L'Université Bordeaux 1, France, 2008.

MATLAB. **version 8.4.0 (R2014b)**. Natick, Massachusetts: The MathWorks Inc., 2014.

MATLAB; HDLCODER, T. **version 3.5 (R2014b)**. Natick, Massachusetts: The MathWorks Inc., 2014. Disponível em: <<http://www.mathworks.com/products/hdl-coder/>>.

MATLAB; HDLVERIFIER, T. **version 4.5 (R2014b)**. Natick, Massachusetts: The MathWorks Inc., 2014. Disponível em: <<http://www.mathworks.com/products/hdl-verifier/>>.

MATLAB; SIMULINK. **version 8.4.0 (R2014b)**. Natick, Massachusetts: The MathWorks Inc., 2014.

MITOLA, J. **Cognitive radio, an integrated agent architecture for software defined radio**. Tese (Doutorado) — Royal Inst. of Technology, Stockholm, Sweden, 2000.

OETKEN, G. A new approach for the design of digital interpolating filters. **Acoustics, Speech and Signal Processing, IEEE Transactions on**, v. 27, n. 6, p. 637–643, Dec 1979. ISSN 0096-3518.

OPPENHEIM, A. V.; WILLSKY, A. S.; NAWAB, S. H. **Signals and Systems (2Nd Ed.)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN 0 13 814757 4.

PARKEY, C.; MIKHAEL, W. Time interleaved analog to digital converters: Tutorial 44. **Instrumentation Measurement Magazine, IEEE**, v. 16, n. 6, p. 42–51, December 2013. ISSN 1094-6969.

PEDRONI, V. **Circuit Design with VHDL**. [S.l.]: MIT Press, 2010. ISBN 9780262014335.

POULTON, K. et al. A 20 GS/s 8 b ADC with a 1 MB memory in 0.18 $\mu$ m CMOS. In: **Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. 2003 IEEE International**. [S.l.: s.n.], 2003. p. 318–496 vol.1. ISSN 0193-6530.

PROAKIS, J. G.; MANOLAKIS, D. G. **Digital Signal Processing (3rd Ed.): Principles, Algorithms, and Applications**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN 0-13-373762-4.

Quartus II Web Edition. **version 13.1**. San Jose, Califórnia: Altera Inc., 2013.

RAZAVI, B. Design Considerations for Interleaved ADCs. **IEEE Journal of Solid-State Circuits**, v. 48, n. 8, p. 1806–1817, August 2013.

RIVET, F. et al. A Mixed-Signal Built-In Self-Calibrated Time-Interleaved ADC in 65nm CMOS Technology. **Solid-State Circuits, IEEE Journal of**, v. 45, n. 5, p. 979–988, 2010.

ROERMUND, A. van et al. Smart AD and DA convertes. **IEEE International Symposium on Circuits and Systems**, p. 4062–4065, May 2005.

SEO, M.; RODWELL, M. J.; MADHOW, U. Comprehensive digital correction of mismatch errors for a 400-msamples/s 80-dB SFDR time-interleaved analog-to-digital converter. **Microwave Theory and Techniques, IEEE Transactions on**, v. 53, n. 3, p. 1072–1082, March 2005. ISSN 0018-9480.

STEPANOVIC, D. **Calibration Techniques for Time-Interleaved SAR A/D Converters**. Tese (Doutorado) — EECS Department, University of California, Berkeley, Dec 2012. Disponível em: <<http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-225.html>>.

SUMANEN, L.; WALTARI, M.; A.I.HALONEN, K. A 10-bit 200-MS/s CMOS parallel pipeline A/D converter. **Solid-State Circuits, IEEE Journal of**, v. 36, n. 7, p. 1048–1055, 2001.

VÄLIMÄKI, V. A new filter implementation strategy for Lagrange interpolation. In: **Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on**. [S.l.: s.n.], 1995. v. 1, p. 361–364 vol.1.

VÄLIMÄKI, V. **Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters**. [S.l.]: Helsinki University of Technology, Espoo, Finland, 1995.

VESMA, J. **Optimization and Applications of Polynomial-Based Interpolation Filters**. Tese (Doutorado) — Tampere University of Technology, Finland, 1999.

VESMA, J.; SARAMAKI, T. Optimization and efficient implementation of fractional delay FIR filters. In: **Electronics, Circuits, and Systems, 1996. ICECS '96., Proceedings of the Third IEEE International Conference on**. [S.l.: s.n.], 1996. v. 1, p. 546–549 vol.1.

VOGEL, C. **Modeling, identification and compensation of channel mismatch errors in time interleaved analog-to-digital converters**. Tese (Doutorado) — Graz University of Technology, Austria, July 2005.

VOGEL, C.; JOHANSSON, H. Time-Interleaved Analog-To-Digital Converters:Status and Future Directions. In: **Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on**. [S.l.: s.n.], 2006.

ZHAO, L. et al. A 1.6-Gsps High-Resolution Waveform Digitizer Based on a Time-Interleaved Technique. **Nuclear Science, IEEE Transactions on**, v. 60, n. 3, p. 2180–2187, June 2013. ISSN 0018-9499.

## APÊNDICE A – DEDUÇÃO MATEMÁTICA DA IMAGEM NA SAÍDA DO TIADC

Aqui é apresentada a dedução matemática, adaptada de (JAMAL et al., 2004), da imagem na saída de um TIADC devido ao desvio de *clock skew*. Considere o TIADC de dois canais em que o canal 1 realiza a amostragem do sinal de entrada após um intervalo de tempo  $T_{TIADC} + \Delta t$  do canal 2. Se o canal 2 realiza a amostragem nos instantes  $2nT_{TIADC}$  e o canal 1 nos instantes  $2nT_{TIADC} + T_{TIADC} + \Delta t$ , pode-se deduzir que o TIADC realiza a amostragem do sinal de entrada nos instantes  $nT_{TIADC} + \Delta t/2 - (-1)^n \Delta t/2$ . A saída  $y(n)$  do TIADC para um sinal de entrada  $x(t) = \cos(2\pi f_0 t + \phi)$  será portanto:

$$y(n) = \cos(2\pi f_0 t + \phi) \Big|_{t=nT_{TIADC} + \Delta t/2 - (-1)^n \Delta t/2} = \cos(2\pi f_0 [nT_{TIADC} + \Delta t/2 - (-1)^n \Delta t/2] + \phi). \quad (70)$$

Utilizando a relação  $\cos(a - b) = \cos(a)\cos(b) + \sin(a)\sin(b)$  na equação (70) é obtido:

$$y(n) = \cos(2\pi f_0 (nT_{TIADC} + \Delta t/2) + \phi) \cos((-1)^n 2\pi f_0 \Delta t/2) + \sin(2\pi f_0 (nT_{TIADC} + \Delta t/2) + \phi) \sin((-1)^n 2\pi f_0 \Delta t/2). \quad (71)$$

Considerando que a função cosseno é par, que a função seno é ímpar e que  $(-1)^n = \cos(n\pi)$ , é possível reescrever a equação (71) como:

$$y(n) = \cos(2\pi f_0 (nT_{TIADC} + \Delta t/2) + \phi) \cos(\pi f_0 \Delta t) + \sin(2\pi f_0 (nT_{TIADC} + \Delta t/2) + \phi) \cos(n\pi) \sin(\pi \Delta t). \quad (72)$$

E considerando que  $\sin(a)\cos(n\pi) = \sin(a - n\pi)$  e que  $n\pi = n\pi F_{TIADC} T_{TIADC}$  na equação (72), pode-se deduzir então a equação (73).

$$y(n) = \cos(\pi f_0 \Delta t) \cos(2\pi f_0 T_{TIADC} + \pi f_0 \Delta t + \phi) + \sin(\pi f_0 \Delta t) \sin((2\pi f_0 - \pi F_{TIADC}) n T_{TIADC} + \pi f_0 \Delta t + \phi) \quad (73)$$