



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

Tese de Doutorado

Conversor Configurável Analógico para Informação

Vanderson de Lima Reis

Campina Grande – PB, Paraíba, Brasil
©Vanderson de L. Reis, Janeiro de 2017



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
Centro de Engenharia Elétrica e Informática
Programa de Pós-Graduação em Engenharia Elétrica

Conversor Configurável Analógico para Informação

Vanderson de Lima Reis

Tese de Doutorado submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências, no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação.

Prof. Raimundo Carlos Silvério Freire, D.Sc.
Prof. Edmar Candeia Gurjão, D.Sc.
Orientadores

Campina Grande – PB
Janeiro - 2017

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

R375c

Reis, Vanderson de Lima.

Conversor configurável analógico para informação / Vanderson de Lima Reis.– Campina Grande, 2017.

84 f. il. color.

Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2017.

"Orientação: Prof. Dr. Raimundo Carlos Silvério Freire, Prof. Dr. Edmar Candeia Gurjão".

Referências.

1. Amostragem Compressiva. 2. Conversor Analógico para Informação. 3. Instrumentação. 4. Sinais Esparsos. 5. Simulação de Sistemas Dinâmicos. I. Freire, Raimundo Carlos Silvério. II. Gurjão, Edmar Candeia. III. Título.

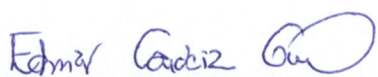
CDU 621.3(043)

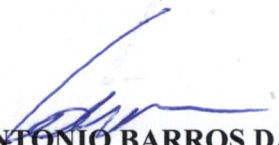
"CONVERSOR CONFIGURÁVEL DE ANALÓGICO PARA INFORMAÇÃO"

VANDERSON DE LIMA REIS

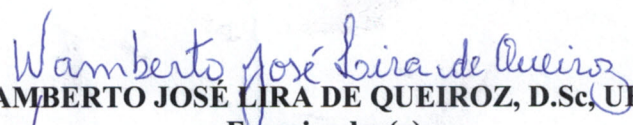
TESE APROVADA EM 20/04/2017

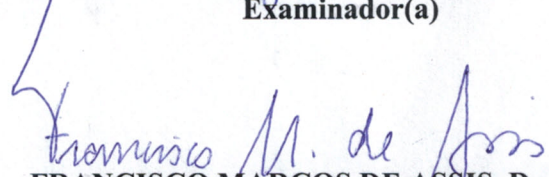

RAIMUNDO CARLOS SILVÉRIO FREIRE, Dr., UFCG
Orientador(a)


EDMAR CANDEIA GURJAO, D.Sc., UFCG
Orientador(a)


EDUARDO ANTONIO BARROS DA SILVA, Ph.D., COPPE/UFRJ
Examinador(a)


SEBASTIAN YURI CAVALCANTI CATUNDA, D.Sc., UFRN
Examinador(a)


WAMBERTO JOSÉ LIRA DE QUEIROZ, D.Sc., UFCG
Examinador(a)


FRANCISCO MARCOS DE ASSIS, Dr., UFCG
Examinador(a)

CAMPINA GRANDE - PB

DEDICATÓRIA

Ao Senhor Jesus Cristo, por ser o Autor e Consumador da minha fé;

À minha filha Rebeca Freitas Reis, por ser como eu sou;

Ao meu filho Matteo Freitas Reis, por ser como eu queria ser;

À minha saudosa avó Eliza Nunes de Santiago Lima, pela sua companhia na minha infância;

Ao saudoso mestre Gino Pierre, por ter me mostrado a beleza da eletrônica;

AGRADECIMENTOS

Ao meu orientador, D.Sc. Raimundo Carlos Silvério Freire pelo acompanhamento constante e pelos seus conselhos;

Ao meu orientador, D.Sc. Edmar Candeia Gurjão por ser o mentor principal do tema deste trabalho e pelas suas ajudas que transcenderam suas atribuições como orientador;

À minha família que abdicou do direito da minha presença;

Ao Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – IFAM;

À Fundação de Amparo à Pesquisa do Estado do Amazonas – FAPEAM;

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES;

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq;

A todos que de alguma forma me ajudaram durante esse doutorado (foram muitos);

“The phenomenon of ubiquitous compressibility raises very natural questions: why go to so much effort to acquire all the data when most of what we get will be thrown away? Can we not just directly measure the part that will not end up being thrown away?”

(Donoho [1])

RESUMO

Nos conversores Analógicos Digitais (ADC) com frequência de conversão baseada no Teorema de Nyquist, o parâmetro básico para orientar a aquisição é a largura de banda do sinal. O tratamento da informação e a remoção da redundância são realizados após a representação digital obtida do sinal. A Amostragem Compressiva foi proposta como uma técnica de digitalização que explora a esparsidade do sinal em um determinado domínio, para capturar apenas seu conteúdo de informação, com uma taxa que pode ser menor do que a preconizada pelo Teorema de Nyquist. As arquiteturas em *hardware* para implementar a Amostragem Compressiva são chamadas de Conversores Analógicos para Informação (AIC). Os AIC propostos na bibliografia exploram a esparsidade do sinal em um determinado domínio, e por isso cada arquitetura é específica para uma classe de sinais. Nesta tese propõe-se um AIC configurável, baseado em arquiteturas conhecidas, capaz de adquirir sinais de várias classes, alterando seus parâmetros de configuração. No trabalho desenvolveu-se um modelo computacional, que permite analisar o comportamento dinâmico do AIC, e dos parâmetros de *hardware* propostos, bem como foi feita a implementação física da arquitetura proposta. Verificou-se a adaptabilidade dessa arquitetura a partir dos resultados obtidos, pois foi possível fazer a aquisição de mais de uma classe de sinais.

Palavras-chave: Amostragem Compressiva, Conversor Analógico para Informação, Instrumentação, Sinais Esparsos, Simulação de Sistemas Dinâmicos.

ABSTRACT

In analog-to-digital converters (ADC) based on Nyquist Theorem, the basic parameter to guide acquisition is the bandwidth of the signal. The information processing and redundancy removal are performed after the digital representation obtained from the signal. Compressed Sensing was proposed as a digitalization technique that exploits the sparsity of the signal in a given domain to capture only its information content, at a rate that may be lower than that advocated by the Nyquist Theorem. The hardware architectures to implement Compressed Sensing are called Analog to Information Converters (AIC). The AICs proposed in the bibliography exploit the sparsity of the signal in a given domain, and therefore each architecture is specific for a class of signals. This thesis proposes a configurable AIC, based on known architectures, capable of acquiring signals from several classes, changing its configuration parameters. A computational model was developed to analyze the dynamic behavior of AIC and proposed hardware parameters, as well as the physical implementation of the proposed architecture. It was verified the adaptability of the proposed architecture from the obtained results, since it was possible to perform the acquisition of more than one class of signals.

Key-words: Compressed Sensing, Analog to Information Converter, Instrumentation, Sparse Signals, Dynamic Signal Simulation.

LISTA DE FIGURAS

Figura 2.1: Componentes básicos dos Sistemas de Controle e Aquisição de Dados [4].....	9
Figura 2.2: Sistemas de compactação com e sem perdas	11
Figura 2.3: Bola unitária em \mathbb{R}^2 , para algumas normas l_p [9].....	13
Figura 2.4: Operação de medida em CS (compactação) e tamanho dos vetores	15
Figura 2.5: Operação de medida em CS no domínio não esparsa do sinal.	15
Figura 2.6: Matriz de medida e de transformação do domínio esparsa do sinal \mathbf{x}	15
Figura 2.7: Linhas e colunas nas matrizes Φ e Ψ , respectivamente	16
Figura 2.8: Comportamentos possíveis em uma transformação linear.....	17
Figura 2.9: A métrica distância e preservada do domínio das medidas.	18
Figura 2.10: Reconstrução por l_2 , usando CVX.	20
Figura 2.11: Visualização geométrica da solução do sistema pela norma l_2	20
Figura 2.12: Bola unitária em \mathbb{R}^3 , para algumas normas l_p	22
Figura 2.13: Geometria da bola da norma l_1 com o hiperplano do sistema $\mathbf{y} = \Theta\mathbf{s}$	22
Figura 2.14: Reconstrução por l_1 , usando CVX.	23
Figura 2.15: Modelo dos ruídos do canal e do sistema	24
Figura 2.16: Diagrama em blocos do Demodulador Aleatório – RD[16].	26
Figura 2.17: Comportamento dos sinais nas etapas do RD [17]	26
Figura 2.18: Diagrama de bloco do MWC [21].....	27
Figura 2.19: A combinação linear em banda base de 2 canais do MWC [22]	28
Figura 2.20: Diagrama em blocos do NUS.	28
Figura 2.21: Diagrama em bloco do RMPI [38].....	29
Figura 2.22: Diagrama de blocos do Xampling.....	30
Figura 2.23: Xampling - Uma estrutura para processamento da união de subespaço[44]	30
Figura 3.1: Ilustração do Cenário de Ensaio do AIC.....	32
Figura 3.2: Imagem do Cenário de Ensaio do AIC	34
Figura 3.3: LFSR Arquitetura de Galois (ou <i>One-to-Many</i>) para o Polinômio $1 + x + x^4 + x^5 + x^6$	37
Figura 3.4: LFSR Arquitetura de Fibonacci (ou <i>Many-to-One</i>) para o Polinômio $1 + x + x^4 + x^5 + x^6$	37
Figura 3.5: LFSR Programável (<i>taps</i> variáveis) para Arquitetura de Fibonacci	38
Figura 3.6: Bloco do LFSR Programável (Configurável), descrito em FPGA.....	38
Figura 3.7: Bloco do Circuito do Sincronismo do <i>Reset</i>	39

Figura 3.8: Bloco do Circuito de sincronismo do DAQ	40
Figura 3.9: AIC <i>Hardware</i> Analógico, Vista de Superior (a esquerda) e Vista em Perspectiva (a direita).....	41
Figura 3.10: Circuito Divisor Resistivo com <i>Offset</i>	41
Figura 3.11: Divisor Resistivo com <i>Offset</i> , dos 8 canais: Esquemático (a esquerda) e Imagem (a direita).....	42
Figura 3.12: Multiplicador de 4 Quadrante AD633: Esquemático (a esquerda) Imagem na PCI (a direita).....	43
Figura 3.13: Resposta em Frequência Assintótica: Integrador Ideal(em Azul) e Integrador em <i>Hardware</i> (em vermelho)	44
Figura 3.14: Filtro Integrador com Circuito de <i>Reset</i>	44
Figura 3.15: <i>Card</i> de configuração da Capacitância C : Leiaute (a esquerda) e Imagem (a direita).....	45
Figura 3.16: Fluxograma para Reconstrução usando o CVX de Medidas Feitas no Domínio Não Esperso.....	48
Figura 3.17: Fluxograma para Reconstrução usando o CVX e o OMP de Medidas Feitas no Domínio Esperso	49
Figura 3.18: Comparativo dos sinais x_t , $\hat{x}[n]$ e $\tilde{x}[n]$, sem ajustes de ganho e fase.....	50
Figura 3.19: Construção do sinal modelo $\tilde{x}[n]$	50
Figura 3.20: Dados de Entrada e Saída da Função Busca Ganho.....	51
Figura 3.21: Dados de Entrada e Saída da Função busca Δt com um erro ε	51
Figura 3.22: Construção do Novo Sinal Modelo \tilde{x}_n	51
Figura 3.23: Sinal Modelo: a) $\tilde{x}[n]$ sem ajuste de fase b) $\tilde{x}[n]$ com ajuste de fase	52
Figura 3.24: Novos Dados de Entrada e Saída da Função Busca Ganho	52
Figura 3.25: Comparativo dos Sinais Antes e Depois do Ajuste de Fase (Δt).....	53
Figura 3.26: LFSR Configurável, no Modelo Simulink.....	54
Figura 3.27: Condicionamento dos Sinais do LFSR, no Modelo Simulink	55
Figura 3.28: Multiplicador no Simulink, Equivalente ao CI AD633	56
Figura 3.29: Filtro Modelado no Simulink.....	56
Figura 3.30: ADC (DAQ) Modelado no Simulink.....	57
Figura 4.1: Sinais Internos do AIC em <i>Software</i> (Simulado) para 1 tom:	59
Figura 4.2: Reconstrução com CVX das Medidas do AIC em <i>Software</i> (Simulado) para 1 tom:	60

Figura 4.3: Sinais Internos do AIC em <i>Hardware</i> para 1 tom :	61
Figura 4.4: Reconstrução com CVX das Medidas do AIC em <i>Hardware</i> para 1 tom:	62
Figura 4.5: Reconstrução com CVX usando ε , das Medidas do AIC em <i>Hardware</i> para 1 tom:	62
Figura 4.6: Reconstrução com CVX de 10 conjuntos de Medidas com AIC em <i>Hardware</i> para 1 tom:	63
Figura 4.7: Sinais Internos do AIC em <i>Software</i> (Simulado) para 2 tons:	64
Figura 4.8: Reconstrução com CVX das Medidas do AIC em <i>Software</i> (Simulado) para 2 tons:	65
Figura 4.9: Sinais Internos do AIC em <i>Hardware</i> para 2 tons :	65
Figura 4.10: Reconstrução com CVX das Medidas do AIC em <i>Hardware</i> para 2 tons:	66
Figura 4.11: Reconstrução com CVX usando ε , das Medidas do AIC em <i>Hardware</i> para 2 tons:	66
Figura 4.12: Reconstrução com CVX de 10 conjuntos de Medidas com AIC em <i>Hardware</i> para 2 tons:	67
Figura 4.13: Sinais Internos do AIC em <i>Software</i> (Simulado) para 3 tons:	68
Figura 4.14: Reconstrução com CVX das Medidas do AIC em <i>Software</i> (Simulado) para 3 tons:	69
Figura 4.15: Sinais Internos do AIC em <i>Hardware</i> para 3 tons :	69
Figura 4.16: Reconstrução com CVX das Medidas do AIC em <i>Hardware</i> para 3 tons:	70
Figura 4.17: Reconstrução com CVX usando ε , das Medidas do AIC em <i>Hardware</i> para 3 tons:	70
Figura 4.18: Reconstrução com CVX de 10 conjuntos de Medidas com AIC em <i>Hardware</i> para 3 tons:	71
Figura 4.19: Sinais Internos do AIC em <i>Software</i> (Simulado) para o sinal esparso xt	73
Figura 4.20: Reconstrução com CVX e OMP das Medidas do AIC em <i>Software</i> (Simulado) para o sinal st	74
Figura 4.21: Sinais Internos do AIC em <i>Hardware</i> para st :	75
Figura 4.22: Reconstrução com CVX e OMP das Medidas do AIC em <i>Hardware</i> para o sinal st	75
Figura 4.23: Reconstrução com CVX (com ε) e OMP das Medidas do AIC em <i>Hardware</i> para o sinal st	76

Figura 4.24: Repetitividade do AIC em <i>Hardware</i> para o sinal $s(t)$, para 10 conjuntos de	
Medidas:	77

LISTA DE TABELAS

Quadro 3.1: Expressões e Exemplos dos números das sequências dos LFSR com XOR.....	36
Quadro 3.2: Parâmetros e Exemplos de construção de um LFSR.....	37
Quadro 4.1: Configurações Gerais para Realização de Medidas no AIC.	58
Quadro 4.2: Comparativos das 3 Reconstruções para uma Senóide (1 tom).	63
Quadro 4.3: Comparativos das 3 Reconstruções para 2 tons.	67
Quadro 4.4: Comparativos das 3 Reconstruções para 3 tons.	71
Quadro 4.5: Configurações Realizadas nas Medidas no AIC em <i>Software e Hardware</i>	72
Quadro 4.6: Comparativos das 3 Reconstruções para o sinal <i>st</i>	76

Lista de Abreviaturas e Siglas

A/D	Analógico-Digital
A2D	Analog-to-Digital
A2I	Analog-to-Information
ADC	Analog-to-digital converter
AHDL	Altera Hardware Description Language
AIC	Analog-to-Information Converter
ASIC	Application-Specific Integrated Circuit
AWG	Arbitrary Waveform Generator
CD	Compact Disc
CI	Circuito Integrado
COTS	Commercial Off-The-Shelf
CS	Compressed Sensing
D/A	Digital-Analógico
D2A	Digital-to-analog
DAC	Digital-to-analog converter
DAQ	Data Acquisition
dB	Decibel
DC	Direct Current
DFT	Discrete Fourier Transform
DNL	Differential Nonlinearity
DVD	Digital Versatile Disc
ENOB	Effective Number of Bits

FFD	Flip-Flops tipo D
FPGA	Field-Programmable Gate Array
GF	Galois Field
GND	Ground
HDL	Hardware Description Language
HW	Hardware
IDFT	Inverse Discrete Fourier Transform
IEEE	Institute of Electrical and Electronics Engineers
INL	Integral Nonlinearity
LAN	Local Area Network
LFSR	Linear Feedback Shift Register
LTNUS	Level-Triggered Nonuniform Sampling
MSE	Mean Squared Error
MWC	Modulated Wideband Converter
NUS	Non-Uniform Sampler
OMP	Ortogonal Matching Pursuit
PC	Personal computer
PCI	Placa de Circuito Impresso
PLD	Programmable Logic Device
PLL	Phase Locked Loop
PNUS	Periodic Non-Uniform Sampling
PRBS	Pseudo Random Binary Sequence
PRNG	Pseudo Random Number Generator

PSNR	Peak Signal-to-Noise Ratio
PSPICE	Simulation Program with Integrated Circuit Emphasis for PC
RD	Randon Demodulation
RF	Radiofrequência
RIP	Propriedade da Isometria Restrita
RMPI	Random Modulation Pre-Integrator
RNUS	Randomized Nonuniform Sampling
S/H	Sample-and-Hold
SADC	Sistemas de Aquisição de Dados Clássicos
SFDR	Spurious Free Dynamic Range
SINAD	Signal-to-Noise-and-Distortion Ratio
SMD	Surface Mount Device
SNR	Signal-to-Noise Ratio
SPICE	Simulation Program with Integrated Circuit Emphasis
SR	Shift Register
SW	Software
THD	Total Harmonic Distortion
USB	Universal Serial Bus
VHDL	Very High Speed Integrated Circuit HDL
X-ADC	Xampling ADC
X-DSP	Xampling DSP
XOR	Exclusive Or
ZOH	Zero-Order Hold

Lista de Símbolos e Notação

$\mathbb{A}, \mathbb{N}, \mathbb{R}, \mathbb{C}, \dots$	conjuntos numéricos
\mathbb{R}^N	conjunto de vetores reais de dimensão N
$\mathbb{R}^{M \times N}$	conjunto de matrizes reais de ordem (ou dimensão) $M \times N$
A, B, Φ, Ψ, \dots	matrizes
$\mathbf{x}, \mathbf{y}, \mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\psi} \dots$	vetores
$a, A, b, B, \alpha, \beta, \lambda, \dots$	escalares
\mathbf{a}_j	j -ésima coluna da matriz A
$[\mathbf{a}_i]^T$	i -ésima linha da matriz A
a_{ij}	elemento da i -ésima linha e j -ésima coluna da matriz A
x_i	i -ésimo elemento do vetor \mathbf{x}
$A_{M \times N}$	matriz A de ordem $M \times N$ (de M linhas e N colunas)
$\#\mathbb{A}$	cardinalidade de \mathbb{A} , i.e. o número de elementos do conjunto \mathbb{A}
A^T	transposta da matriz A
$ \cdot $	determinante para uma matriz, ou valor absoluto para um escalar
$\text{supp}\{\mathbf{x}\}$	suporte de \mathbf{x} definido com $\{i : x_i \neq 0\}$
$\ \mathbf{x}\ _p$ ou $\ \mathbf{x}\ _{l_p}$	norma l_p ($p \geq 1$) ou pseudo-norma ($0 < p < 1$) de \mathbf{x}
$\langle \mathbf{x}, \mathbf{y} \rangle$	produto interno (ou escalar) dos vetores \mathbf{x} e \mathbf{y}
A^{-1}	inversa da matriz A
A^+	pseudo-inversa da matriz A

Sumário

1	INTRODUÇÃO	6
1.1	Motivação e Relevância	7
1.2	Objetivos	7
1.2.1	Objetivo Geral	7
1.2.2	Objetivos Específicos	7
1.3	Organização do Documento	8
2	FUNDAMENTAÇÃO TEÓRICA	9
2.1	Sistemas de Aquisição de Dados	9
2.2	Amostragem Compressiva	10
2.2.1	Normas	11
2.2.2	Esparsidade	13
2.2.3	Medidas	14
2.2.4	Incoerência	16
2.2.5	Critérios da Matriz de Medida	16
2.3	Regularização	18
2.3.1	Reconstrução por l_2	19
2.3.2	Reconstrução Teórica - l_0	20
2.4	Convexidade	21
2.4.1	Reconstrução Prática - l_1	22
2.4.2	Reconstrução Prática - l_1 com ruído	23
2.5	Arquiteturas dos AIC	25
2.5.1	Demodulador Aleatório	26
2.5.2	Conversor de Banda Larga Modulada	27
2.5.3	Amostrador Não Uniforme	28

2.5.4	Pré-Integrador de Modulação Aleatória	29
2.5.5	Xampling	30
3	DESENVOLVIMENTO	31
3.1	Visão geral	31
3.2	AIC em <i>Hardware</i>	34
3.2.1	AIC <i>Hardware</i> Digital	35
3.2.2	AIC <i>Hardware</i> Analógico	40
3.3	Reconstrução e Ajuste em <i>Software</i>	46
3.3.1	Reconstrução	46
3.3.2	Ajustes e Condicionamento de Sinal.	49
3.4	AIC em <i>Software</i>	53
4	RESULTADOS	58
4.1	Medidas no Domínio Não Esparso	58
4.1.1	Sinal com um tom	59
4.1.2	Sinal com dois tons	64
4.1.3	Sinal com três tons	68
4.2	Medidas no Domínio Esparso	72
5	CONCLUSÃO E TRABALHOS FUTUROS	78
5.1	Conclusão	78
5.2	Trabalhos Futuros	79
6	REFERÊNCIAS	81

1 INTRODUÇÃO

O ano de 2007 foi o primeiro ano em que a humanidade conseguiu gerar mais dados na sua história do que conseguia armazenar, considerando todo espaço utilizável em discos rígidos, fitas, CD, DVD e memórias (volátil e não volátil) no mercado, com cerca de 264 exabytes (1 exabyte = 10^{18} bytes). Nos anos seguintes essa diferença só foi aumentando (volume gerado e armazenado) [2]. Em 2006 esse valor já havia chegado a 180 exabytes, crescendo para 1600 exabytes em 2011, para 4,4 zettabytes (1 zettabyte = 10^{21} bytes) em 2013 e para 2020 é estimado 44 zettabytes [3].

Pode-se questionar o quanto de informação existe nos dados gerados, ou melhor, de forma complementar, o quanto de informação redundante existe? Nesse sentido, pesquisas têm sido realizadas para auxiliar a captura dos sinais de forma mais eficiente, de modo que apenas a informação contida no sinal seja obtida. Os Conversores Analógico-Digitais (ADC) clássicos são baseados no critério de Nyquist-Shannon e fundamentados na banda do sinal a ser digitalizado. Mesmo que nessa banda exista muita redundância de informação, todo sinal analógico é convertido em dados digitais, ficando para processamentos posteriores a possibilidade de serem usados eliminadores (ou redutores) de redundância, chamados de compactadores de dados. Como uma outra abordagem, foi proposta a Amostragem Compressiva (tradução do termo *Compressed Sensing*), que usa a representação esparsa (muitos zeros) de um sinal em um determinado domínio para realizar a sua digitalização de forma mais eficiente.

Os Conversores Analógico para Informação (AIC) fundamentados na teoria da Amostragem Compressiva buscam mudar esse modelo dos ADC clássicos, e essa quebra de paradigma é justificada pelo fato de ser possível reconstruir um sinal original sem respeitar a taxa de Nyquist, sob certas condições.

Neste trabalho é apresentado o desenvolvimento de um AIC mais flexível, que pode ser adaptado para diversos tipos de sinais esparsos.

1.1 Motivação e Relevância

Algumas arquiteturas dos AIC vêm sendo propostas, umas desenvolvidas em circuitos integrados (ASIC), outras com componentes comerciais e outras com partes em ASIC e partes com circuitos comerciais. De qualquer forma, muitas dessas arquiteturas são moldadas para sinais esparsos com certas características específicas. A partir de uma revisão bibliográfica, constatou-se que essas arquiteturas possuem uma estrutura básica semelhante. Baseando-se nesse fato, neste trabalho, propõe-se o desenvolvimento de uma arquitetura que seja configurável, podendo ser modificada de acordo com a classe de sinais de entrada.

Outra vantagem de uma arquitetura configurável é facilitar a proposição e avaliação de subsistemas necessários para o desenvolvimento de um AIC, como é mostrado na Seção 3.3, como também que possa ser utilizada em outras pesquisas sobre Amostragem Compressiva.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo desta tese de doutorado é propor e validar uma arquitetura de um conversor analógico para informação (AIC) que seja configurável, permitindo que o mesmo seja utilizado para medições de uma maior variedade de sinais esparsos. O AIC configurável, deve permitir que seu *hardware* possa ser adaptado às características do sinal que se deseja medir.

1.2.2 Objetivos Específicos

Desenvolvimento de uma plataforma de simulação no Ambiente Simulink do MATLAB, que seja capaz de estimar os parâmetros de configuração necessários a serem utilizados no *hardware*, para uma reconstrução bem sucedida do sinal medido.

Desenvolvimento de uma interface para que as configurações dos parâmetros do AIC sejam simples de realizar.

Desenvolvimento da parte digital do AIC em FPGA; e da parte analógica, em PCI (Placa de Circuito Impresso) com componentes comerciais.

Proposição de um modelo (cenário) de ensaio para avaliar o *hardware* desenvolvido.

1.3 Organização do Documento

Este texto é composto de quatro capítulos, a saber:

No Capítulo 1, Introdução, apresenta-se o trabalho de forma geral, fornecendo uma visão global do assunto tratado, a sua relevância, a definição dos problemas para serem solucionados, sua delimitação, os objetivos e a organização do trabalho.

No Capítulo 2, Fundamentação Teórica, são apresentados conceitos sobre sistemas de aquisição de dados clássicos, amostragem compressiva e o estado da arte das arquiteturas existentes dos AIC, essas definições são de fundamental importância para o entendimento do *hardware* desenvolvido.

No Capítulo 3, Desenvolvimento, é feita uma exposição geral da plataforma desenvolvida (o AIC) e o cenário de ensaio, bem como das simulações e ensaios realizadas e, por fim, uma descrição do *hardware* desenvolvido e das reconstruções realizadas.

No Capítulo 4 apresentam-se os Resultados e as Conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Sistemas de Aquisição de Dados

Os sistemas de controle e aquisição de dados clássicos têm como componente principal os conversores de sinais analógicos para sinais digitais, abreviados de ADC, A2D ou A/D. Contudo esses sistemas podem possuir outros componentes dependendo das características do sinal a ser digitalizado. Na Figura 2.1 pode ser visto um diagrama de blocos, com vários componentes desses sistemas.

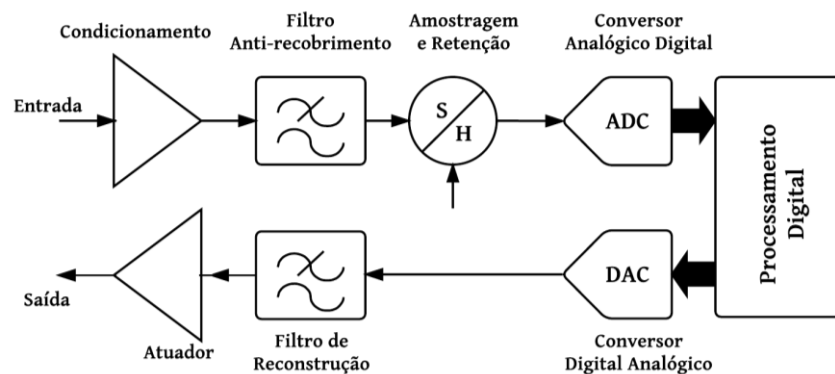


Figura 2.1: Componentes básicos dos Sistemas de Controle e Aquisição de Dados [4]

O circuito de condicionamento pode ser utilizado para ajustar (atenuar ou amplificar) a faixa de tensão (ou corrente) do sinal de entrada, filtrar sinais não desejados, deslocar nível, linearizar e adequar o sinal à faixa de entrada do ADC. O filtro anti-recobrimento (*anti-aliasing*) é usado para evitar que ruídos fora da faixa de frequências de interesse do sinal de entrada sejam transferidos para o sinal útil durante o processo de amostragem. O circuito de amostragem e retenção (*Sample-and-Hold, S/H*) é usado para amostrar o sinal de entrada e reter o valor amostrado durante o tempo que o conversor A/D estiver fazendo a conversão. O sinal uma vez digitalizado é então processado, seja por um microcomputador, um microcontrolador, uma FPGA ou qualquer outro tipo de processador digital, em seguida um processo no sentido contrário pode ser realizado. Um sinal digital é enviado para um conversor digital para analógico (DAC, D2A ou D/A), em seguida um filtro de reconstrução é utilizado para filtrar as altas frequências do sinal e por fim um circuito atuador (*buffer*) é usado para produzir uma saída que possa ter condições de atuar satisfatoriamente em uma determinada carga.

A escolha de um ADC pode ser feita, basicamente, pela escolha da resolução (número de bits) e taxa de amostragem, bem como pelos parâmetros estáticos e dinâmicos relacionados ao conversor, tais como:

- a) Estáticos: ganho, erro de ganho, *offset*, erro de *offset*, INL (*integral nonlinearity*), DNL (*differential nonlinearity*) e outros.
- b) Dinâmicos: SNR (*signal-to-noise ratio*), SINAD (*signal-to-noise-and-distortion ratio*), ENOB (*effective number of bits*), SFDR (*spurious free dynamic range*), THD (*total harmonic distortion*) e outros.

Várias arquiteturas têm sido propostas para a implementação de ADC, dentre elas pode-se destacar: Rampa Dupla, Aproximação Sucessiva, Sigma-Delta e Paralela [5]. É importante salientar que todas digitalizam o sinal tendo como parâmetro sua largura de banda, sem levar em conta a possível presença de redundância.

2.2 Amostragem Compressiva

O conhecimento prévio de algumas características do sinal a ser digitalizado é um requisito básico em sistemas de digitalização. Nos Sistemas de Aquisição de Dados Clássicos (SADC), conforme visto na Seção 2.1, a principal característica a ser conhecida é o valor da máxima frequência do sinal ou da largura de banda em sistemas passa-faixa. A taxa de amostragem do sinal deve então respeitar o critério de Nyquist, ou seja, deve ser no mínimo duas vezes a largura de banda do sinal amostrado, para isso, usam-se os filtros anti-recobrimento antes da amostragem para evitar que ruídos fora da faixa útil de frequências sejam amostrados junto com o sinal.

Na Amostragem Compressiva (do inglês, *Compressed Sensing* - CS) não é diferente, só que neste caso a principal característica do sinal a ser respeitada é que ele tenha uma representação com poucos valores não nulos em algum domínio, conhecida como esparsidade. A necessidade de respeitar essa característica pode ser considerada uma desvantagem para a aplicação da Amostragem Compressiva, contudo poucos sinais não conseguem ter uma determinada esparsidade quando representados em um domínio adequado. Por exemplo, representações não esparsas no domínio do tempo podem ser esparsas no domínio da frequência, da mesma forma que é difícil ter um sinal esparsos nos dois domínios ou não esparsos também nos dois domínios. É a dualidade tempo frequência. Contudo o importante é conhecer um determinado domínio em que esse sinal seja o mais esparsos possível.

A grande vantagem da Amostragem Compressiva é que ela permite digitalizar apenas a informação relevante do sinal a uma taxa de amostragem bem menor que a taxa de Nyquist, e assim, o sinal digitalizado já está em uma forma comprimida (codificada, compactada). Nos SADC, quando é usada compressão, ela é feita após a digitalização. Como pode ser visto na Figura 2.2, esse processo pode levar um sinal digital (imagem A_0 a esquerda) que é submetido a um codificador resultando em A_c , que possui um tamanho reduzido, com a quantidade de dados menor que A_0 . A decodificação de A_c pode resultar na imagem original (A_0) ou em uma versão da imagem original com algumas perdas (A_p), dependendo dos codificadores e decodificadores que foram utilizados.

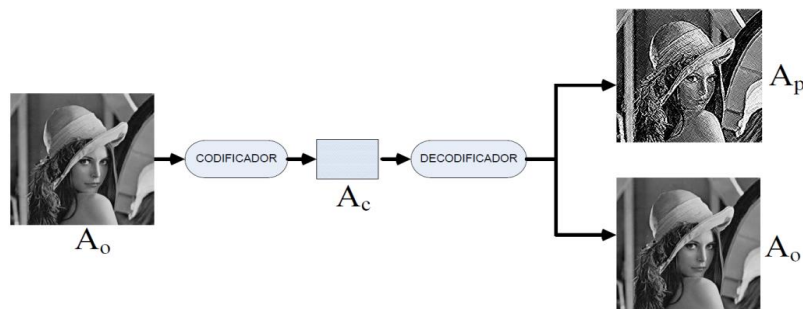


Figura 2.2: Sistemas de compactação com e sem perdas

Nos sistemas de aquisição de dados que utilizam a teoria da Amostragem Compressiva o sinal digitalizado já está na forma comprimida A_c , ficando para o processamento após digitalização apenas o que se pode chamar de decodificação (ou de reconstrução do sinal).

A Amostragem Compressiva é baseada nas teorias de Projeções Aleatórias, Álgebra Linear e Otimização. Nessa área se destacam os trabalhos precursores de Donoho [6], e Candès e Tao [7]. O termo Amostragem Compressiva pode ser encontrado na língua inglesa como *Compressed Sensing* (CS), ou *Compressive Sensing* ou *Compressed Sampling*.

Embora alguns autores descrevam que a CS viola o teorema Shannon-Nyquist [8], um melhor entendimento é que a CS se vale de informação adicional do sinal (esparsidade) para poder trabalhar com uma taxa de amostragem menor que a descrita pelo teorema Shannon-Nyquist. A seguir são apresentadas algumas definições que são de fundamental importância para o entendimento da Amostragem Compressiva.

2.2.1 Normas

Os sinais de interesse podem ser considerados como vetores em um espaço Euclidiano de N dimensões, \mathbb{R}^N , e para esses vetores define-se o conceito de norma.

Definição 2.1: Seja S um espaço vetorial com elementos \mathbf{x} . Uma função real $\|\mathbf{x}\|$ é dita uma norma se satisfizer as seguintes propriedades [9]:

P₁: $\|\mathbf{x}\| \geq 0$ para qualquer $\mathbf{x} \in S$.

P₂: $\|\mathbf{x}\| = 0$ se e somente se $\mathbf{x} = \mathbf{0}$.

P₃: $\|\alpha\mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$, sendo α um escalar.

P₄: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, chamada de desigualdade triangular.

Assim pode-se definir uma variedade de tipos de normas, tais como:

a) Norma l_1 : (é a soma do modulo dos valores do vetor \mathbf{x} , chamada de norma da soma).

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|. \quad (2.1)$$

b) Norma l_2 : (representa o tamanho do vetor, chamada de norma Euclidiana).

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2}. \quad (2.2)$$

c) Norma l_p : (é genericamente uma norma de ordem p , sendo l_1 e l_2 casos particulares para $p = 1$ e $p = 2$ respectivamente).

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}, \quad 0 < p \leq \infty. \quad (2.3)$$

d) Norma $l_{+\infty}$: (é o maior valor em módulo dos elementos do vetor \mathbf{x} , chamada de norma do máximo).

$$\|\mathbf{x}\|_{+\infty} = \max_{1 \leq i \leq n} |x_i|. \quad (2.4)$$

e) Norma $l_{-\infty}$: (é o menor valor em módulo dos elementos do vetor \mathbf{x} , chamada de norma do mínimo).

$$\|\mathbf{x}\|_{-\infty} = \min_{1 \leq i \leq n} |x_i|. \quad (2.5)$$

f) Norma l_0 : (é a quantidade de elementos não nulos de \mathbf{x} , ou seja, sua cardinalidade).

$$\|\mathbf{x}\|_0 = \#\{i: x_i \neq 0\}. \quad (2.6)$$

A rigor a norma l_0 é uma quase-norma, pois não satisfaz a propriedade P₄.

Essas normas quando calculadas para um determinado vetor \mathbf{x} , possuem como resultado a relação descrita na Equação (2.7). Essa relação mostra que essas normas são equivalentes em termos de limites, ou seja, quando no limite uma tende a zero então todas também vão tender.

$$\|\mathbf{x}\|_{+\infty} \leq \dots \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1. \quad (2.7)$$

Cada uma dessas normas introduz a sua própria geometria (topologia). Como exemplo, uma bola (*ball*) unitária no \mathbb{R}^2 , definida na Equação (2.8), pode ser visualizada na Figura 2.3 para algumas normas l_p .

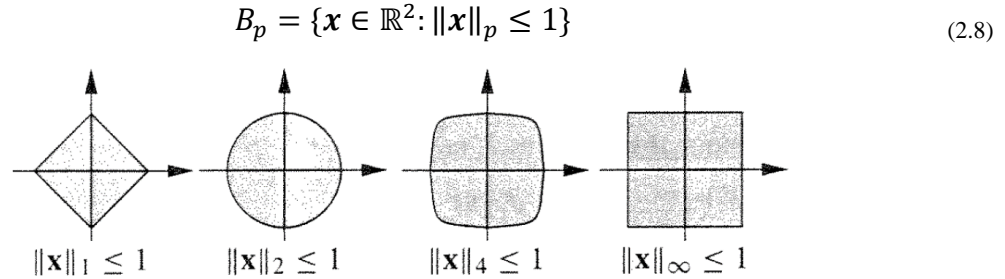


Figura 2.3: Bola unitária em \mathbb{R}^2 , para algumas normas l_p [9]

Existe um equivalente contínuo dessas funções. Seja $x(t)$ uma função contínuo sobre um determinado intervalo $[a, b]$, a norma genérica l_p de $x(t)$ pode ser escrita como na Equação (2.9), que é o caso contínuo do caso discreto visto na Equação (2.3).

$$\|x(t)\|_p = \left(\int_a^b |x(t)|^p dt \right)^{1/p}, \quad \text{para } 1 \leq p \leq \infty. \quad (2.9)$$

Outras notações podem ser usadas para identificar qual norma está sendo aplicada a um determinado vetor ou sinal, como exemplo: $\|\mathbf{x}\|_p = \|\mathbf{x}\|_{l_p}$.

2.2.2 Esparsidade

Definição 2.2: Seja \mathbf{x} um sinal de N elementos e com muitos desses elementos nulos, então se diz que esse sinal é esparso e a quantidade dos elementos não nulos desse sinal é a sua esparsidade.

A esparsidade pode ser calculada pela quase-norma l_0 , Equação (2.6).

É certo que a definição de esparsidade é relativa ao domínio que se encontra o sinal \mathbf{x} , ou seja, uma transformação linear de um sinal não esparso pode representar esse sinal em outro domínio em que esse sinal seja esparso.

Considere:

- C1: $\mathbf{x} \in \mathbb{R}^N$, um sinal em um domínio não esparso,
- C2: $\Psi \in \mathbb{R}^{N \times N}$, uma transformação linear que leva o sinal \mathbf{x} para um domínio esparso,
- C3: $\Psi^{-1} \in \mathbb{R}^{N \times N}$, a transformação inversa Ψ ,
- C4: $\mathbf{s} \in \mathbb{R}^N$, o sinal \mathbf{x} representado em um domínio esparso.

Dessa forma a Equação (2.10) representa a transformação linear (Ψ) de um sinal no domínio não esparso \mathbf{x} para um domínio esparso \mathbf{s} .

$$\mathbf{s} = \Psi \mathbf{x} \quad (2.10)$$

A transformação linear inversa (Ψ^{-1}) desse sinal é representada por

$$\mathbf{x} = \Psi^{-1} \mathbf{s}. \quad (2.11)$$

2.2.3 Medidas

Em Amostragem Compressiva uma medida é definida como a projeção de um vetor em um determinado espaço. São feitas M medidas (ou amostras) do sinal \mathbf{x} que tem N componentes, com M muito menor que N , assim sendo, considera-se também:

C5: $\mathbf{y} \in \mathbb{R}^M$, as medidas do sinal \mathbf{x} ,

C6: $\Phi \in \mathbb{R}^{M \times N}$, a matriz de medida do sinal \mathbf{x} no domínio não esparso,

C7: $\Theta \in \mathbb{R}^{M \times N}$, a matriz de medida do sinal \mathbf{s} (que é sinal \mathbf{x} no domínio esparso), têm-se

$$\mathbf{y} = \Phi \mathbf{x}, \quad (2.12)$$

e

$$\mathbf{y} = \Theta \mathbf{s} \quad (2.13)$$

o processo de medidas feitas em Amostragem Compressiva, seja com o sinal no domínio esparso ou não.

A matriz de medida (*measurement matrix*) é chamada também de matriz de sensoriamento (*sensing matrix*).

Das definições C1 a C7, verifica-se, pelas Equações (2.12) e (2.13), o processo de medidas feitas em Amostragem Compressiva, seja com o sinal no domínio esparso ou não.

Igualando-se as Equações (2.12) e (2.13), e observando-se as Equações (2.10) e (2.11) verifica-se as relações que essas matrizes devem possuir, ou seja,

$$\Theta = \Phi \Psi^{-1}, \quad (2.14)$$

$$\Phi = \Theta \Psi. \quad (2.15)$$

Analisando as dimensões desses vetores (matrizes), a Equação (2.13) pode ser representada na Figura 2.4, em que a ordem das matrizes e o efeito de compactação são destacados. Um dos requisitos necessários para a reconstrução do sinal é que a esparsidade de \mathbf{s} obedeça

$$K = \|\mathbf{s}\|_0 \ll M \ll N. \quad (2.16)$$

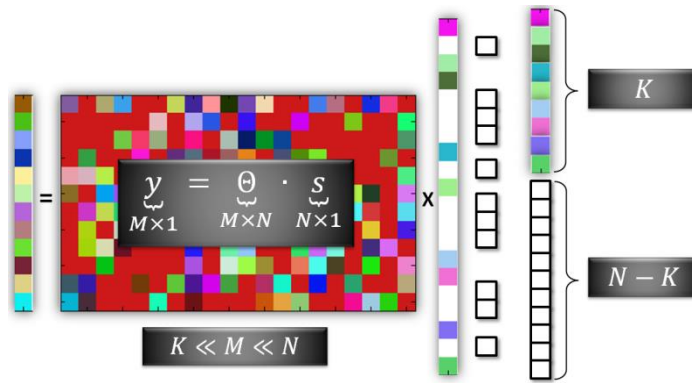


Figura 2.4: Operação de medida em CS (compactação) e tamanho dos vetores

É importante destacar que o sinal a ser medido em CS não necessita estar no domínio esparso, conforme observado na Equação (2.12) e ilustrado na Figura 2.5.

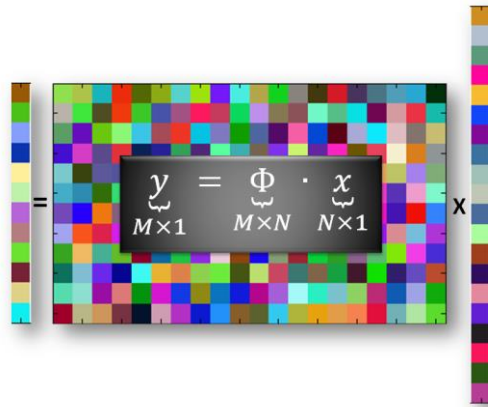


Figura 2.5: Operação de medida em CS no domínio não esparso do sinal.

Na Figura 2.6 pode ser visto uma equivalência da Figura 2.5 substituindo o sinal x pelo sinal s . Contudo, alguns requisitos são necessários para a matriz de medida Φ e a matriz de transformação Ψ , pois essas matrizes devem ser incoerentes.

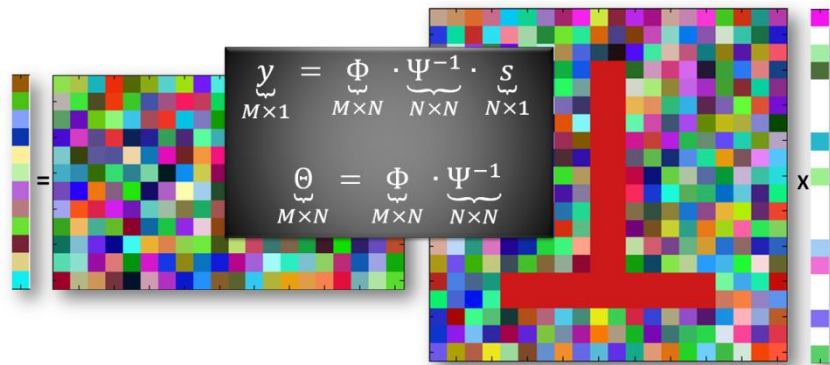


Figura 2.6: Matriz de medida e de transformação do domínio esparso do sinal x .

2.2.4 Incoerência

O conceito de incoerência é estabelecido por meio da definição de coerência, como segue:

Definição 2.3: Sendo $\Phi = [\phi_1 \ \phi_1 \ \dots \ \phi_M]^T \in \mathbb{R}^{M \times N}$ e $\Psi = [\psi_1 \ \psi_1 \ \dots \ \psi_N] \in \mathbb{R}^{N \times N}$ tais que $\|\phi_i\|_2 = 1$, para qualquer $i = 1, 2, \dots, M$ e $\|\psi_j\|_2 = 1$, para qualquer $j = 1, 2, \dots, N$. É definida coerência entre Φ e Ψ como

$$\mu(\Phi, \Psi) = \sqrt{N} \cdot \max_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \{|\langle \phi_i, \psi_j \rangle|\}. \quad (2.17)$$

As colunas $\{\psi_j\}$ da transformação Ψ não podem ser esparsamente representadas pelas linhas $\{\phi_i\}$ da matriz de medida Φ . Ou seja, para que Ψ e Φ possam ser usadas elas devem ser incoerentes, portanto a medida de coerência dada na Equação (2.17).

Na Figura 2.7 é possível visualizar como as matrizes Φ e Ψ são construídas (ou representadas) pelos vetores linhas e colunas ϕ_i e ψ_j respectivamente.

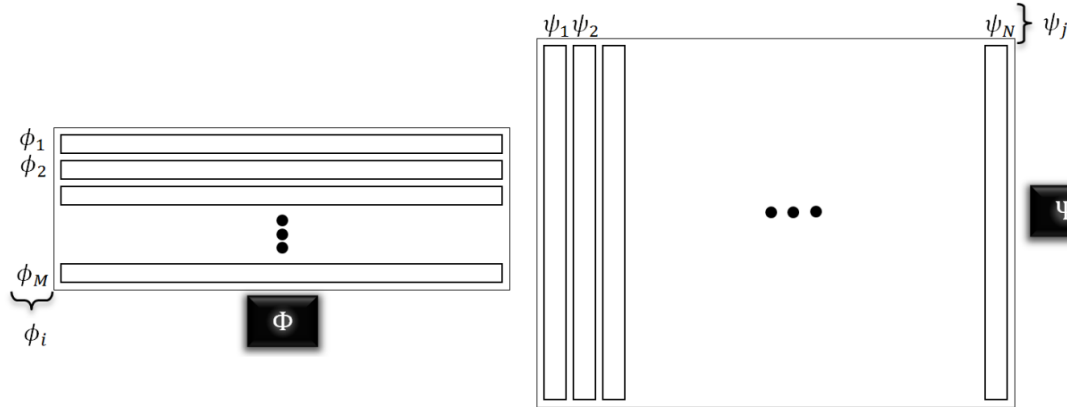


Figura 2.7: Linhas e colunas nas matrizes Φ e Ψ , respectivamente

Sendo a coerência $\mu(\Psi, \Phi) \in [1, \sqrt{N}]$, a mínima coerência será a máxima incoerência, assim idealmente a coerência deve ser $\mu(\Psi, \Phi) = 1$, considerando as matrizes ortonormais.

2.2.5 Critérios da Matriz de Medida

Não é qualquer transformação linear que pode ser usada como matriz de medida em CS, alguns critérios (ou condições) devem ser respeitados, essas condições são descritas a seguir.

2.2.5.1 Critério do espaço nulo

Para quaisquer vetores $\mathbf{s}_1 \neq \mathbf{s}_2$ e uma matriz Θ é necessário que $\Theta\mathbf{s}_1 \neq \Theta\mathbf{s}_2$. Desta forma sinais distintos têm medidas distintas ($\mathbf{y}_1 \neq \mathbf{y}_2$), pois se $\Theta\mathbf{s}_1 = \Theta\mathbf{s}_2 = \mathbf{y}$ não é possível saber qual dos dois sinais gerou a medida \mathbf{y} . Assim devem-se estabelecer condições para que isso não venha a ocorrer, uma condição necessária é o critério do espaço nulo.

Espaço nulo (ou núcleo) de uma transformação linear é definido como o conjunto de todos os vetores que aplicados a essa transformação geram o vetor nulo. Observe que o conceito está associado à transformação linear, ou seja: $\mathcal{N}(\Theta) = \{\mathbf{s} \mid \Theta\mathbf{s} = \mathbf{0}\}$.

Na Figura 2.8 tem-se uma ilustração que resume essas situações: em a) é desejável para reconstrução do sinal que qualquer \mathbf{s} esteja associado a uma medida distinta, em b) é indesejável que dois sinais estejam associados à mesma medida e em c) o núcleo da transformação linear Θ que é a matriz de medida.

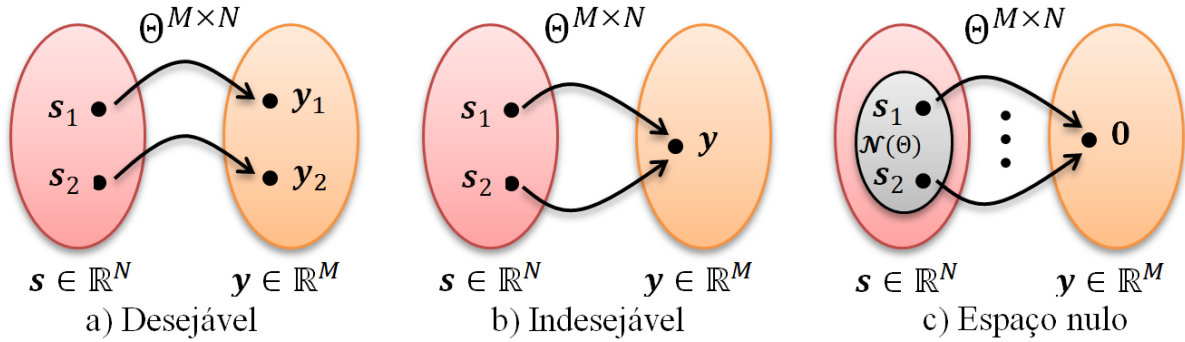


Figura 2.8: Comportamentos possíveis em uma transformação linear

O critério do espaço nulo estabelece que para quaisquer $\mathbf{s}_1 \neq \mathbf{s}_2$, ambos k -esparsos, é necessário que qualquer vetor do espaço nulo tenha no mínimo esparsidade maior que $2k$. Ou seja, para $\forall \mathbf{s} \in \mathcal{N}(\Theta)$ seja respeitado que $\|\mathbf{s}\|_0 > 2k$, sendo $\mathbf{s}_1 \neq \mathbf{s}_2$ ambos k -esparsos e com $\Theta\mathbf{s}_1 = \Theta\mathbf{s}_2$ tem-se que $\Theta\mathbf{s}_1 - \Theta\mathbf{s}_2 = \mathbf{0}$ ou melhor $\Theta(\mathbf{s}_1 - \mathbf{s}_2) = \mathbf{0}$. Assim, além desse vetor $(\mathbf{s}_1 - \mathbf{s}_2)$ fazer parte do espaço nulo, ele é $2k$ -esparso. Dessa forma, se for garantido que o espaço nulo não tenha vetores $2k$ -esparsos então não será possível que $\Theta\mathbf{s}_1 = \Theta\mathbf{s}_2$ ocorra, e portanto é possível a unicidade da solução.

2.2.5.2 Propriedade da Isometria Restrita (RIP)

Essa propriedade estabelece as condições necessárias e suficientes para garantir a unicidade da reconstrução do sinal medido, mesmo na presença de ruído (ou seja, sua robustez). Esse princípio encontra-se em [10] e [11] bem como suas demonstrações.

Definição da RIP: Para cada inteiro $s = 1, 2, \dots$, define-se a constante de isometria δ_s de uma matriz Θ como o menor número tal que

$$(1 - \delta_s) \|\mathbf{s}\|_{l_2}^2 \leq \|\Theta \mathbf{s}\|_{l_2}^2 \leq (1 + \delta_s) \|\mathbf{s}\|_{l_2}^2 \quad (2.18)$$

Válido para todos os vetores \mathbf{s} que sejam s -esparso. Um vetor é dito ser s -esparso se tem no máximo s valores diferentes de zero.

Uma pequena demonstração baseada na definição dessa propriedade prova essa unicidade. Suponha que $\mathbf{y} = \Theta \mathbf{s}_1 = \Theta \mathbf{s}_2$, com $\mathbf{s}_1 \neq \mathbf{s}_2$ e $\|\mathbf{s}_1\|_{l_0} = \|\mathbf{s}_2\|_{l_0} = k$; ou seja, tem-se hipoteticamente dois sinais diferentes \mathbf{s}_1 e \mathbf{s}_2 com a mesma esparsidade k , que geram a mesma medida \mathbf{y} , isso vai contra a unicidade.

Fazendo $\mathbf{h} = \mathbf{s}_1 - \mathbf{s}_2$, tem-se que $\|\mathbf{h}\|_{l_0} = 2k$, isto é, \mathbf{h} é $2k$ -esparso. Assim pode-se fazer $\Theta \mathbf{h} = \Theta \mathbf{s}_1 - \Theta \mathbf{s}_2 = \mathbf{y} - \mathbf{y} = \mathbf{0}$, e pela RIP: $(1 - \delta_{2k}) \|\mathbf{h}\|_{l_2}^2 \leq \|\Theta \mathbf{h}\|_{l_2}^2 = 0$, como $(1 - \delta_{2k}) > 0$, tem-se que $\mathbf{h} = \mathbf{0}$ e como $\mathbf{h} = \mathbf{s}_1 - \mathbf{s}_2$ implica que $\mathbf{s}_1 = \mathbf{s}_2$.

Conclui-se por contradição que se uma medida $\mathbf{y} = \Theta \mathbf{s}_1 = \Theta \mathbf{s}_2$ então $\mathbf{s}_1 = \mathbf{s}_2$ e tem-se a unicidade garantida para uma matriz de medida Θ com uma constante igual a δ_{2k} .

Outra consequência importante da RIP é que a métrica distância é preservada no domínio das medidas (no domínio comprimido), ou seja,

$$(1 - \delta_{2k}) \|\mathbf{s}_1 - \mathbf{s}_2\|_{l_2}^2 \leq \|\Theta \mathbf{s}_1 - \Theta \mathbf{s}_2\|_{l_2}^2 \leq (1 + \delta_{2k}) \|\mathbf{s}_1 - \mathbf{s}_2\|_{l_2}^2, \quad (2.19)$$

sendo sempre respeitada a constante δ_{2k} para a matriz Θ . Na Figura 2.9 é possível visualizar uma ilustração desta consequência, sendo d_s a distância entre dois vetores do sinal \mathbf{s} e d_y a distância entre dois vetores do sinal \mathbf{y} .

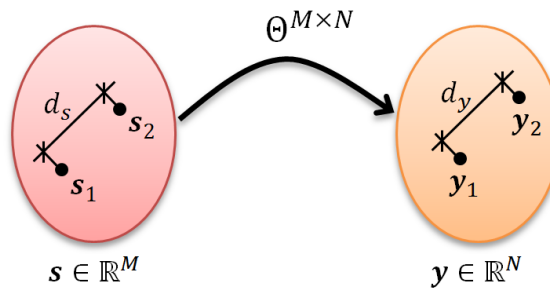


Figura 2.9: A métrica distancia é preservada no domínio das medidas.

Sendo $d_s = \alpha \cdot d_y$ e quanto menor for δ_{2k} menor será α .

2.3 Regularização

O sistema linear $\mathbf{y} = \Theta \mathbf{s}$ tem mais incógnitas (variáveis) que equações, e pode não ter solução ou ter infinitas soluções com vários graus de liberdade (variáveis independentes).

Certas condições garantem ter infinitas soluções. Contudo, a fim de estreitar a escolha para uma única solução, são necessários critérios adicionais. A forma geral para fazer isso é a regularização.

Assim uma função $J(\tilde{\mathbf{s}})$ é introduzida, que avalia a partir de alguma restrição a conveniência de uma possível solução $\tilde{\mathbf{s}}$, com valores pequenos sendo preferidos. Definindo assim o problema de otimização geral (\mathbf{P}_J) [12] como

$$(\mathbf{P}_J): \min_{\tilde{\mathbf{s}}} J(\tilde{\mathbf{s}}) \text{ sujeito a } \mathbf{y} = \Theta \mathbf{s} \quad (2.20)$$

A função $J(\cdot)$ é chamada de função objetiva, $\tilde{\mathbf{s}}$ é a variável de decisão e $\mathbf{y} = \Theta \mathbf{s}$ é a função de restrição. A otimização é linear quando todas as funções (objetiva e de decisão) são lineares, caso alguma dessas funções não seja linear a otimização é classificada como otimização não linear. Dependendo do contexto a otimização linear pode ser chamada de programação matemática, programação linear ou pesquisa operacional.

2.3.1 Reconstrução por l_2

O problema de minimizar a norma l_2 é formulado pela otimização (\mathbf{P}_2)

$$(\mathbf{P}_2): \min_{\tilde{\mathbf{s}} \in \mathbb{R}^N} \|\tilde{\mathbf{s}}\|_{l_2} \text{ sujeito a } \Theta \tilde{\mathbf{s}} = \mathbf{y}. \quad (2.21)$$

A escolha naturalmente usada em muitas aplicações de engenharia é a minimização da norma l_2 quadrada, equivalente ao problema (\mathbf{P}_2) . Usando os multiplicadores de Lagrange é possível provar que a solução é única e na forma fechada da Equação (2.22). Sendo $\Theta^T(\Theta\Theta^T)^{-1}$ chamada pseudo-inversa de Moore-Penrose de Θ e representada por Θ^+ .

$$\tilde{\mathbf{s}} = \Theta^T(\Theta\Theta^T)^{-1}\mathbf{y} = \Theta^+\mathbf{y}. \quad (2.22)$$

Pode-se assim encontrar uma solução ótima do problema, sem a necessidade de fazer uma busca. Contudo essa solução representa a resposta com menor energia e não é a resposta de interesse, ou seja, a mais esparsa. Esta solução é também conhecida como problema de mínimos quadrados, regressão dos mínimos quadrados ou otimização dos mínimos quadrados.

Usando o *software* CVX baseado na plataforma MATLAB, disponibilizado gratuitamente para usuários acadêmicos em [13], pode-se verificar a resposta na reconstrução pela minimização da norma l_2 .

Na Figura 2.10, pode ser observado o sinal original (em azul) e o sinal reconstruído (em vermelho), com um PSNR = 11.1114 dB e um SNR = 2.3479, é fácil verificar pelo gráfico que os valores reconstruídos são bem diferentes do sinal original e que estes têm valores pequenos (menor energia) e não esparsos.

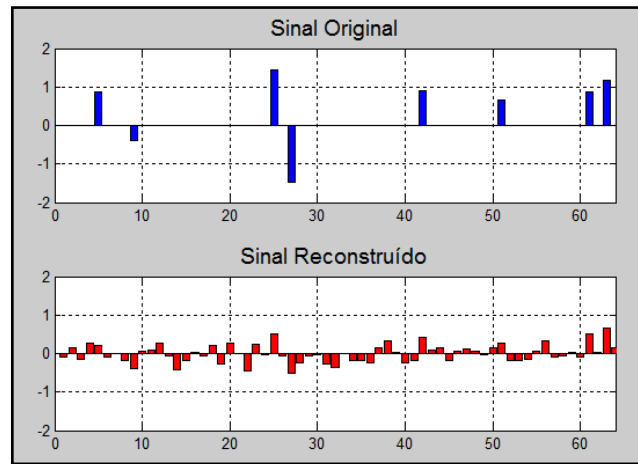


Figura 2.10: Reconstrução por l_2 , usando CVX.

Uma visualização gráfica do problema pode ser observada na Figura 2.11, em que o sistema $\mathbf{y} = \Theta \mathbf{s}$ é representado por um hiperplano no \mathbb{R}^N e a bola é definida pela norma l_2 , conforme visto na Figura 2.3 sendo que agora não é mais no \mathbb{R}^2 e sim no \mathbb{R}^3 .

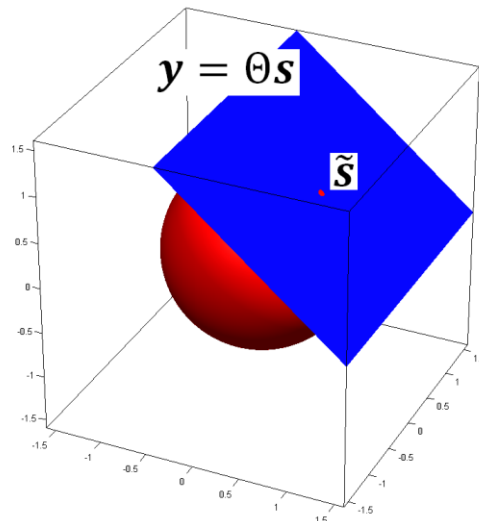


Figura 2.11: Visualização geométrica da solução do sistema pela norma l_2 .

A solução pela norma l_2 é o ponto $\tilde{\mathbf{s}}$ que é o único ponto comum à bola e ao plano, portanto a menor distância entre a bola e o plano. Contudo a solução mais esparsa está no eixo das coordenadas que faz intersecção com o plano e sendo de menor valor.

2.3.2 Reconstrução Teórica - l_0

A busca pela solução mais esparsa do sistema de equações leva naturalmente para a minimização da norma l_0 , a qual gera a solução com maior esparsidade. Neste caso a reconstrução do sinal é muito difícil, pois é um problema NP-completo do ponto de vista

computacional (custo temporal). Assim o problema de otimização (P_0) sendo a função objetiva $J(\tilde{\mathbf{s}}) = \|\tilde{\mathbf{s}}\|_{l_0}$ é

$$(P_0): \min_{\tilde{\mathbf{s}} \in \mathbb{R}^N} \|\tilde{\mathbf{s}}\|_{l_0} \text{ sujeito a } \Theta \tilde{\mathbf{s}} = \mathbf{y}. \quad (2.23)$$

A solução (P_0) é na verdade uma busca combinatória e é feita simplesmente por uma varredura em todos os possíveis subespaços esparsos de \mathbb{R}^N . Em [12] é encontrado o seguinte exemplo: suponha que Θ é do tamanho de 500x2000 ($M=500$, $N=2000$), e suponha que a solução esparsa para (P_0) tem esparsidade 20 ($\|\mathbf{s}\|_{l_0} = 20$). Deseja-se encontrar o conjunto apropriado de 20 colunas de Θ . Assim, é feita uma busca exaustivamente de todas essas opções: $\binom{2000}{20} = \frac{2000!}{(2000-20)!20!} \approx 3,9182 \cdot 10^{47}$. Para cada uma destas opções o sistema linear $\Theta \tilde{\mathbf{s}} = \mathbf{y}$ deve ser testado, isto é, se essa igualdade é satisfeita.

Embora a minimização da norma l_0 leve a uma solução exata, o custo temporal invalida tal estratégia, mesmo para valores pequenos de N .

2.4 Convexidade

O motivo da solução pela minimização da norma l_2 ser única é devido ao fato da norma l_2 ser estritamente convexa, e isto está relacionado com a geometria definida pelas normas. Contudo, vamos verificar formalmente a definição de convexidade [12].

Definição de conjunto convexo: Um conjunto Ω é convexo se $\forall \mathbf{s}_1, \mathbf{s}_2 \in \Omega$ e $\forall t \in [0, 1]$, a combinação convexa $\mathbf{s} = t\mathbf{s}_1 + (1-t)\mathbf{s}_2$ também pertence a Ω .

Definição de função convexa: Uma função $J(\mathbf{s}): \Omega \rightarrow \mathbb{R}$ é convexa se $\forall \mathbf{s}_1, \mathbf{s}_2 \in \Omega$ e $\forall t \in [0,1]$, o ponto da combinação convexa $\mathbf{s} = t\mathbf{s}_1 + (1-t)\mathbf{s}_2$ satisfaz

$$J(t\mathbf{s}_1 + (1-t)\mathbf{s}_2) \leq tJ(\mathbf{s}_1) + (1-t)J(\mathbf{s}_2). \quad (2.24)$$

Dessa forma, sendo $\Omega = \{\mathbf{s} \mid \Theta \mathbf{s} = \mathbf{y}\}$ um conjunto convexo e escolhendo uma função $J(\mathbf{s})$ também convexa, é garantido que o problema de otimização seja convexo como um todo.

Como visto na Figura 2.3, pela geometria de algumas normas no \mathbb{R}^2 , pode-se fazer o mesmo para uma bola unitária no \mathbb{R}^3 , definida por

$$B_p = \{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x}\|_p \leq 1\}. \quad (2.25)$$

Assim, pela visualização da geometria das normas l_2 , l_1 e $l_{0,7}$ na Figura 2.12, pode ser constatada a convexidade estrita de l_2 , a convexidade de l_1 e a não convexidade de $l_{0,7}$.

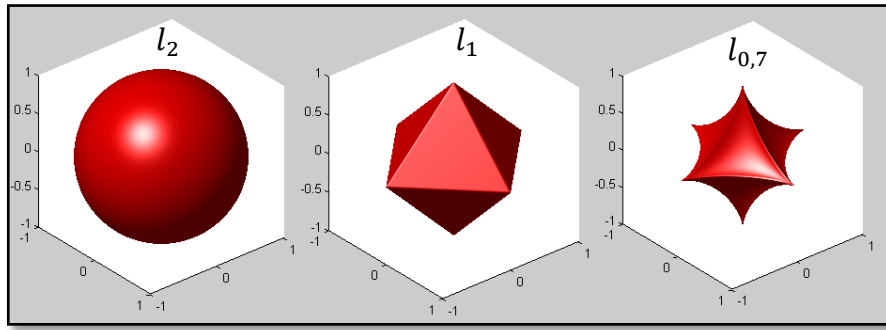


Figura 2.12: Bola unitária em \mathbb{R}^3 , para algumas normas l_p

A rigor qualquer norma l_p com $p < 1$ é não convexa, assim como para qualquer $p > 1$ a norma é estritamente convexa.

2.4.1 Reconstrução Prática - l_1

Dessa forma, para não se ter o problema de uma busca combinatória de l_0 e uma solução não esparsa de l_2 , deve-se procurar qual é a norma que seja pelo menos convexa e produza a solução mais esparsa possível. Assim, a escolha mais conveniente é a minimização pela norma l_1 , ou seja, para $J(\tilde{\mathbf{s}}) = \|\tilde{\mathbf{s}}\|_{l_1}$, caracterizada pela otimização (P_1) , escrita de maneira formal

$$(P_1): \min_{\tilde{\mathbf{s}} \in \mathbb{R}^N} \|\tilde{\mathbf{s}}\|_{l_1} \text{ sujeito a } \Theta \mathbf{s} = \mathbf{y}. \quad (2.26)$$

Graficamente, a solução por l_1 pode ser visualizada na Figura 2.13, sendo o sistema $\mathbf{y} = \Theta \mathbf{s}$ representado pelo hiperplano em azul e norma l_1 pela bola em vermelho.

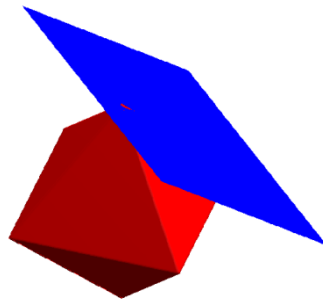


Figura 2.13: Geometria da bola da norma l_1 com o hiperplano do sistema $\mathbf{y} = \Theta \mathbf{s}$.

Sabendo que a solução mais esparsa se encontra em um dos eixos das coordenadas, basta verificar qual dos eixos tocará primeiro no hiperplano à medida que o raio desta bola for crescendo, e por fim verificar a magnitude desta solução esparsa.

Usando-se o CVX pode-se verificar também a reposta na reconstrução do sinal pela minimização da norma l_1 . Na Figura 2.14, pode ser observado o sinal original (em azul) e o sinal reconstruído (em vermelho), com um PSNR=182,1119dB e um SNR=173,3484. É fácil

verificar pelo gráfico que os valores reconstruídos são bem próximos do sinal original, e que estes valores são igualmente esparsos.

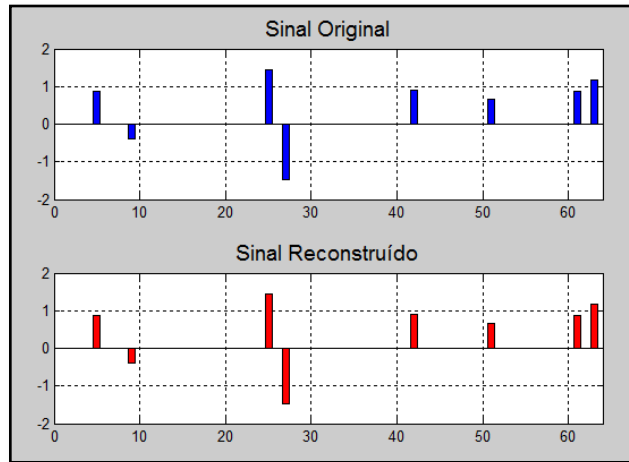


Figura 2.14: Reconstrução por l_1 , usando CVX.

Alguns algoritmos podem ser aplicados para a reconstrução de sinais a partir das medidas y obtidas usando a Amostragem Compressiva. Alguns desses algoritmos foram desenvolvidos em outros contextos, mas são perfeitamente aplicáveis no problema em questão. Este trabalho se limita em como fazer essas medidas e não na reconstrução do sinal, uma descrição mais adequada para os algoritmos de reconstrução pode ser encontrada em [14].

Pode-se classificar algumas dessas técnicas em quatro categorias:

1. Relaxação convexa (*Convex relaxation*): utiliza um problema de otimização convexa para recuperar o sinal esparsos.
2. Arcabouço Bayesiano (*Bayesian framework*): assume uma distribuição a priori do sinal esparsos e utilizando estimação recupera o sinal esparsos.
3. Otimização não-convexa (*Nonconvex optimization*): utiliza métodos de otimização não convexa para recuperar o sinal esparsos.
4. Combinatórias: fazem uma busca sobre todos os possíveis conjuntos de suporte para determinar em quais deles estão os coeficientes do sinal esparsos.

2.4.2 Reconstrução Prática - l_1 com ruído

Na reconstrução por l_1 , mesmo na presença de ruído é possível recuperar o sinal original e dependendo das características desse ruído, é definido na reconstrução um erro tolerável \mathcal{E} . Deve ficar claro que esse ruído pode ser tanto do sistema (chamado *noise folding*), quanto o que já acompanha o sinal original. Esse último pode ter sido somado ao

sinal no canal de comunicação. Na Figura 2.15 é mostrada uma ilustração de como esses dois ruídos podem ser modelados com o sistema.

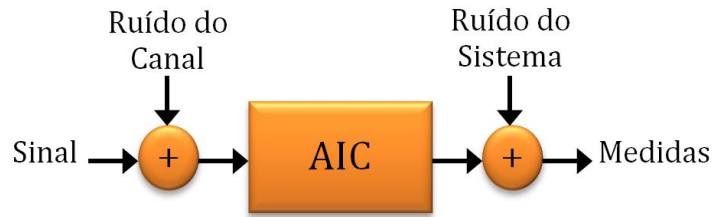


Figura 2.15: Modelo dos ruídos do canal e do sistema

Na Equação (2.27) é feita a representação das medidas com a introdução apenas do ruído do sistema (\mathbf{n}). Na Equação (2.28) a representação apenas com o ruído do canal (\mathbf{z}) e na Equação (2.29) com os dois ruídos juntos, que é o pior caso.

$$\mathbf{y} = \Theta \mathbf{s} + \mathbf{n} \quad (2.27)$$

$$\mathbf{y} = \Theta(\mathbf{s} + \mathbf{z}) \quad (2.28)$$

$$\mathbf{y} = \Theta(\mathbf{s} + \mathbf{z}) + \mathbf{n} \quad (2.29)$$

Observando a Equação (2.30) é possível verificar como esse ruído tolerável pode ser especificado na reconstrução do sinal original.

$$\min_{\tilde{\mathbf{s}} \in \mathbb{R}^N} \|\tilde{\mathbf{s}}\|_{l_1} \quad \text{sujeito a} \quad \|\Theta \mathbf{s} - \mathbf{y}\|_{l_2} \leq \mathcal{E} \quad (2.30)$$

2.5 Arquiteturas dos AIC

Conforme descrito na Seção 2.1 os Sistemas de Aquisição de Dados estão fundamentados no teorema de Nyquist, ou seja, no conteúdo espectral do sinal e não na informação nele contida. Isto leva a questionamentos, como a celebre indagação de Donoho [1].

*“The phenomenon of ubiquitous compressibility raises very natural questions: why go to so much effort to acquire **all** the data when **most** of what we get will be thrown away? Can we not just **directly measure** the part that will not end up being thrown away?”*

Desta forma algumas arquiteturas de *hardware* vêm sendo desenvolvidas com a finalidade de digitalizar apenas a informação, esses conversores são chamados de Conversores Analógicos para Informação (*Analog to Information Converter*) abreviados por AIC ou algumas vezes por A2I, fazendo alusão aos ADC e A2D.

De um modo geral, para extrair a informação do sinal $x(t)$ que seja suficiente para reconstruí-lo a partir das medidas realizadas, é necessário em um intervalo finito $t \in [0, T]$ utilizar um conjunto de funções teste $\{\phi(t)\}_{j=1}^M$ e realizar M medidas, cada uma da forma

$$y[j] = \int_0^T x(t) \phi_j(t) dt, \quad j = 1, 2, \dots, M, \quad (2.31)$$

que é a representação analógica de uma única medida vista na Equação (2.12), reescrita aqui $\mathbf{y} = \Phi \mathbf{x}$, que para cada medida não deixa de ser um produto escalar (ou uma projeção), isto é $y[j] = \langle \mathbf{x}[n], \boldsymbol{\phi}_j[n] \rangle$, com o tamanho de $\mathbf{y}[n]$ igual a M e bem menor que N .

Esse sistema de medição é constituído fundamentalmente de três componentes, a saber:

- * *Hardware* para gerar os sinais de teste $\phi_j(t)$;
- * M multiplicadores que geram os produtos do sinal $x(t)$ com cada um dos $\phi_j(t)$;
- * M integradores que geram as medidas da integração do produto anterior, sendo sempre o estado inicial com valor zero.

A implementação de conversores AIC vem sendo foco de várias pesquisas, contudo na bibliografia especializada encontram-se poucos projetos práticos, e mesmo esses na sua maioria são para sinais específicos. Várias arquiteturas de AIC vêm sendo propostas, sempre com o objetivo de evitar a taxa de Nyquist. A seguir apresenta-se uma descrição das principais arquiteturas desses conversores.

2.5.1 Demodulador Aleatório

A arquitetura do Demodulador Aleatório (do inglês, *Random Demodulation*- RD) [15] é representada genericamente na Figura 2.16, e correlaciona o sinal de entrada $f(t)$ (que contém a informação desejada) com uma sequência pseudo-aleatória de pulsos com amplitudes de ± 1 . Esse sinal é chamado de sequência de *chipping* $p_c(t)$.

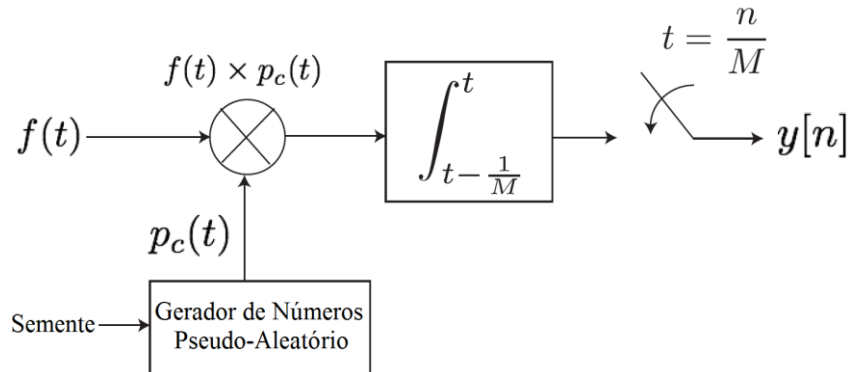


Figura 2.16: Diagrama em blocos do Demodulador Aleatório – RD[16].

O RD realiza uma multiplicação de $f(t)$ por $p_c(t)$, resultando em um espalhamento espectral. Desta forma, a informação contida em $f(t)$ também se espalha por todo o espectro. Em seguida, uma integração delimita parte do espectro que será digitalizado com uma taxa de amostragem inferior a de Nyquist (que depende do comprimento de integração). Uma visualização do comportamento dos sinais no domínio do tempo e da frequência de cada etapa pode ser vista na Figura 2.17.

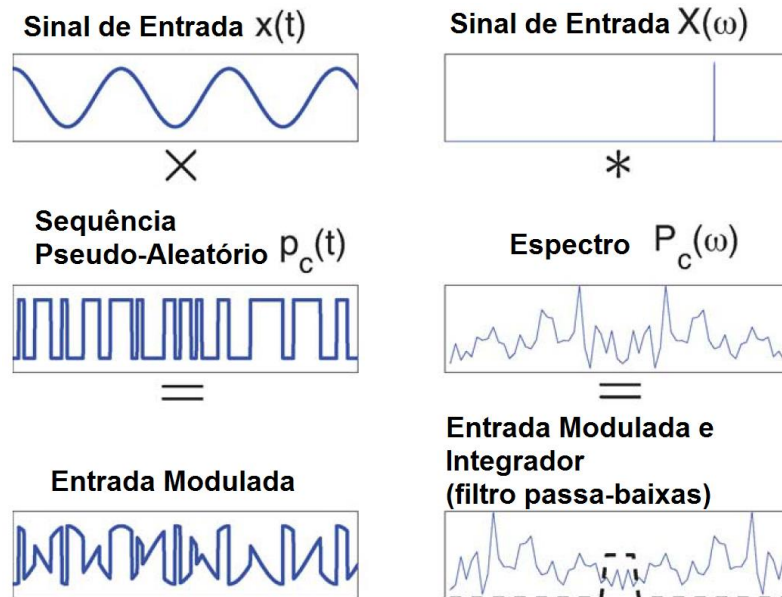


Figura 2.17: Comportamento dos sinais nas etapas do RD [17]

Segundo [17], esse espalhamento espectral é visto como uma assinatura do sinal, ou seja, para uma mesma sequência $p_c(t)$ cada sinal terá um formato próprio depois da multiplicação. Assim, o RD apresenta baixa complexidade de implementação, contudo suporta apenas a reconstrução de sinais multi-tom discreto. Alguns trabalhos com essa arquitetura podem ser vistos em [16]–[20].

2.5.2 Conversor de Banda Larga Modulada

O Conversor de Banda Larga Modulada (do inglês, *Modulated Wideband Converter-MWC*) é uma arquitetura paralela, em que cada elemento paralelo é chamando de canal. O sinal de entrada é do tipo RF multibanda, assim cada canal recebe o mesmo sinal RF de entrada e mistura com uma forma de onda distinta para cada canal. Dessa forma, na saída de cada misturador tem-se em banda base uma combinação das bandas ao longo do espectro de frequência. Esse sinal na banda base é então filtrado (filtro passa-baixa) e amostrado uniformemente a taxa de sub-Nyquist (ADC de taxa baixa por canal). O número de canais depende da taxa de amostragem em cada canal.

Na Figura 2.18 é possível observar a estrutura paralela do MWC, sendo $x(t)$ o sinal de RF multibanda de entrada, $p_i(t)$ é uma sequência pseudoaleatória de ± 1 com frequência muito alta (o índice i representa cada canal e vai até m), $\tilde{x}_i(t)$ é o sinal na banda base de cada canal i referente a uma determinada combinação das bandas do espectro e por fim $y_i[n]$ é o sinal de banda base filtrado e amostrado.

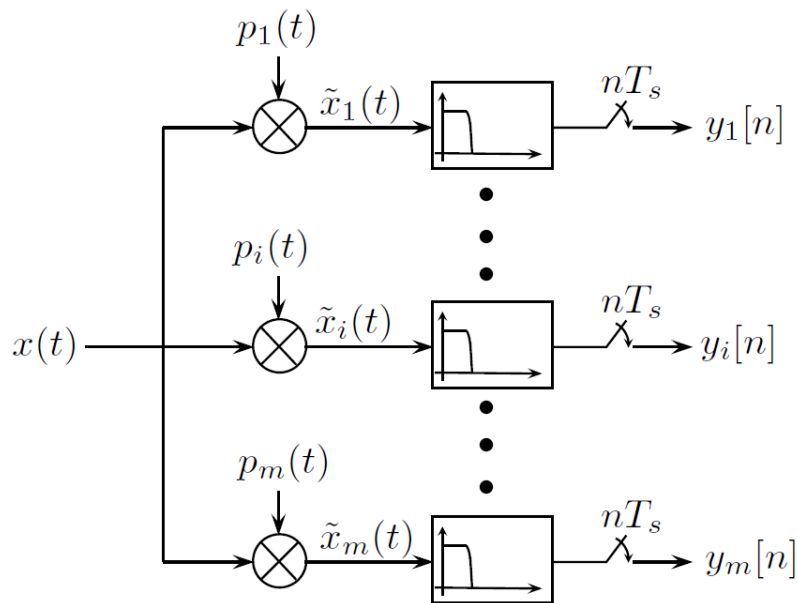


Figura 2.18: Diagrama de bloco do MWC [21]

Uma ilustração de um possível espectro de frequência multibanda de $x(t)$, bem como duas possíveis combinações lineares em banda base $y_i[n]$ e $y_{i'}[n]$ são mostradas na Figura 2.19. É possível observar que para cada $y[n]$ existe informação dos sinais espalhados no espectro. Assim, cada banda dessa é digitalizada para depois ser feita a reconstrução de todo espectro.

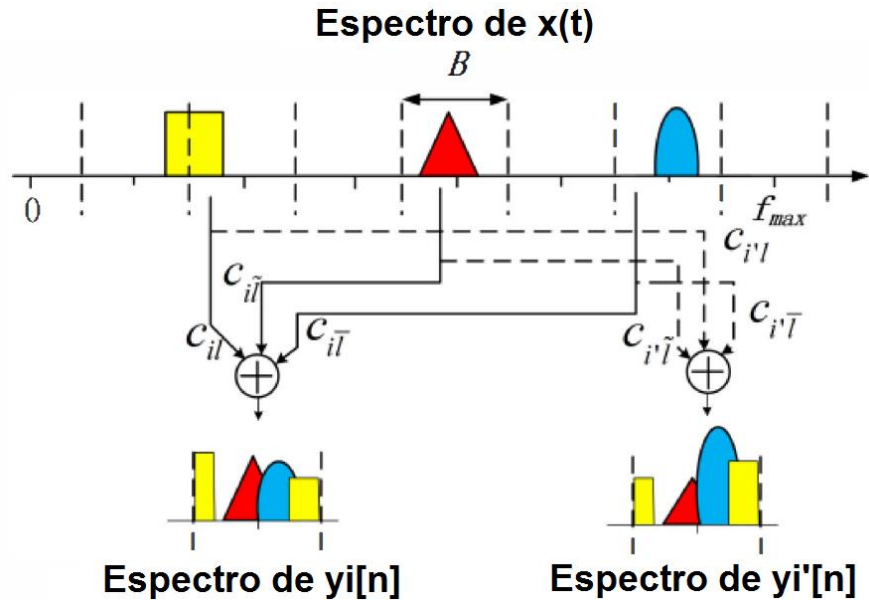


Figura 2.19: A combinação linear em banda base de 2 canais do MWC [22]

O MWC é configurado para um determinado padrão estático do espectro multibanda, e cada vez que esse padrão espectral muda, uma nova configuração deve ser realizada. Alguns trabalhos sobre essa arquitetura podem ser vistos em [21]–[31].

2.5.3 Amostrador Não Uniforme

O Amostrador Não Uniforme (do inglês, *Non-Uniform Sampler-NUS*) é outra maneira de digitalizar a informação de sinais que são esparsos no domínio da frequência. Na Figura 2.20 pode ser observada sua arquitetura, o sinal $x(t)$ é digitalizado por um A2D que trabalha na taxa de Nyquist. O sinal já digitalizado vai para um seletor que através de uma sequência aleatória define se o valor é descartado ou mantido.

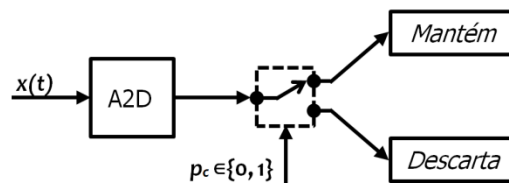


Figura 2.20: Diagrama em blocos do NUS.

Dessa forma, a implementação de um NUS necessita de um conversor A2D, um gerador de sequências aleatórias e uma chave seletora. Existem algumas vertentes dessa arquitetura que se diferenciam basicamente pela maneira como o *clock* é usado, listadas aqui: *Periodic Non-Uniform Sampling* (PNUS), *Randomized Nonuniform Sampling* (RNUS) e *Level-Triggered Nonuniform Sampling* (LTNUS); uma breve descrição dessas vertentes do NUS pode ser vista em [32]. De modo geral, alguns trabalhos relacionados com esta arquitetura podem ser encontrados em [33]–[37].

2.5.4 Pré-Integrador de Modulação Aleatória

O Pré-Integrador de Modulação Aleatória (do inglês, *Random Modulation Pre-Integrator-RMPI*) possui características tanto da arquitetura RD como do MWC. Na Figura 2.21 é possível observar algumas destas semelhanças.

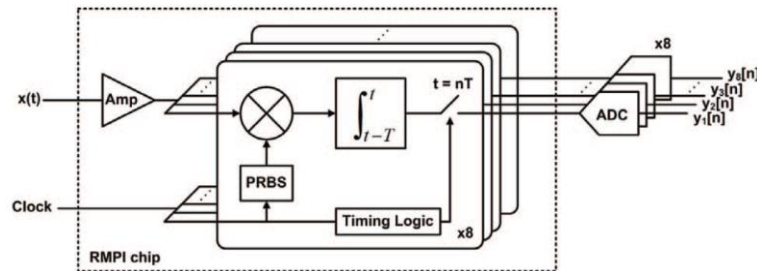


Figura 2.21: Diagrama em bloco do RMPI [38].

O RMPI possui a mesma estrutura paralela do MWC, contudo usa um circuito integrador semelhante ao RD. Cada canal possui um multiplicador que funciona como um *down-converter* que ao invés de um oscilador, usa um Sequência Binária Pseudo-Aleatória (*Pseudo Random Binary Sequence* – PRBS). Essas sequências são usadas por serem incoerentes com qualquer base estruturada. Segundo Yoo [38] essa arquitetura é relativamente universal, pois funciona com sinais que são esparsos em qualquer domínio fixo. Essa arquitetura foi usada com sucesso na medição de sinais de trens de impulsos curtos de radar embarcados numa banda ultra-larga, e pode ser encontrada em detalhes na tese de doutorado de Becker [39]. Alguns trabalhos com essa arquitetura podem ser encontrados em [38], [40]–[43].

2.5.5 Xampling

Xampling é baseada no conceito de união de subespaços. É uma arquitetura mais geral e mais complexa que as anteriores. Na Figura 2.22 é mostrado o seu diagrama de blocos que consiste de um bloco que faz a compressão pela união de subespaços P , e em seguida o sinal é digitalizado por um A2D clássico em uma taxa menor que a de Nyquist. Essa taxa é menor por causa da compressão realizada pelo operador P .



Figura 2.22: Diagrama de blocos do Xampling.

O termo Xampling faz uma alusão ao termo *sampling*, sendo que o prefixo X do termo indica uma redução da taxa, da mesma forma que um X-ADC é um conversor analógico para digital de baixa taxa, ou mesmo um ADC Xampling.

Um detalhamento dessa arquitetura pode ser visto na Figura 2.23, constituído de uma parte analógica chamada X-ADC (as vezes chamada de *front-end*) e uma outra digital chamada de X-DSP (que pode ser implementada em *hardware* ou *software*). Um terceiro grande bloco, que na teoria não faz parte do AIC, é chamado *subspace reconstruction*.

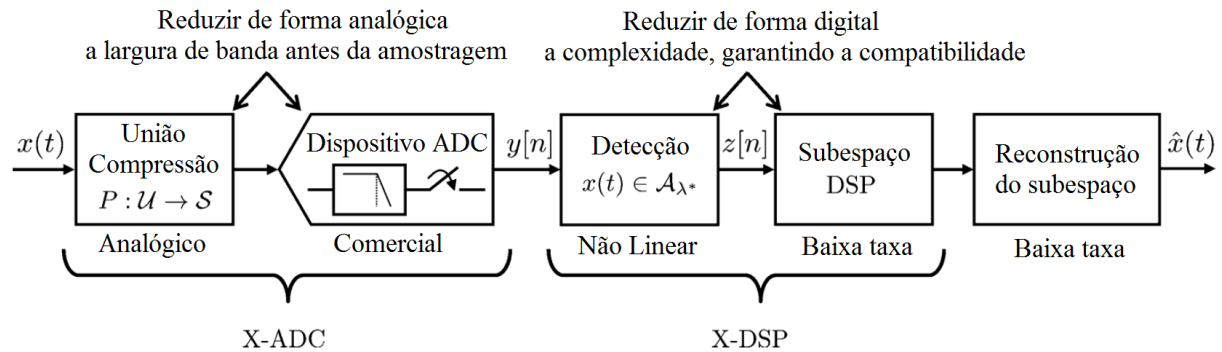


Figura 2.23: Xampling - Uma estrutura para processamento da união de subespaço[44]

O X-ADC (*front-end*) pode ser implementado, em princípio, com qualquer das arquiteturas já vistas, isto depende muito das características do sinal a ser medido, contudo na prática muitos trabalhos utilizam o MWC. Trabalhos relacionados com essa arquitetura podem ser encontrados em [44]–[48].

3 DESENVOLVIMENTO

Na revisão bibliográfica, apresentada na Seção 2.5, foi constatado que vários parâmetros devem ser configurados no *hardware* para garantir a reconstrução do sinal, tais como: tipo, tamanho e frequência da Sequência Binária Pseudo-Aleatória (PRBS), frequência de corte, seletividade e ganho do filtro, além de características dos pulsos de sincronismo do filtro, tais como: período e tempo de *reset*, período do pulso de sincronismo do ADC, etc.

É certo que esses parâmetros dependem da arquitetura escolhida, e essa última vai depender das características do sinal esparsa a ser adquirido. Constata-se também que as diversas arquiteturas se assemelham muito (com exceção da NUS).

Neste trabalho, foi desenvolvido e implementado um *hardware* que pode configurado de acordo com as características gerais do sinal esparsa a ser adquirido, e que é descrito neste capítulo.

3.1 Visão geral

Tendo em vista que o objetivo deste trabalho é propor uma arquitetura de um AIC que seja configurável, foi necessário construir um cenário de ensaio relativamente flexível, para que seja possível ensaiar essa arquitetura em diversas situações e assim validar o *hardware* proposto, bem como determinar as suas limitações. Na Figura 3.1 está a ilustração desse cenário. Alguns instrumentos importantes para a construção deste cenário foram omitidos por não serem utilizados diretamente nas aquisições, mas para verificações eventuais, sendo eles multímetro, osciloscópio, fonte de tensão e outros. O cenário construído é descrito a seguir.

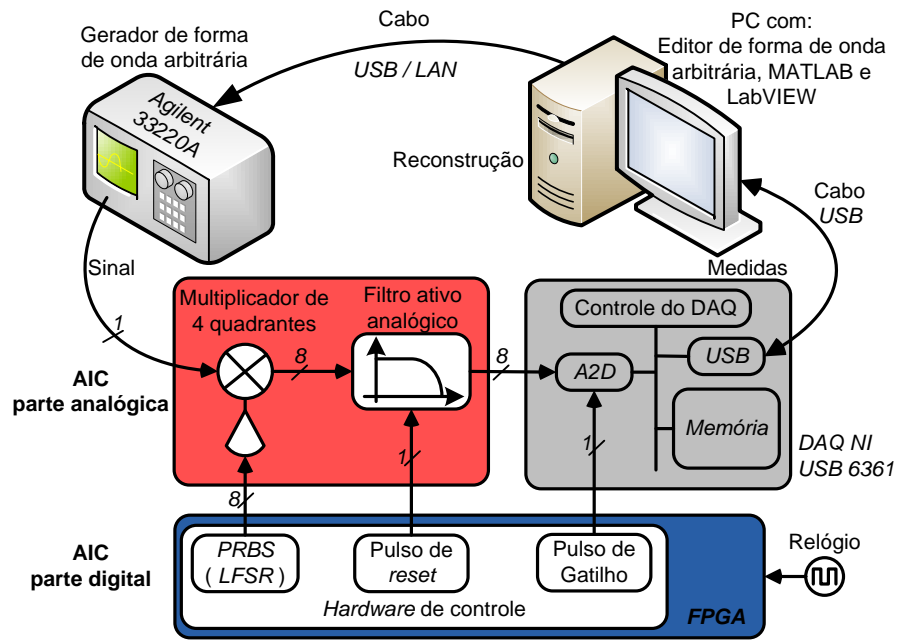


Figura 3.1: Ilustração do Cenário de Ensaio do AIC

O AIC desenvolvido é constituído basicamente de um *hardware* digital (AIC parte Digital) implementado em uma FPGA (da empresa Altera) e de um *hardware* analógico (AIC parte Analógica) implementado em uma PCI (Placa de Circuito Impresso) utilizando componentes comerciais (classificados de *Commercial Off-The-Shelf* - COTS), uma descrição mais detalhada sobre esse *hardware* é feita na Seção 3.2.

O gerador de forma de onda arbitrária (*Arbitrary Waveform Generator*-AWG) utilizado (Keysight Technologies modelo 33220A) é um gerador de funções, que fornece os sinais de teste previamente editados. Desta forma, esses sinais são bem conhecidos e isto é de fundamental importância para que se possa verificar o desempenho do AIC após a reconstrução do sinal, feito por meio das aquisições realizadas.

O sinal na saída, na parte analógica do AIC, é digitalizado por uma placa de aquisição de dados (DAQ) comercial (National Instruments DAQ USB 6361), o sinal digitalizado é chamado formalmente de Medida (*Measure*), e é enviado para um computador pessoal (PC) por meio de um cabo USB, este DAQ foi utilizado por ter flexibilidade nas suas configurações em termos de: frequência de amostragem, número de canais de entrada analógica, número de amostras, faixa de tensão do sinal de entrada, entrada de gatilho (*trigger*) e outras.

O PC é um elemento indispensável não apenas para o ensaio e reconstrução do sinal, mas também para o desenvolvimento do próprio AIC. Vários *softwares* são utilizados para essas finalidades. A seguir é feita uma pequena descrição das principais ferramentas de *software* usadas neste trabalho.

IntuiLink Waveform Editor: é um aplicativo no qual é feita a edição dos sinais de testes e em seguida enviado para o AWG por meio de um cabo USB (ou cabo de rede). Esses sinais ficam gravados no computador e no gerador em uma memória não volátil de forma normalizada. Assim, para um determinado sinal de teste é possível modificar no gerador parâmetros como frequência, amplitude, etc.

LabVIEW: é classificado como uma linguagem de programação gráfica (chamada de Linguagem G). Embora seja uma ferramenta com muitas funções para engenharia, sua principal função neste trabalho foi configurar os diversos parâmetros do DAQ e salvar as aquisições realizadas no mesmo.

MATLAB: é um *software* da MathWorks e pode ser classificado como uma linguagem de programação, essa ferramenta foi utilizada neste trabalho principalmente para reconstrução do sinal, usando os dados das aquisições pelo LabVIEW ou por um simulador. No MATLAB é possível acessar uma ferramenta de modelagem, simulação e análise de sistemas dinâmicos chamada de Simulink. Nesta ferramenta foi feita a simulação do comportamento dinâmico do *hardware* analógico e digital do AIC (AIC em *Hardware*), uma melhor descrição destas simulações é feita na seção 3.4. Várias outras atividades são também realizadas com estas ferramentas.

OrCAD: utilizado para edição, simulação e projeto de PCI de circuitos eletrônicos, foi principalmente usado no *hardware* do AIC analógico. As simulações com esta ferramenta foram realizadas por meio de uma versão do SPICE (*Simulation Program with Integrated Circuit Emphasis*), denominado PSPICE, pois é uma versão projetada para o PC.

Quartus II: é uma ferramenta de projeto para dispositivos lógicos programáveis - PLD (principalmente às FPGA), desenvolvida pela Altera para os seus PLD. Neste *software* a edição (ou descrição) do *hardware* pode ser feita de várias formas, a saber: 1) pelas linguagens de descrição de *hardware* – HDL, tais como: Verilog, VHDL e AHDL; 2) por esquemáticos (ou diagrama de blocos); 3) por máquinas de estados finitos (ou autômatos finitos determinísticos), dentre outros. Várias destas formas foram usadas neste trabalho. Além da Edição, no Quartus II foram realizadas as demais etapas do projeto da parte digital

do AIC, a saber: Compilação, Simulação (com o Quartus II e o ModelSim), Verificação e Gravação.

Na Figura 3.2 é mostrada uma imagem deste cenário de ensaio do AIC.

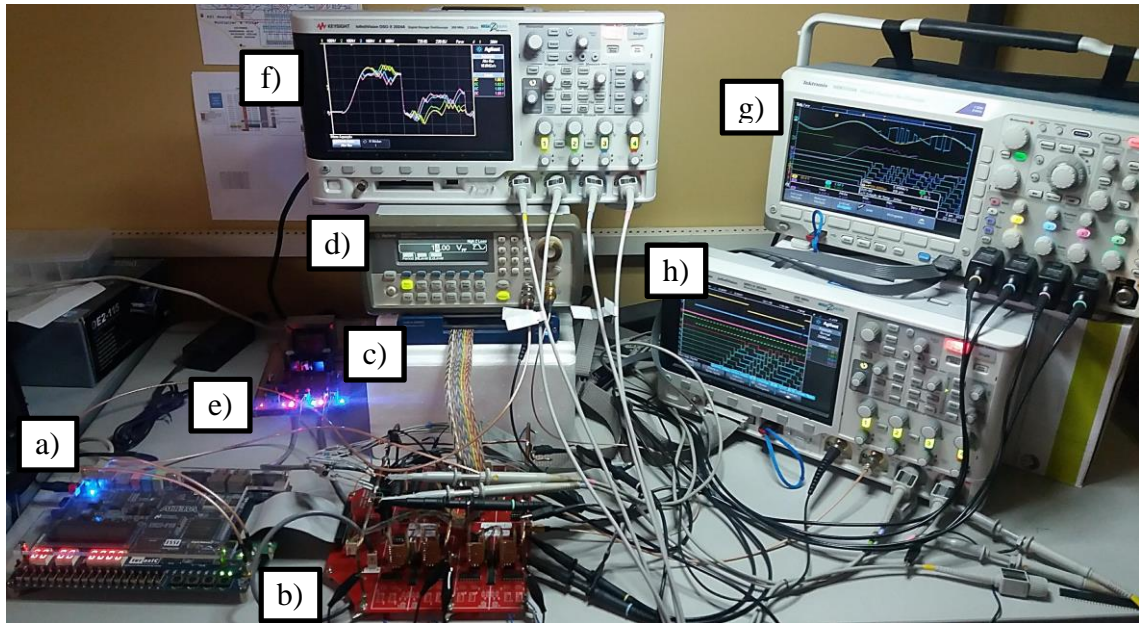


Figura 3.2: Imagem do Cenário de Ensaio do AIC

3.2 AIC em *Hardware*

O AIC em *Hardware* é construído de duas partes, uma digital implementada em uma FPGA e uma analógica desenvolvida em uma placa de circuito impresso (PCI). Essas duas partes juntas realizam as aquisições para posterior reconstrução. As reconstruções são realizadas em um computador pessoal (PC).

Na imagem da Figura 3.2 é possível verificar um ensaio realizado no AIC em *Hardware*, a saber: em a) A placa de desenvolvimento com a FPGA utilizada, na FPGA desta placa está implementada a parte digital do AIC; em b) A parte analógica do AIC, ligada à parte digital por meio de um cabo com várias vias de dados; em c) O DAQ (com o ADC) realiza a digitalização dos 8 sinais do AIC, isso ocorre após receber um sinal de gatilho vindo de um circuito interno da FPGA, esses sinais são ligados do AIC para o DAQ por meio de outro cabo (esse com detalhes coloridos); em d) O gerador de forma de onda arbitrária, no qual previamente já foram gravados os sinais a serem adquiridos; em e) Uma fonte de alimentação DC, desenvolvida especialmente para alimentar os circuitos da parte analógica do AIC; em f) Um Osciloscópio com os sinais na saída de 4 integradores (filtros); em g) Um

Osciloscópio com vários sinais ao longo de um canal, dentre outros; e em h) Um Osciloscópio com os sinais de controle, dentre outros.

3.2.1 AIC *Hardware* Digital

A parte digital do AIC (AIC *Hardware* Digital) é implementada em um *chip* de FPGA (Altera EP4CE115 [49]), utilizado na placa de desenvolvimento DE2-115 [50]. A ferramenta de desenvolvimento principal foi o Quartus II.

Conforme visto na Figura 3.1, a parte digital do AIC é constituída de: 1) Circuito de controle (*Hardware* de Controle); 2) Gerador PRBS (*LFSR*); 3) Circuito de sincronismo do *reset* (Pulso de *reset*); e 4) Circuito de sincronismo do DAQ (Pulso de Gatilho).

O projeto foi descrito em hierarquia e seu nível mais alto (*top-level*) é um esquemático que possui blocos funcionais descritos em VHDL, Verilog e esquemático. Uma descrição de cada parte do circuito digital do AIC é feita a seguida:

3.2.1.1 Circuito de Controle (*Hardware* de Controle)

O circuito de controle habilita dois sinais de relógio (*clock*) sincronizados, que são usados: no circuito de sincronismo do reset; no circuito de sincronismo do DAQ; e no Circuito Gerador PRBS (*LFSR*). É possível habilitar esses sinais de relógio de duas formas:

Manual: apenas pressionado o botão externo ligado a FPGA.

Automático: por meio de um sinal de gatilho (*trigger*) externo, que pode estar associado a alguma característica do sinal a ser medido, como por exemplo, o início do ciclo positivo do sinal.

Por fim, é possível modificar as frequências dos sinais de relógio por meio de um bloco de PLL (*Phase Locked Loop*) Programável interno à FPGA. Essas frequências são configuradas e estão associadas à frequência fundamental do sinal a ser medido, uma descrição mais detalhada do funcionamento de um PLL programável pode ser encontrada em [51].

3.2.1.2 Gerador PRBS

O Gerador PRBS é um *hardware*, ou *software*, que gera uma sequência de números binários pseudo-aleatórios, também conhecido como Gerador de Números Pseudo-aleatório (*Pseudo Random Number Generator* - PRNG), quando feito em *hardware* é implementado usando um registrador de deslocamento com realimentação linear (*Linear Feedback Shift Register* – LFSR). Esse LFSR é construído usando um registrador de deslocamento simples (*shift register* - SR) com *flip-flops* tipo D (ou T), que realimenta o primeiro *flip-flop* com uma soma em modulo 2 (portas XOR ou XNOR) das saídas de alguns *flip-flops* do registrador de deslocamento, essas saídas (ou pontos de saídas) são chamados de *taps*.

A escolha das combinações dos *taps* caracteriza um polinômio de módulo 2. Como exemplo: para os *taps* [1, 2] o polinômio é $1 + 0x + 1x^2 + 1x^3$ (ou $1 + x^2 + x^3$) e para os *taps* [0, 2] o polinômio é $1 + 1x + 0x^2 + 1x^3$ (ou $1 + x + x^3$). Esses polinômios sempre começam e terminam da forma $1 + \dots + 1x^n$, sendo n o número de *flip-flops*.

Esses polinômios (ou *taps*) são baseados na teoria algébrica de corpos finitos (ou Campo de Galois). É justificada a escolha desses polinômios para garantir que um LFSR de n bits tenha o máximo de sequências possíveis. Assim, deve-se escolher um polinômio primitivo de ordem n , ou mais formalmente, escolher um polinômio primitivo em $GF(2^n)$. No Quadro 3.1 são apresentadas as expressões para o cálculo dos parâmetros das sequências dos LFSR de n bits e exemplificado numericamente para n igual a 3, 7 e 16.

LFSR de n bits	n	$n=3$	$n=7$	$n=16$
Tamanho da Sequência	$2^n - 1$	7	127	65535
# de 1	2^{n-1}	4	64	32768
# de 0	$2^{n-1} - 1$	3	63	32767
# de bordas	2^{n-1}	4	64	32768

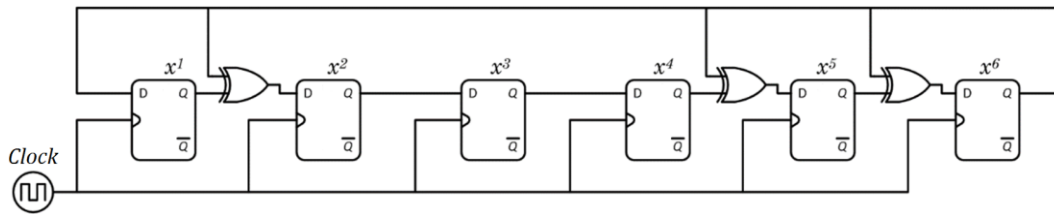
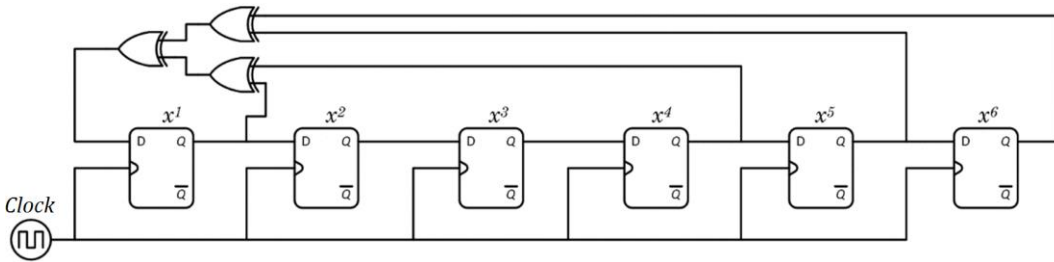
Quadro 3.1: Expressões e Exemplos dos números das sequências dos LFSR com XOR.

Esses polinômios podem ser gerados no MATLAB, contudo é possível encontrar na bibliografia técnica tabelas com valores de n bem elevados. Um pequeno exemplo pode ser visto no Quadro 3.2. Se o primeiro *Flip-Flop* for numerado como 1 e não como 0, os *taps* vão ficar com a numeração de acordo com os graus dos polinômios primitivos, i.e. $1 + x^2 + x^5$ fica [2, 5].

# de <i>Flip-Flops</i>	Tamanho da Sequência	<i>taps</i>	Polinômio Primitivo
2	3	[0,1]	$1 + x + x^2$
4	15	[0,3]	$1 + x + x^4$
8	255	[1,2,3,7]	$1 + x^2 + x^3 + x^4 + x^8$
16	65535	[1,2,4,15]	$1 + x^2 + x^3 + x^5 + x^{16}$
32	4294967295	[1,5,6,31]	$1 + x^2 + x^6 + x^7 + x^{32}$
64	$\approx 1.844e19$	[59,60,62,63]	$1 + x^{60} + x^{61} + x^{63} + x^{64}$

Quadro 3.2: Parâmetros e Exemplos de construção de um LFSR.

As arquiteturas (ou configurações) dos LFSR podem ser classificadas pelo modo como é feita a realimentação. Na Figura 3.3 é ilustrado um exemplo da arquitetura de Galois (ou *One-to-Many*) e na Figura 3.4 a arquitetura de Fibonacci (ou *Many-to-One*) para o mesmo polinômio primitivo [52], [53].

Figura 3.3: LFSR Arquitetura de Galois (ou *One-to-Many*) para o Polinômio $1 + x + x^4 + x^5 + x^6$ Figura 3.4: LFSR Arquitetura de Fibonacci (ou *Many-to-One*) para o Polinômio $1 + x + x^4 + x^5 + x^6$

Se a soma dos atrasos das portas XOR em série na arquitetura de Fibonacci for próxima do período do sinal de relógio, é recomendado usar a arquitetura de Galois, pois o atraso é apenas de uma porta XOR. É possível fazer implementações em FPGA desses circuitos com frequência de relógio próximas de 1 GHz [53].

Tendo em vista que o LFSR deve ser também configurável, a fim de escolher um grau e um polinômio desse determinado grau, foi descrito o LFSR programável (configurável) baseado na arquitetura de Fibonacci com *taps* variáveis [54] e ilustrado na Figura 3.5. Seu polinômio genérico é $1 + c_1x^1 + c_2x^2 + c_3x^3 \cdots + c_nx^n$ e, dessa forma, para configurar um polinômio, basta programar o registrador $[c_1 \ c_2 \ c_3 \ \cdots \ c_n]$.

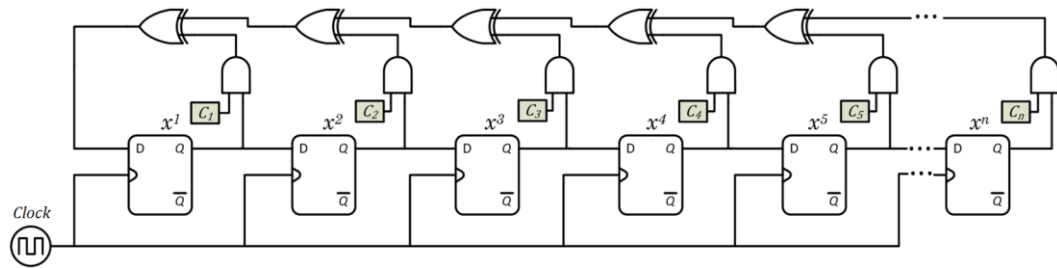


Figura 3.5: LFSR Programável (*taps* variáveis) para Arquitetura de Fibonacci

Na Figura 3.6 é possível visualizar o bloco funcional do LFSR Programável implementado em FPGA, com suas entradas e saídas, tendo: um barramento de 16 bits para configurar o polinômio ($C[15..0]$); uma entrada de clock ($CLOCK_LFSR$); uma entrada para *preset* dos flip-flops internos (início); e por fim as 17 saídas possíveis de serem utilizadas ($S00...S16$).

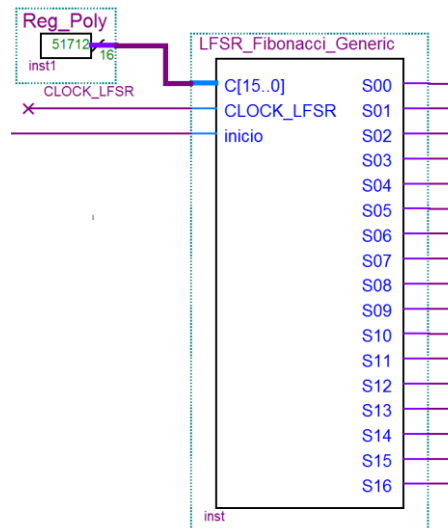


Figura 3.6: Bloco do LFSR Programável (Configurável), descrito em FPGA

É possível gerar sequências determinísticas para a matriz de medida, gravando a matriz em uma memória, contudo dois fatores devem ser analisados: 1) dependendo do tamanho desta sequência (matriz de medida) o *hardware* ocupado pela memória pode ser maior que o *hardware* do LFSR e 2) dependendo da memória escolhida, o tempo de acesso dos dados desta memória pode ser superior ao tempo do *hardware* do LFSR.

3.2.1.3 Circuito de sincronismo do *reset* (Pulso de *reset*)

O Circuito de Sincronismo do *Reset* (Pulso de *reset*) foi projetado para zerar a tensão no capacitor do circuito integrador (ou filtro passa baixa) por meio de uma chave analógica (como a 4066), dessa forma foi desenvolvido um bloco em FPGA para configurar não apenas

a frequência desse sinal, mas também o tempo que a chave analógica descarrega o capacitor. Esse tempo deve ser o menor possível e depende da constante de tempo RC , sendo C o valor da capacitância usada no circuito integrador e R o valor da resistência interna da chave analógica. Esse circuito de Sincronismo é uma versão melhorada do descrito em [55].

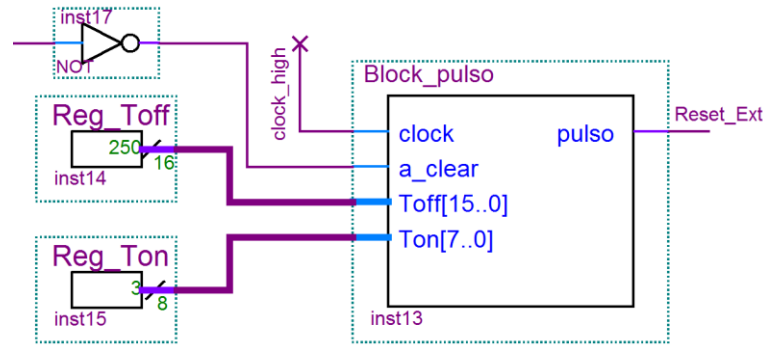


Figura 3.7: Bloco do Circuito do Sincronismo do *Reset*

Na Figura 3.7 é mostrado o diagrama de blocos deste circuito, que consiste em uma entrada para zerar os contadores internos (*a_clear*); uma entrada de relógio (*clock*); um barramento de 16 bits para configurar o tempo em que o sinal fica em nível lógico baixo (*Toff[15..0]*); um barramento de 8 bits para configurar o tempo em que o sinal fica em nível lógico alto (*Ton[7..0]*); e por fim a saída do sinal de *reset* (*Reset_Ext*).

Esse bloco é constituído de 2 contadores e 2 comparadores. Um conjunto contador comparador para determinar o nível lógico alto e o outro conjunto para determinar o nível lógico baixo.

Existe ainda um pequeno circuito extra para garantir que antes de iniciar a primeira medida o sinal de reset esteja em nível lógico alto, e assim os capacitores estejam descarregados inicialmente.

3.2.1.4 Circuito de sincronismo do DAQ (Pulso de Gatilho)

Após um determinado período, o sinal de gatilho (*trigger*) é enviado ao DAQ que toma uma amostra dos sinais na saída de cada integrador. Cada amostra (ou medida) realizada pelo DAQ só ocorre no momento em que o sinal de gatilho é ativado (nível lógico alto). O sinal de gatilho deve estar sincronizado com o sinal de *reset*, pois logo após a ocorrência do sinal de gatilho deve ocorrer o sinal de *reset*, para dar início a um novo conjunto de medidas.

Na Figura 3.8 é possível visualizar o bloco que configura e gera esse gatilho, tendo: uma entrada de *clear* assíncrono (*Aclear*) para zerar o seu contador interno no início da

primeira medida; uma entrada de *clear* síncrono (Sclear) que é acionado com o sinal de *reset* do circuito de Sincronismo do *Reset* (Sync Reset), garantindo assim o sincronismo entre os dois blocos; uma entrada de *clock* (clock) que deve ser a mesma do bloco da Figura 3.7; um barramento de 16 bits para configurar o tempo em que o gatilho deve ocorrer; e por fim uma saída para controlar o DAQ (Sync_DAQ).

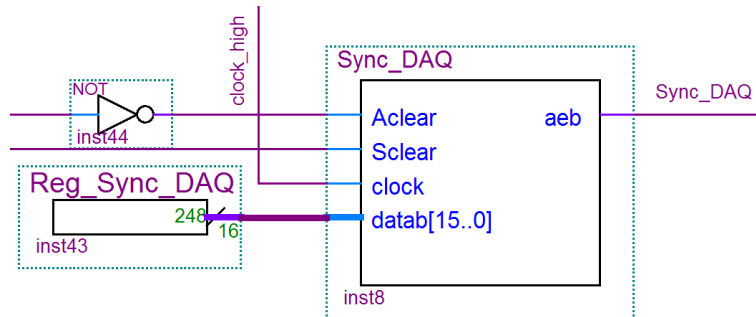


Figura 3.8: Bloco do Circuito de sincronismo do DAQ

É de extrema importância que o pulso de gatilho do DAQ ocorra um pouco antes do sinal de *reset*, por isso o registrador *Reg_Sync_DAQ* tem um valor inferior ao *Reg_Toff*.

O bloco representado na Figura 3.8 é constituído basicamente de um contador e um comparador. Esse conjunto contador comparador é usado para determinar o nível lógico alto em que o gatilho deve ocorrer.

3.2.2 AIC Hardware Analógico

O *hardware* analógico do AIC é constituído de uma Placa de Circuito Impresso (PCI) dividida em 8 canais (CH1, ..., CH8), cada canal é constituído de: um divisor resistivo com *offset* (Condicionamento de Sinal), um multiplicador de 4 quadrantes e um filtro com circuito de *reset*. Esse circuito realiza um produto interno de sinais analógicos em cada canal simultaneamente, de forma sincronizada.

Na Figura 3.9 é possível visualizar esse *hardware* em duas vistas. Uma grade de cobre aterrada (ligada no GND) foi colocada para sustentação mecânica de várias pontas de prova e para facilitar o acesso dessas pontas de prova ao GND, conforme mostrado na Figura 3.2b).

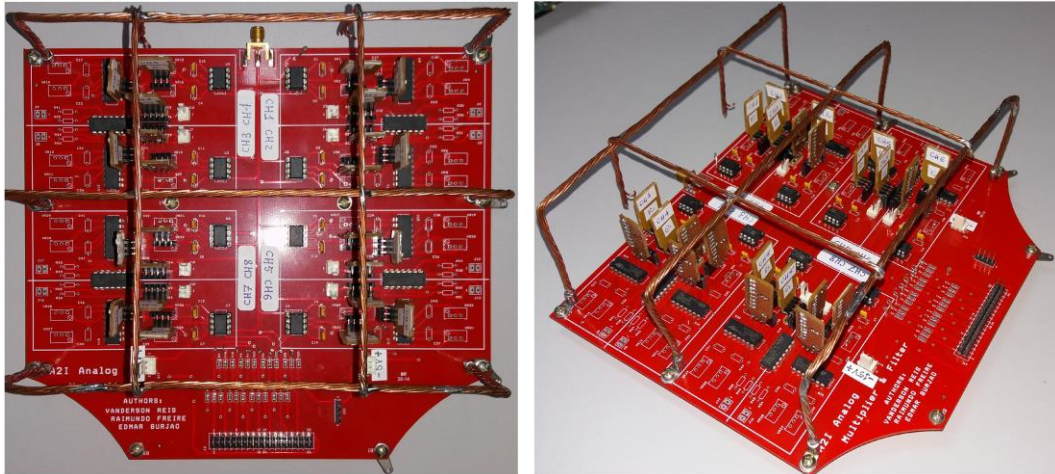


Figura 3.9: AIC Hardware Analógico, Vista de Superior (a esquerda) e Vista em Perspectiva (a direita)

3.2.2.1 Condicionamento de Sinal

Um circuito de condicionamento de sinal converte os níveis de tensão de $3,3\text{ V}$ (nível lógico alto) e 0 V (nível lógico baixo) vindos do LFSR (na FPGA) para $+1\text{ V}$ e -1 V respectivamente. Esses sinais do LFSR com os novos níveis de tensão são usados nos multiplicadores de cada canal.

O Divisor Resistivo com *Offset* de um canal é ilustrado na Figura 3.10 e faz uma atenuação no sinal com um deslocamento de nível, de forma que o seu valor máximo de tensão seja $+1\text{ V}$ e o mínimo -1 V . Pode ser visto também como um somador resistivo com pesos diferentes para cada sinal, sendo que um desses sinais é um nível DC para fazer o deslocamento (*offset*).

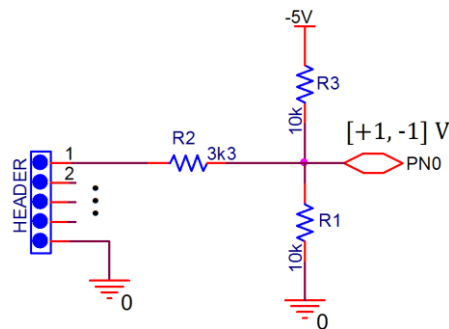


Figura 3.10: Circuito Divisor Resistivo com *Offset*

Na Equação (3.1) é representada a operação que deve ser realizada no sinal de entrada para se ter a excursão desejada de $+1\text{ V}$ a -1 V .

$$V_{PN} = \frac{V_{PRBS}}{1,65} - 1, \quad (3.1)$$

sendo V_{PRBS} o sinal na entrada e V_{PN} o sinal na saída do circuito de condicionamento.

Usando o teorema da superposição na análise do circuito da Figura 3.10 é deduzida a Equação (3.2), com tensão do sinal de saída no ponto PN (V_{PN}) em função da tensão do sinal de entrada V_{PRBS} para esse circuito.

$$V_{PN} = \frac{(R_1 \parallel R_3) \cdot V_{PRBS}}{(R_1 \parallel R_3) + R_2} + \frac{(R_1 \parallel R_2) \cdot V_{DC}}{(R_1 \parallel R_2) + R_3}. \quad (3.2)$$

Fazendo os valores dos resistores $R_1 = 10 \text{ k}\Omega$, $R_2 = 3,3 \text{ k}\Omega$ e $R_3 = 10 \text{ k}\Omega$, e com $V_{DC} = -5 \text{ V}$, tem-se um resultado muito próximo ao obtido pela Equação (3.1). Um ajuste fino pode ser feito no valor de V_{DC} se for necessário.

Na Figura 3.11 é possível visualizar o esquemático (a esquerda) e a imagem (a direita), deste circuito para os 8 canais. Os resistores são todos do tipo SMD de filme espesso (*thick film*) da série E24 com tolerância de $\pm 1\%$, esses resistores foram escolhidos por duas razões: tolerâncias e tamanho.

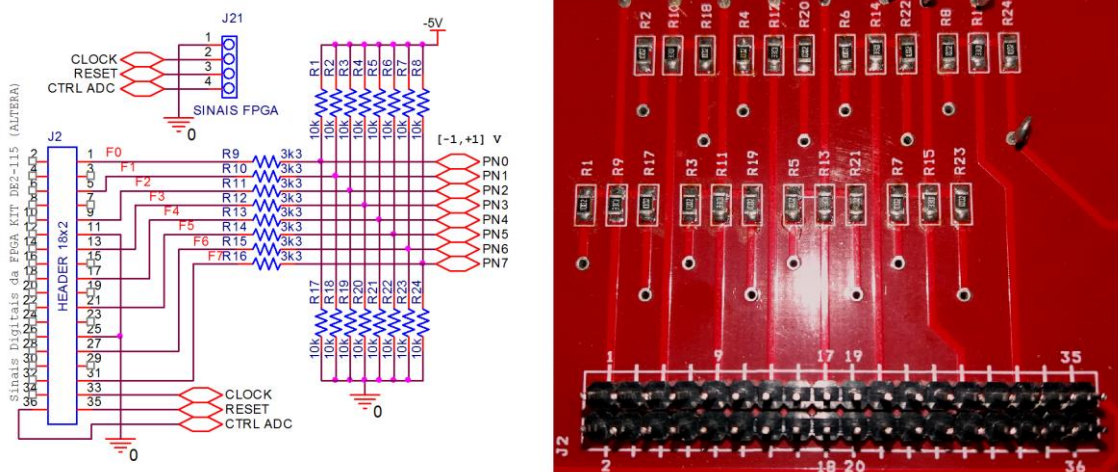


Figura 3.11: Divisor Resistivo com *Offset*, dos 8 canais: Esquemático (a esquerda) e Imagem (a direita)

É possível visualizar também na Figura 3.11 o conector J2 que, com um cabo tipo *flat*, liga eletricamente o AIC *hardware* Digital ao AIC *hardware* Analógico. Além dos 8 sinais do PRBS, também são ligados o sinal de relógio (CLOCK) no pino 33, o sinal de *reset* (RESET) no pino 35 e o sinal de gatilho (CTRL ADC) no pino 36.

3.2.2.2 Multiplicador

O circuito integrado (CI) utilizado para a função de multiplicação foi o AD633 da ANALOG DEVICES, seu circuito interno e externo é ilustrado em um esquemático na Figura 3.12 (a esquerda) e uma imagem contendo dois multiplicadores (a direita).

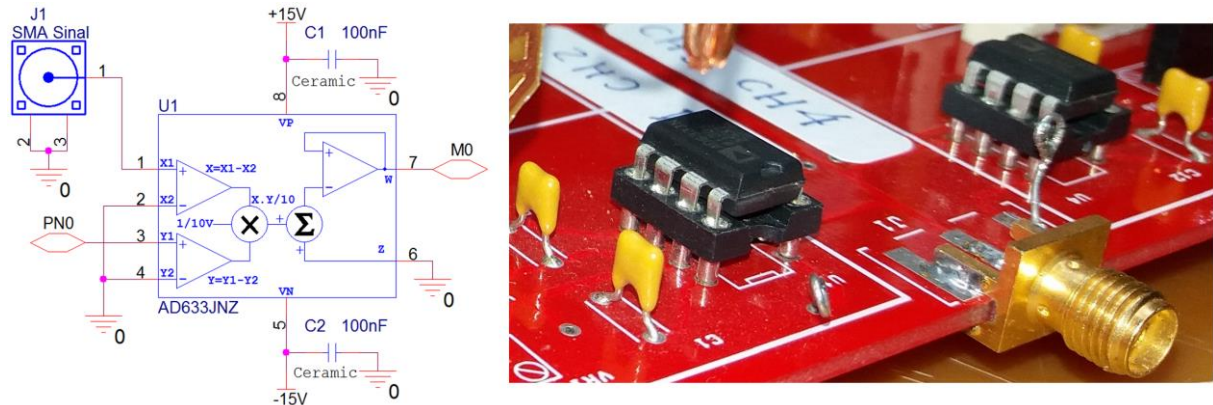


Figura 3.12: Multiplicador de 4 Quadrante AD633: Esquemático (a esquerda) Imagem na PCI (a direita)

É classificado como um multiplicador de 4 quadrantes, pois para suas duas entradas existem quatro combinações possíveis de polaridades e sua saída terá a polaridade correta para cada uma dessas combinações de entrada.

Sua função de transferência é dada pela Equação (3.3). O sinal na saída do núcleo multiplicador interno do CI é dividido por 10 V e depois entra em um núcleo somador com uma entrada chamada Z e por fim em um circuito isolador (*buffer*).

$$W = \frac{(X_1 - X_2)(Y_1 - Y_2)}{10V} + Z. \quad (3.3)$$

Caso haja necessidade, é possível usar as entradas diferenciais do multiplicador e isolar os dois sinais a serem multiplicados.

Seu erro total em fundo de escala é $\pm 2\%$ no máximo e típico de $\pm 1\%$, mas dentre outras aplicações possíveis, a entrada Z pode ser usada para fazer um ajuste fino do erro de *offset*, tendo em vista que esse tem uma grande contribuição no erro total.

3.2.2.3 Integrador (Filtro)

O circuito integrador ideal, com função de transferência igual a $1/s$, não pode ser implementado fisicamente por diversas razões, assim uma aproximação pode ser feita usando um filtro passa baixas de primeira ordem, cuja função de transferência pode ser escrita conforme Equação (3.4), sendo G o ganho do circuito e a constante de tempo $RC = 1/\omega_o$, com ω_o indicando a frequência de corte. Chamado neste texto de integrador prático.

$$\frac{V_o(s)}{V_i(s)} = \frac{G \cdot \omega_o}{s + \omega_o}. \quad (3.4)$$

Uma ilustração é mostrada na Figura 3.13 destacando as diferenças entre as respostas de um Integrador Ideal (em azul) e de um Integrador em *Hardware* (em vermelho), verificando o ganho G e a frequência de corte ω_o .

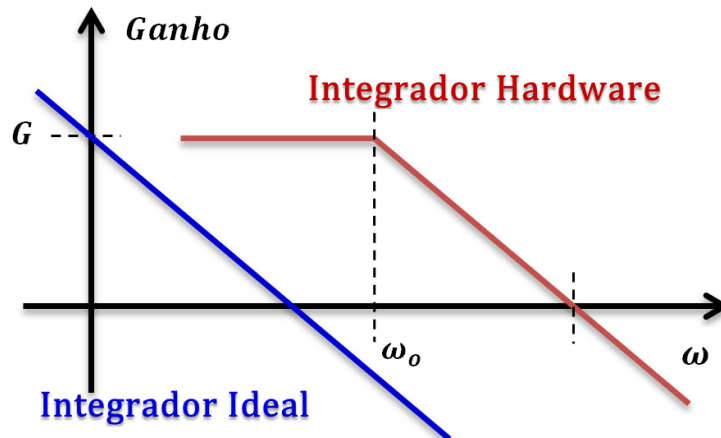


Figura 3.13: Resposta em Frequência Assintótica: Integrador Ideal(em Azul) e Integrador em *Hardware*(em vermelho)

Na Figura 3.14 pode ser visualizado o esquemático do circuito do filtro implementado. O amplificador operacional usado foi o CI TL084 e para chave analógica o CI HCF4066, utilizada para fazer o *reset* do filtro, descarregando o capacitor.

A entrada M0 é ligada na saída do multiplicador, a entrada RESET é ligada ao circuito de sincronismo do *reset* implementado na FPGA e a saída OUT_0_1 é ligada ao DAQ para fazer as medidas, havendo um circuito deste para cada canal.

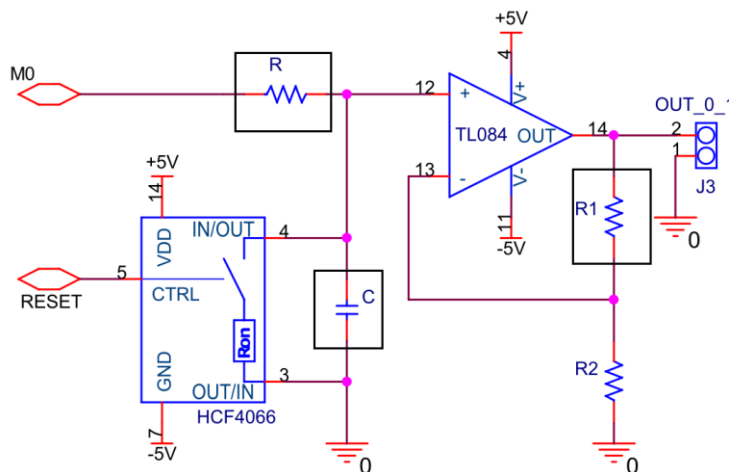


Figura 3.14: Filtro Integrador com Circuito de *Reset*

A função de transferência deste circuito é mostrada na Equação (3.5), com ganho $G = (R_1 + R_2)/R_2$ e frequência de corte $\omega_o = 1/RC$. Nota-se que o ganho e a frequência de corte são ajustados (configuradoras) de forma independente um do outro, esses dois ajustes são uns dos poucos que são feitos de forma manual, três pequenos PCI (*cards*) foram desenvolvidos para facilitar a troca de valores de R , C e R_1 , sendo R_2 fixo com valor de $1\text{ k}\Omega$.

$$\frac{V_o(s)}{V_i(s)} = \frac{R_1 + R_2}{R_2} \cdot \frac{1}{1 + RCs}. \quad (3.5)$$

Esses *cards*, bem como todo o AIC da parte analógica, podem ser visualizados na Figura 3.9, os *cards* são identificados como R, C e G para configurar os valores de R , C e R_1 respectivamente. É possível identificar também no PCB os 8 canais delimitados por retângulos (*Silkscreen*) em branco.

Dependendo das características do sinal a ser medido pode haver necessidade de trocar o valor da frequência de corte e, conseqüentemente, o valor do ganho, valores que são facilmente alterados apenas com um pequeno curto numa determinada trilha do *card*, conforme pode ser visto na Figura 3.15.

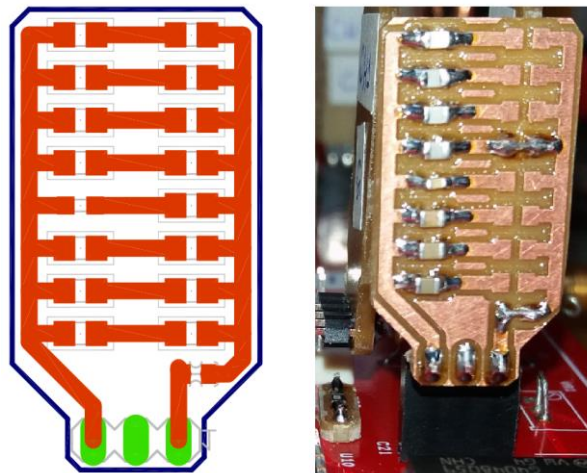


Figura 3.15: Card de configuração da Capacitância C : Leiaute (a esquerda) e Imagem (a direita)

Caso haja alguma mudança no valor do capacitor C é necessário fazer uma nova configuração no registrador Reg_Ton do circuito do sincronismo de *reset*, visto na Figura 3.7, tendo em vista que o tempo de descarga do capacitor deve ser maior que $5R_{on}C$, sendo R_{on} o valor da resistência interna da chave analógica.

3.3 Reconstrução e Ajuste em *Software*

Após as realizações das medidas, é feita a reconstrução do sinal original (na forma digital), e em seguida são realizados alguns ajustes. Esse último é feito tanto no sinal reconstruído quanto no sinal de referência, com o objetivo de validar a reconstrução de forma qualitativa e quantitativa.

As reconstruções a partir das medidas (simuladas), feitas no AIC em *Software*, são realizadas da mesma forma que as feitas a partir das medidas (ensaiadas), feitas no AIC em *Hardware*, contudo, nas medidas ensaiadas é possível fazer outras reconstruções considerando um erro ε , em virtude principalmente de ruídos e distorções inerentes do *hardware*, conforme descrito na Equação (2.30). Todas as reconstruções são feitas usando o MATLAB.

3.3.1 Reconstrução

Diversos algoritmos podem ser aplicados para a reconstrução do sinal original, a partir das amostras obtidas (medidas) usando a Amostragem Compressiva. Alguns desses algoritmos foram desenvolvidos em outros contextos, mas são perfeitamente aplicáveis ao problema em questão. Este trabalho é delimitado ao estudo e construção do *hardware* para geração das medidas e não aos algoritmos de reconstrução, contudo é de fundamental importância a utilização desses algoritmos para validar o *hardware* proposto. L1-Magic, TVAL3, Sparsify e CVX são exemplos de aplicativos usados para reconstrução, que podem ser usados.

A principal ferramenta, usada neste trabalho, para reconstrução é o pacote CVX, da empresa CVX Research, Inc [56]. O CVX realiza otimização convexa e neste trabalho é usado para fazer a busca pela minimização da norma l_1 . Usuários acadêmicos têm acesso total a esse produto para plataforma MATLAB sem nenhum custo. É possível ter acesso ao CVX para outras plataformas computacionais além do MATLAB como R, Python e C++, essas últimas são para fins comerciais e licenciados (têm custo). Um Guia de Usuário do CVX pode ser encontrado em [13].

Foi usado também o *Orthogonal Matching Pursuit* (OMP), que é um algoritmo utilizado para reconstruir um sinal no seu domínio esparso. O OMP utiliza a matriz de medida

Θ e o vetor \mathbf{y} de medidas como entradas e fornece uma estimativa $\tilde{\mathbf{s}}$ do sinal original \mathbf{s} . A cada iteração do algoritmo, uma das colunas de Θ é escolhida (necessariamente a que é mais correlacionada com os resíduos das medidas \mathbf{y}) e então remove-se a contribuição desta coluna para calcular um novo resíduo. Depois de M iterações, o algoritmo retorna a estimativa final do sinal original, uma descrição mais detalhada pode ser encontrada em [57]. Em particular, é possível definir para o OMP informações adicionais como um determinado resíduo limite ou a esparsidade do sinal.

Embora o desenvolvimento dos algoritmos de reconstrução não faça parte deste trabalho, seu uso é de fundamental importância para validar o AIC proposto. Assim é feita uma descrição da reconstrução com etapas anteriores e posteriores ao uso destes algoritmos. Essa descrição é feita separadamente para medidas feitas com sinais no domínio esparso e medidas feitas com sinais no domínio não esparso.

Na Figura 3.16 é mostrado o fluxograma de reconstrução para medidas feitas no domínio não esparso. Inicialmente são feitas as medidas $\mathbf{y} = \Phi\mathbf{x}$. Essa etapa obviamente não faz parte da reconstrução e foi colocada no fluxograma apenas para enfatizar o domínio não esparso em que são feitas as medidas. Cabe destacar que não importa se as medidas \mathbf{y} são oriundas de uma multiplicação simples de uma matriz Φ por um vetor \mathbf{x} ou de um AIC em *software* ou de um AIC em *hardware*. Em princípio, as etapas de reconstrução podem ser as mesmas. Apenas dois dados são fornecidos como entrada para a reconstrução, a matriz de medida Φ e o vetor de medidas \mathbf{y} , no caso deste trabalho, \mathbf{y} tem sempre 8 valores reais, e Φ tem sempre 8 linhas, pois o AIC utilizado possui 8 canais, sendo possível utilizar menos de 8 valores de \mathbf{y} . Como foram usados sinais tonais, cujo domínio de esparsidade é a frequência, a próxima etapa é construir a matriz da transformada discreta de Fourier (DFT) ou mesmo a sua inversa (IDFT) Ψ^{-1} , essa matriz é para fazer com que a busca ocorra no domínio esparso.

Usando o CVX é possível fazer dois tipos de busca, uma semelhante à Equação (2.26) e outra à Equação (2.30), este último sujeita a um determinado erro ε , sendo mais indicada para medidas realizadas no AIC em *Hardware*. Observa-se que Θ foi substituído por $\Phi\Psi^{-1}$ de acordo com a Equação (2.14). O resultado da busca foi um sinal $\tilde{\mathbf{s}}$ no domínio esparso (domínio espectral) e para verificar o sinal $\tilde{\mathbf{x}}$ no domínio não esparso é necessário usar novamente Ψ^{-1} . Por fim, para uma validação quantitativa da reconstrução (no caso o MSE) é feita uma sequência de ajustes de ganho para $\tilde{\mathbf{x}}$ e de fase para o sinal de referência \mathbf{x} , esses ajustes e condicionamentos são descritos na próxima subseção. A implementação do

algoritmo OMP utilizado não forneceu bons resultados para essas medidas \mathbf{y} , devido a não trabalhar com números complexos, por isso não foi usado neste fluxograma.

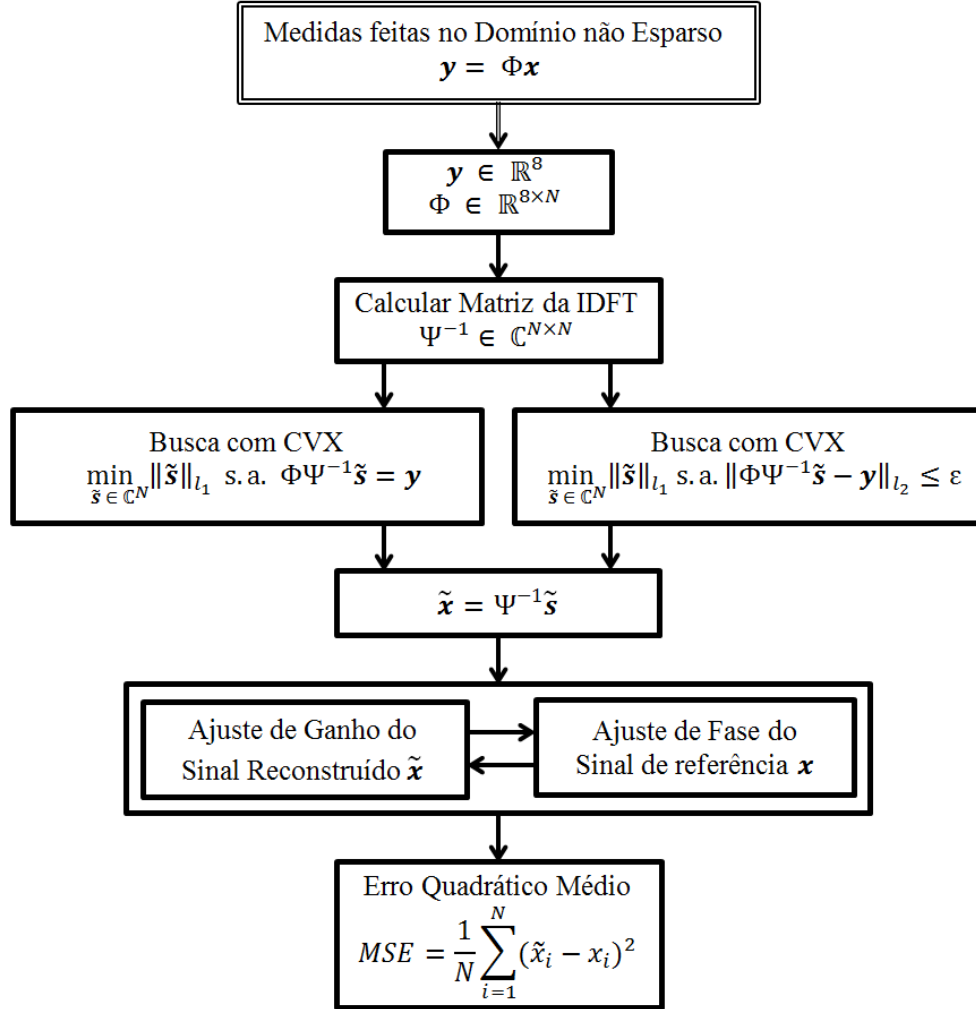


Figura 3.16: Fluxograma para Reconstrução usando o CVX de Medidas Feitas no Domínio Não Esparso

Na Figura 3.17 é mostrado o fluxograma de reconstrução para medidas feitas no domínio esparso. Depois de realizadas as medidas $\mathbf{y} = \Theta \mathbf{s}$, dois dados são fornecidos como entrada para reconstrução: a matriz de medida Θ e o vetor de medidas \mathbf{y} . Para esse tipo de sinal foi utilizado tanto o CVX como o OMP, sendo que o uso do CVX é muito semelhante ao descrito no fluxograma anterior. Como as medidas foram feitas no domínio esparso do sinal, não é necessário nenhum tipo de transformada, pois a busca sempre é feita no domínio esparso. Assim, as reconstruções com o CVX neste fluxograma são idênticas às Equações (2.26) e (2.30). Contudo, é possível realizar a busca do sinal \mathbf{s} também com o OMP, sendo que o algoritmo usado para implementar o OMP exige valor da esparsidade, como mostrado no fluxograma da Figura 3.17. Por fim, é feito um condicionamento no sinal reconstruído $\tilde{\mathbf{s}}$

para avaliar a reconstrução usando o MSE. O condicionamento consistem em simplesmente multiplicar todos os elementos do sinal \tilde{s} , pelo maior quociente, elemento a elemento, do sinal de referência (original) s e do sinal reconstruído \tilde{s} . Como resultado, pelo menos dois elementos, sendo um elemento de mesma posição de cada vetor, vão ficar iguais e assim é possível mensurar a diferença entre os demais. No caso ideal, os dois vetores serão iguais, i.e. $s[n] = \tilde{s}[n]$.

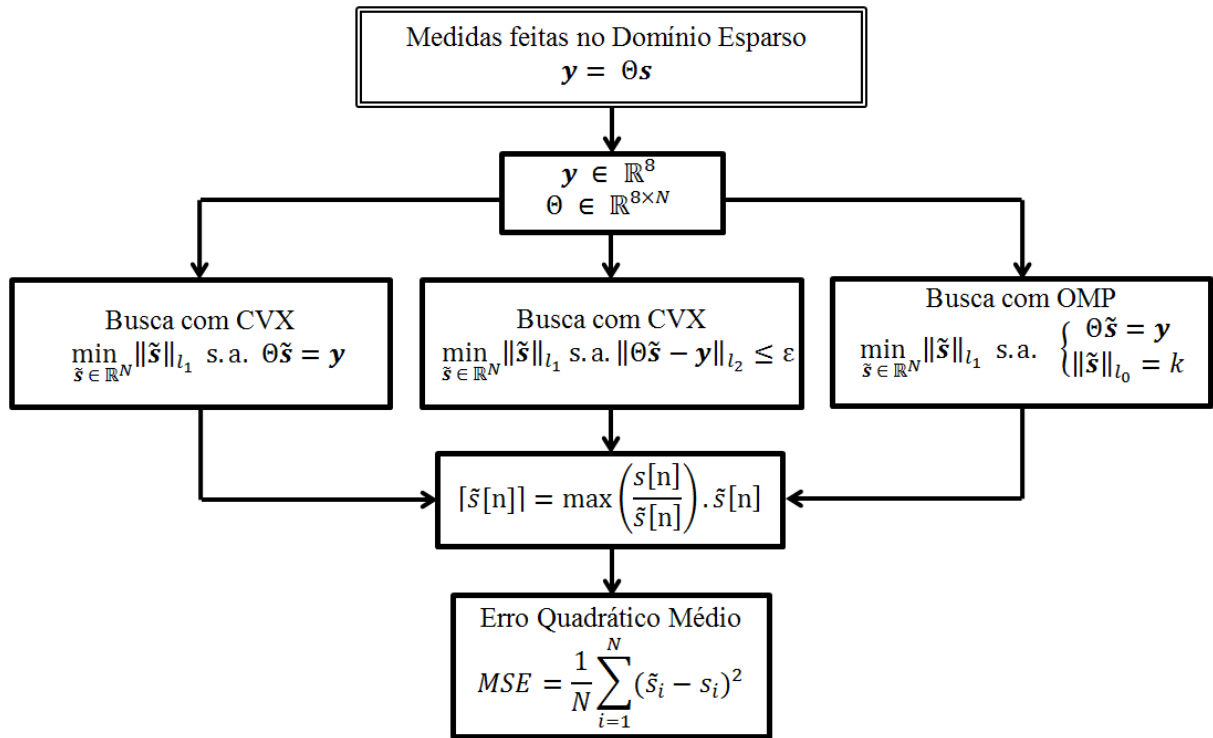


Figura 3.17: Fluxograma para Reconstrução usando o CVX e o OMP de Medidas Feitas no Domínio Esparso

3.3.2 Ajustes e Condicionamento de Sinal.

No fluxograma da Figura 3.16 são apresentados, sem muitos detalhes, dois blocos de ajuste e condicionamento de sinal, chamados de Ajuste de Ganho e Ajuste de Fase. Nesta subseção é detalhado como essas duas etapas funcionam e são importantes para uma boa avaliação quantitativa do sinal \tilde{x} reconstruído.

Para poder comparar, por meio de alguma métrica, o sinal reconstruído \tilde{x} (ou $\tilde{x}[n]$), é necessário criar um sinal discreto de $x(t)$, chamado de sinal de referência (ou modelo, ou original ou ideal) e denotado por $x[n]$ ou $\hat{x}[n]$. Esse sinal possui o mesmo número de amostras de $\tilde{x}[n]$ e amplitude contínua, para evitar erros de quantização. Na Figura 3.18 é possível visualizar um exemplo desses sinais, em: a) $x(t)$ com três componentes espectrais de

1, 6 e 8 kHz com amplitudes de 6, 8 e 3 V, respectivamente; b) $\tilde{x}[n]$ reconstruído, inicialmente sem nenhum ajuste; e c) o sinal modelo $x[n]$ (em azul) junto com o mesmo sinal $\tilde{x}[n]$ (em vermelho).

As amplitudes das amostras de $\tilde{x}[n]$ são muito inferiores às de $x[n]$, devido a atenuação de 20 dB/dec do integrador (filtro) nas medidas $y[n]$. Neste caso, com um MSE de 54,4974 para um ganho de 1, um ajuste de ganho (ou normalização) se faz necessário.

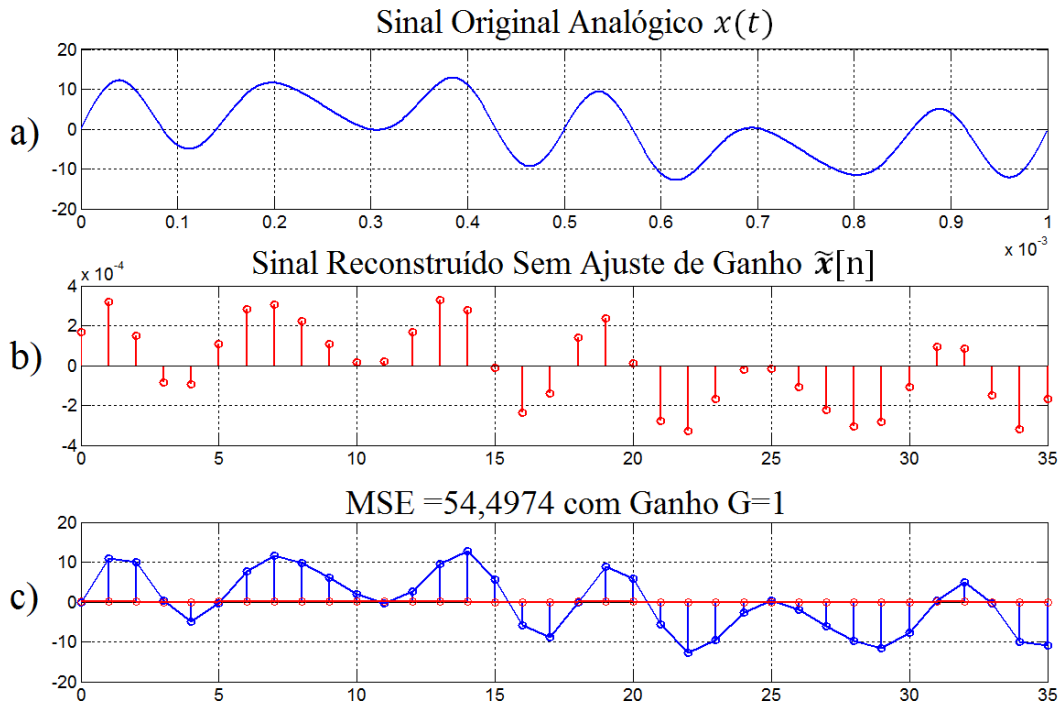


Figura 3.18: Comparativo dos sinais $x(t)$, $\tilde{x}[n]$ e $x[n]$, sem ajustes de ganho e fase

O sinal modelo $x[n]$ é determinado inicialmente pela multiplicação de um trem de impulsos $s(t)$ pelo sinal $x(t)$ e discretizado com período $T = 1/f_{PRBS}$ e amplitude contínua, i.e. sem quantização, conforme ilustrado na Figura 3.19.

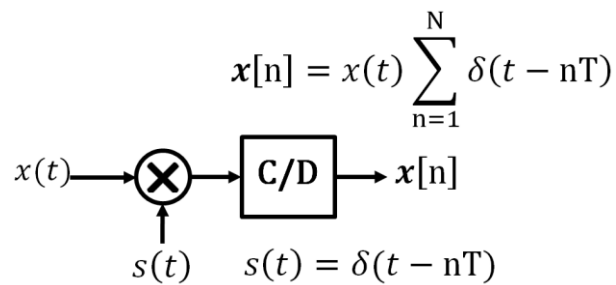


Figura 3.19: Construção do sinal modelo $x[n]$

Com o objetivo de determinar o ganho ótimo G , para um sinal $x(t)$ específico, é feito um pequeno incremento ΔG em $\tilde{x}[n]$ e em seguida calculado o MSE de $\Delta G \cdot \tilde{x}[n]$ e $x[n]$. Esses passos são repetidos até encontrar o valor de G para o menor MSE. Os incrementos poderiam ser realizados com valores menores, contudo verificou-se que após um determinado valor a melhoria no MSE é mínima. Na Figura 3.20 é mostrado um bloco da função Busca Ganho com os dados de entrada e saída.

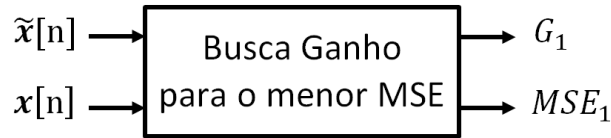


Figura 3.20: Dados de Entrada e Saída da Função Busca Ganho

O sinal $\tilde{x}[n]$ reconstruído pelo CVX possui um deslocamento no tempo Δt nas suas amostras (aqui chamado de fase). Contudo as amostras do sinal modelo $x[n]$ sempre começam em $\Delta t = 0$, i.e. o $x[1] = x(t = 0)$, assim esse sinal não é um bom modelo para validar $\tilde{x}[n]$. A solução proposta é calcular o valor Δt de $G_1 \tilde{x}[1]$, conforme ilustrado na Figura 3.21.

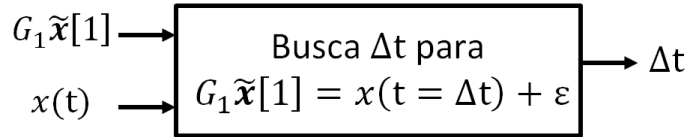


Figura 3.21: Dados de Entrada e Saída da Função busca Δt com um erro ϵ

Conhecendo-se Δt é feito um novo sinal modelo $x[n]$, com $x[1] = x(t = \Delta t)$, denotado agora de $\hat{x}[n]$. A construção desse novo sinal modelo $\hat{x}[n]$ é ilustrado na Figura 3.22.

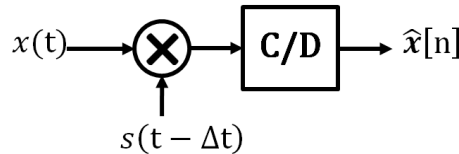


Figura 3.22: Construção do Novo Sinal Modelo $\hat{x}[n]$

A diferença dos sinais modelos $x[n]$ e $\hat{x}[n]$ ocorre apenas no momento em que são feitas as amostras em $x(t)$, chamado de ajuste de fase (Δt) e ilustrada na Figura 3.23.

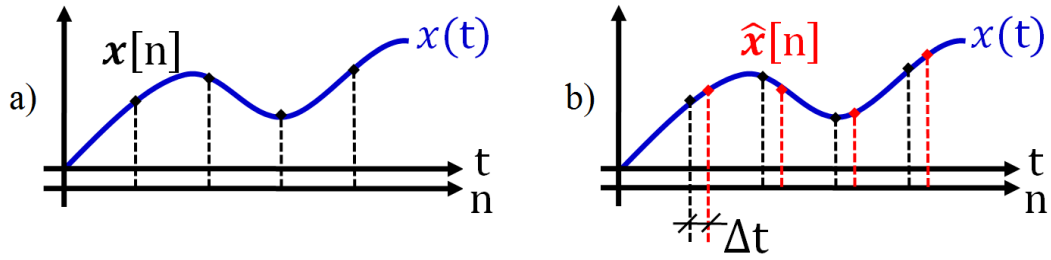


Figura 3.23: Sinal Modelo: a) $x[n]$ sem ajuste de fase b) $\hat{x}[n]$ com ajuste de fase

Por fim, é feita pela segunda vez a busca por G com o mesmo algoritmo (função) da Figura 3.20, sendo que agora as entradas são $G_1\tilde{x}[n]$ e o novo sinal modelo $\hat{x}[n]$. Essa nova busca é ilustrada na Figura 3.24, retornando G_2 como um ajuste fino do ganho.

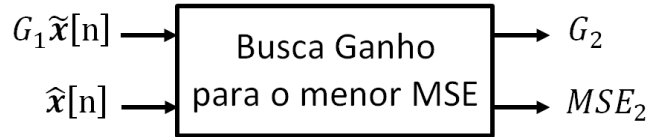


Figura 3.24: Novos Dados de Entrada e Saída da Função Busca Ganho

Assim, o ganho final é $G = G_1 + G_2$, com MSE final igual ao MSE_2 , o valor desse erro agora depende das configurações do AIC em *Software*. Para o AIC em *Hardware*, além das configurações, há também suas imperfeições, que provocam ruído e distorções no sinal processado de forma analógica.

É importante destacar que a busca por Δt deve ocorrer após o primeiro ajuste de ganho, tendo em vista que as amplitudes das amostras de $\tilde{x}[n]$ são tão pequenas que um ajuste inicial de fase (Δt) seria desprezível.

Para efeito de comparação é feita a reconstrução com esses ajustes propostos, e.g. usando o sinal $\tilde{x}[n]$, visto na Figura 3.18b, é feito o primeiro ajuste de ganho com valor $G_1 = 33596$ e $MSE_1 = 10,1295$, ilustrado na Figura 3.25a. Em um comparativo visual pode ser constatado claramente que os sinais estão defasados, sendo $G_1\tilde{x}[n]$ (em vermelho) o sinal reconstruído com o primeiro ajuste de ganho e $x[n]$ (em azul) o sinal modelo inicial. Neste exemplo foi usado um incremento grosseiro do ganho, $\Delta G = 1$.

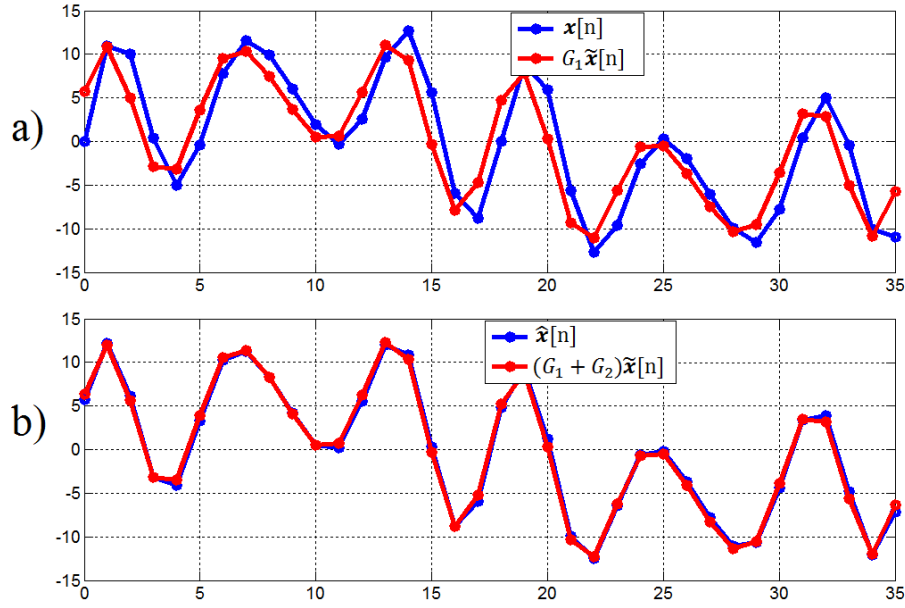


Figura 3.25: Comparativo dos Sinais Antes e Depois do Ajuste de Fase (Δt).

Após o ajuste de fase (Δt) e usando o novo sinal modelo $\hat{x}[n]$, tem-se $G_2 = 3566$ (ganho fino) e assim um ganho total de $G = 37162$ ($G_1 + G_2$). O $MSE_2 = 0.20826$ é o MSE total (ou final). Em uma visualização na Figura 3.25b pode ser constatado que a defasagem foi corrigida entre os sinais $(G_1 + G_2)\hat{x}[n]$ (em vermelho) e $\hat{x}[n]$ (em azul). É possível constatar também que a única mudança em $\tilde{x}[n]$ foi o ganho, já em $x[n]$ não houve mudança, pois $\hat{x}[n]$ é outro sinal modelo baseado no ajuste de fase Δt .

É certo que mudanças de parâmetros na configuração do AIC vão influenciar diretamente no MSE final, neste exemplo foi usado uma matriz de medida $\Phi = [\varphi_{ij}]_{8 \times 36}$, i.e. com 8 canais e 36 amostras.

3.4 AIC em Software

As simulações de partes específicas do *hardware* foram feitas em diversas ferramentas, tais como: OrCAD, Quartus II, ModelSim e Simulink. Contudo, as simulações feitas do sistema foram realizadas no ambiente Simulink integrado ao MATLAB. O AIC em *Software* é um simulador do AIC em *Hardware*, modelado em diagrama de blocos hierárquicos parametrizado no Simulink e controlado pelo MATLAB por meio dos chamados arquivos m (*m-files*), esse simulador é descrito nesta seção.

Existem algumas motivações que justificam fazer a modelagem do comportamento dinâmico do *hardware* do AIC, e em especial no Simulink: 1) uma vez que a reconstrução do

sinal é feita no MATLAB, fica simples interagir a simulação do *hardware* e reconstrução do sinal, pois o MATLAB e o Simulink podem operar conjuntamente; 2) facilitar a configuração dos diversos parâmetros do *hardware* do AIC, baseado nos resultados simulados; e 3) testar várias situações hipotéticas com relativa facilidade e, dependendo do resultado (ou das conclusões), decidir a relevância ou viabilidade de realizar fisicamente no AIC em *Hardware*.

É possível visualizar o modelo Completo no Simulink do AIC em *Software* no Apêndice A. Esse modelo é todo parametrizado por conjunto de variáveis internas a esses blocos que são configuradas externamente por meio de *scripts* do MATLAB.

Da mesma forma que foi descrita as partes do AIC em *Hardware*, é feita uma descrição das partes do AIC simulado.

3.4.1.1 Gerador PRBS (LFSR)

O LFSR modelado no Simulink e apresentado na Figura 3.26, é o mesmo que foi implementado em FPGA e visto na Figura 3.5, seu polinômio primitivo é configurado por uma variável chamada *primitive_polynomial*, nos blocos delay a frequência é configurada pela variável *freq_prbs*. Os blocos *delay* fazem o papel que os *flip-flops* tipo D fazem no LFSR em *hardware*, por isso são nomeados aqui como FFD.

O valor da variável *freq_prbs* contribui para determinar o número de colunas da matriz de medida.

É importante destacar que embora o *hardware* do AIC possa usar até 8 canais, o AIC simulado pode usar até 16, o número de canais é um parâmetro configurável e determinado pelas características do sinal a ser medido.

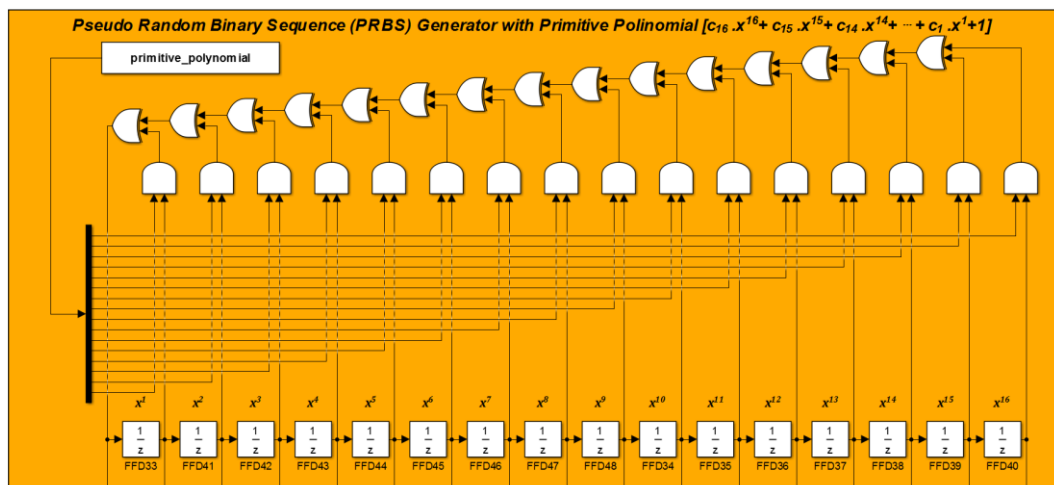


Figura 3.26: LFSR Configurável, no Modelo Simulink

3.4.1.2 Condicionamento de Sinal

O bloco de condicionamento no Simulink guarda alguma similaridade com o somador resistivo do AIC em *hardware*. O sinal de entrada oriundo do LFSR é do tipo lógico de valores 0 e 1. Após ser subtraído do valor 0,5 ele é convertido para o tipo escalar com valores de -0,5 e +0,5. Por fim esse sinal é enviado para o bloco Sign que gera saída 1 para entrada positiva, -1 para entrada negativa e 0 para entrada 0. Na Figura 3.27 é possível ver em detalhes 4 canais deste Condicionador de Sinal.

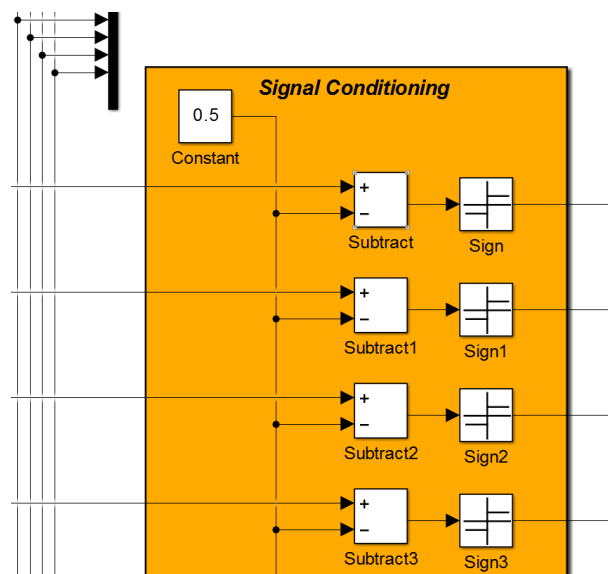


Figura 3.27: Condicionamento dos Sinais do LFSR, no Modelo Simulink

3.4.1.3 Multiplicador

O multiplicador foi implementado usando o bloco Product, que realiza uma multiplicação em 4 quadrantes, contudo para deixar esse multiplicador mais fiel ao multiplicador do CI AD633, foi acrescentado um bloco Gain e configurado para o valor de 0,1 que representa exatamente o valor que é atenuado no CI usado no AIC em *Hardware*. Na Figura 3.28 é possível ver 4 canais deste multiplicador.

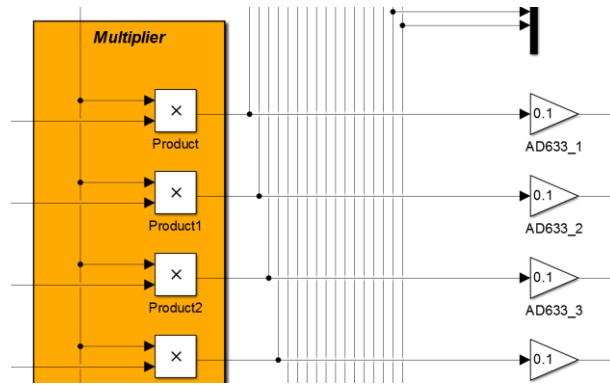


Figura 3.28: Multiplicador no Simulink, Equivalente ao CI AD633

Observa-se que todos os multiplicadores possuem uma entrada em comum, essa entrada é o sinal a ser medido. Esse sinal é definido em um bloco à parte e configurado também em um *script* do MATLAB (*m-file*).

3.4.1.4 Integrador (Filtro)

O filtro também foi modelado para ter o mesmo comportamento dinâmico que o filtro do AIC em *hardware*. Na Figura 3.29 é possível visualizar 4 canais deste bloco e à direita da figura sua construção interna.

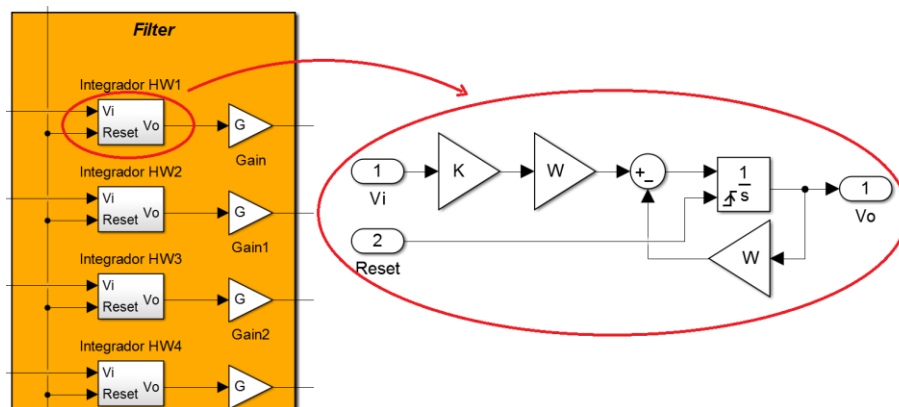


Figura 3.29: Filtro Modelado no Simulink

As variáveis K e W são o ganho e a frequência de corte do filtro, respectivamente, e são determinadas usando-se um *script* do MATLAB, exatamente pelos valores dos resistores e capacitores implementados nos *cards*, como visto na Figura 3.15. Essas variáveis são calculadas de acordo com a Equação (3.5).

Por fim, um bloco de ganho externo é usado para algum ajuste de ganho externo ao filtro. É possível ver uma descrição mais detalhada desse filtro bem como alguns resultados preliminares em [58].

3.4.1.5 ADC

O modelo do ADC pode ser visto na Figura 3.30, esse bloco foi modelado para ser o mais flexível possível, muito semelhante ao que é feito no DAQ (NI DAQ USB 6361). Em relação ao tempo de amostra (*Sample Time*), os blocos internos são ativados por um gatilho (*trigger*) externo de tipo borda de subida (*rising*). A amostragem e retenção (*Sample-and-Hold*, S/H) é de ordem zero (ZOH), o intervalo de quantização do bloco Quantizer é determinado externamente por $(V_{max} - V_{min})/(2^{n_{bits}})$ e por fim no bloco Saturation o limite superior é a variável V_{max} e o limite inferior é a variável V_{min} , as mesmas usadas no bloco Quantizer.

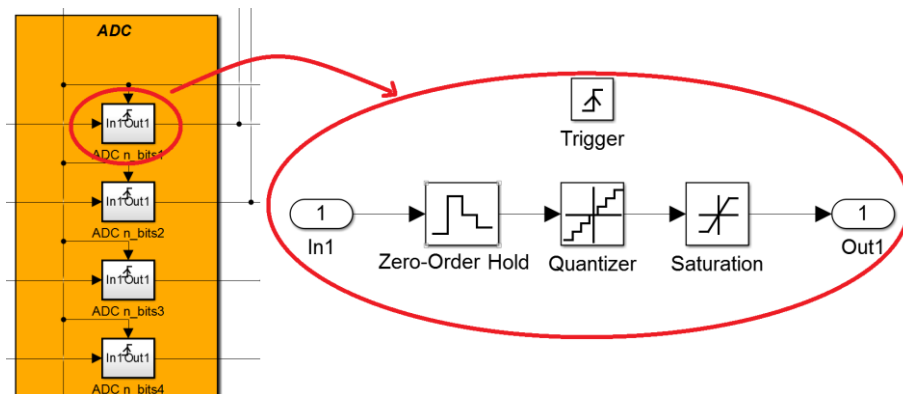


Figura 3.30: ADC (DAQ) Modelado no Simulink

Esse gatilho (*trigger*) externo é gerado por um bloco chamado Sync. Reset and Trigger, que por meio da variável t_{reset} no bloco Pulse Generator determina o momento de ativação pelas características do sinal a ser medido. Esse mesmo sinal de gatilho com um pequeno atraso (*delay*) é responsável por ativar o *reset* no bloco Filter.

O resultado é então passado para o Workspace por meio do bloco To Workspace, que recebe o mesmo sinal de gatilho. Com esse resultado é possível fazer a reconstrução do sinal medido.

4 RESULTADOS

Os resultados do trabalho estão apresentados e discutidos, separadamente, para duas classes de sinais, um em cada seção. Essas classes são de sinais esparsos na frequência e de sinais esparsos no tempo, desta forma as aquisições são realizadas no domínio não esperso e no domínio esperso, respectivamente. É feito um comparativo dos resultados das aquisições (simuladas) realizadas no simulador do AIC (MATLAB/Simulink) e dos resultados das aquisições (experimentais) realizadas no AIC em *hardware*.

Um sinal de referência para avaliar as reconstruções feitas com as aquisições do AIC, poderia ser formado pelas amostras de um ADC, considerando-se não apenas o erro de quantização, mas todas as suas imperfeições, como as mencionadas na Seção 2.1, os resultados ficariam específicos a um determinado ADC, e não genéricos, como desejado.

Uma comparação mais adequada dos sinais reconstruídos é feita com os valores obtidos de uma amostragem, que não possui quantização, exceto pelo padrão IEEE[®] 754-2008 para números de ponto flutuante de dupla precisão (no MATLAB esse padrão é chamado *double*), dessa forma, construindo um sinal de referência mais adequado. Embora seja possível utilizar números com ponto flutuante de dupla precisão estendida, verificou-se pelas reconstruções realizadas não haver necessidade.

4.1 Medidas no Domínio Não Esperso

Os sinais usados nesta seção são denominados de multitonais e são compostos de uma soma de senóides, ou tons, cada um com sua amplitude, frequência e fase. São analisados os casos com um, dois e três tons, e para efeito de comparação as configurações usadas foram as mesmas, quando possíveis. No Quadro 4.1 são descritas essas configurações gerais.

Parâmetros Gerais	AIC <i>Software e Hardware</i>
Polinômio para LFSR	$x^5 + x^3 + x^2 + x + 1$
Ordem da Matriz de Medida	8 x 32
Frequência de corte do filtro	$\approx 50\text{Hz}$
Número de bits do ADC	16
Taxa de amostragem do ADC por canal	1ksps

Quadro 4.1: Configurações Gerais para Realização de Medidas no AIC.

As reconstruções foram feitas usando o programa CVX no MATLAB, com a mesma configuração para as medidas simuladas (AIC SW) e ensaiadas (AIC HW). Contudo, *a posteriori*, algumas mudanças podem ser feitas no CVX para melhorar a reconstrução, i.e. uma calibração do erro.

4.1.1 Sinal com um tom

Inicialmente foram adquiridos resultados de simulação a partir de uma senóide com amplitude de 1 V, frequência de 1 kHz e fase zero. Cabe ressaltar que foram feitas 8 aquisições e reconstruídas 32 amostras, pois a matriz de medida é $\Phi_{8 \times 32}$, conforme configuração geral do Quadro 4.1.

Alguns sinais no simulador do AIC podem ser visualizados na Figura 4.1, sendo que em: a) é possível ver 4 sinais, o sinal original (em verde), o sinal de relógio (*clock*) do LFSR (em amarelo), o sinal de saída do LFSR do canal 1 (em azul), e o sinal na saída do multiplicador também do canal 1 (em vermelho); b) o sinal acumulador na saída do filtro do canal 1; c) todos os sinais digitais dos 8 canais na saída do LFSR; e d) todos os sinais acumuladores na saída dos 8 filtros.

Alguns sinais da Figura 4.1a) e Figura 4.1c) foram deslocados na vertical apenas para uma melhor visualização, evitando algumas sobreposições indesejadas.

É certo que com esse polinômio não é necessário fazer 8 aquisições, apenas 3 ($\Phi_{3 \times 32}$) são suficientes para uma ótima reconstrução do sinal, contudo é desejável ter uma configuração mais geral, que seja possível abranger uma maior variedade de sinais, nesse caso para sinais com mais de um tom.

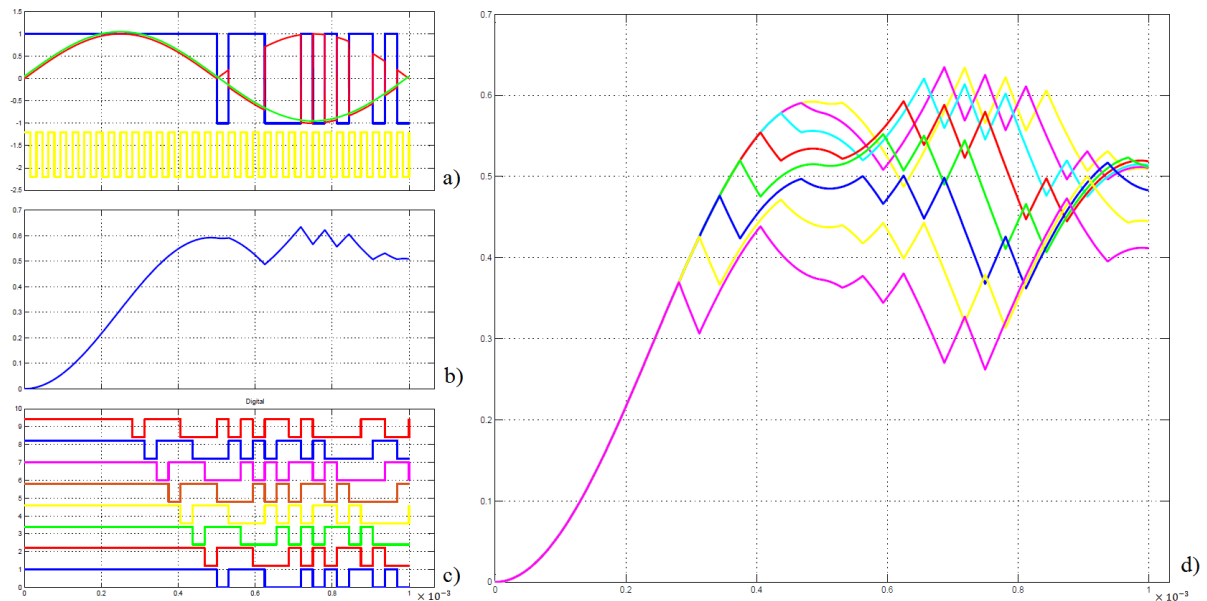


Figura 4.1: Sinais Internos do AIC em *Software* (Simulado) para 1 tom:
a) Sinais Referentes ao Canal 1, b) Na Saída do Filtro do Canal 1, c) Nas Saídas do LFSR e d) Na saída dos 8 filtros.

Como as aquisições foram realizadas no domínio não esparso e a reconstrução é feita no domínio esparso, é necessário transformar os dados dessas aquisições para um domínio esparso, neste caso o domínio da frequência. Assim é usada a matriz da transformada discreta de Fourier Ψ e a reconstrução pode ser descrita de acordo com:

$$\min_{\tilde{\mathbf{s}} \in \mathbb{C}^{32}} \|\tilde{\mathbf{s}}\|_{l_1} \text{ sujeito a } \Phi\Psi\mathbf{s} = \mathbf{y} \in \mathbb{R}^8. \quad (4.1)$$

Como o sinal $\tilde{\mathbf{s}}[n]$ está no domínio da frequência, foi usada novamente a transformada de Fourier para se ter o sinal no domínio do tempo discreto $\tilde{\mathbf{x}}[n]$. Na Figura 4.2 é mostrado o sinal reconstruído, tanto no domínio do tempo quanto no domínio da frequência.

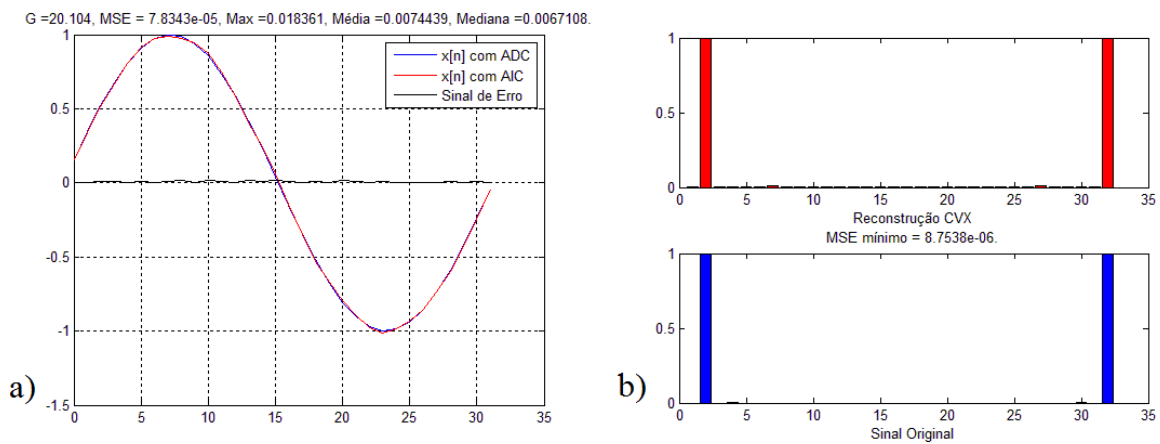


Figura 4.2: Reconstrução com CVX das Medidas do AIC em *Software* (Simulado) para 1 tom:
a) Domínio do Tempo e b) Domínio da Frequência.
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

Algumas métricas para avaliar a reconstrução podem ser vistas na figura 4.2, sendo a principal o MSE. Para ter um sinal de referência e poder avaliar o sinal reconstruído, devem haver alguns ajustes no sinal reconstruído e no sinal de referência, que são os ajustes de ganho e os ajustes de fase, respectivamente. Em seguida, devem-se realizar os cálculos dessas métricas.

Em seguida são feitas as medidas usando o AIC em *hardware* de uma senóide com amplitude de 14 Vpp e frequência de 1 kHz. O valor da amplitude do sinal original deve ser o maior possível, i.e. apenas para melhorar a Relação Sinal Ruído (SNR), sendo esse ruído inerente do sistema como um todo. Sobre certas condições, uma maior SNR implica em um menor MSE.

Na Figura 4.3 é possível verificar vários sinais semelhantes aos vistos na Figura 4.1, sendo que neste caso, esses sinais foram capturados e gravados em 2 osciloscópios digitais, um da empresa Agilent (atual Keysight) e o outra da Tektronix, ambos com 4 entradas

analógicas e 16 entradas digitais. São destacados em seguida os sinais que não foram vistos na Figura 4.1, a saber: em a) o sinal de gatilho do gerador para o *hardware* na FPGA (em laranja), o sinal de *reset* dos filtros analógicos (em azul), o sinal de gatilho do *hardware* na FPGA (AIC) para o DAQ (ADC) (em rosa), os outros já foram descritos e em b) o sinal de *reset* (em verde), o mesmo do outro osciloscópio (em azul), nesse último são vistos os sinais referentes ao canal 1.

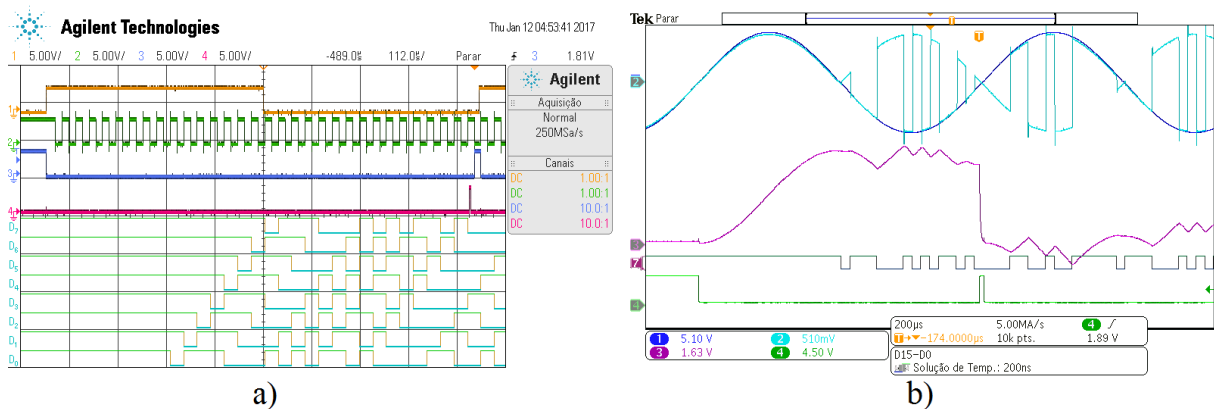


Figura 4.3: Sinais Internos do AIC em *Hardware* para 1 tom :
a) Sinais de Controle (dentre outros) e b) Sinais do Canal 1.

Algumas descrições importantes: 1) Antes da primeira borda de subida do sinal de gatilho do gerador, os filtros estão sendo todo tempo zerados (sinal de *reset* ativo) e não há sinal de relógio (*clock*), por consequência os sinais digitais nas saídas do LFSR estão sempre em nível lógico “um” (em *preset*); 2) os sinais na saída dos filtros também são zerados (em *reset*) após 32 pulsos do sinal do relógio (*clock*), pois a matriz de medida é da ordem de 8×32 ; e 3) o sinal de gatilho do AIC para o DAQ (ADC) ocorre sempre um pouco antes do sinal de *reset* ser ativado após os 32 pulsos do sinal do relógio (*clock*).

A reconstrução dessas medidas feitas no AIC em *hardware* foi realizada da mesma forma que foram feitas nas medidas no AIC em *software*, conforme a Equação (4.1). Os resultados dessa reconstrução podem ser vistos na Figura 4.4. E assim, podem ser comparados com os resultados simulados vistos na Figura 4.2.

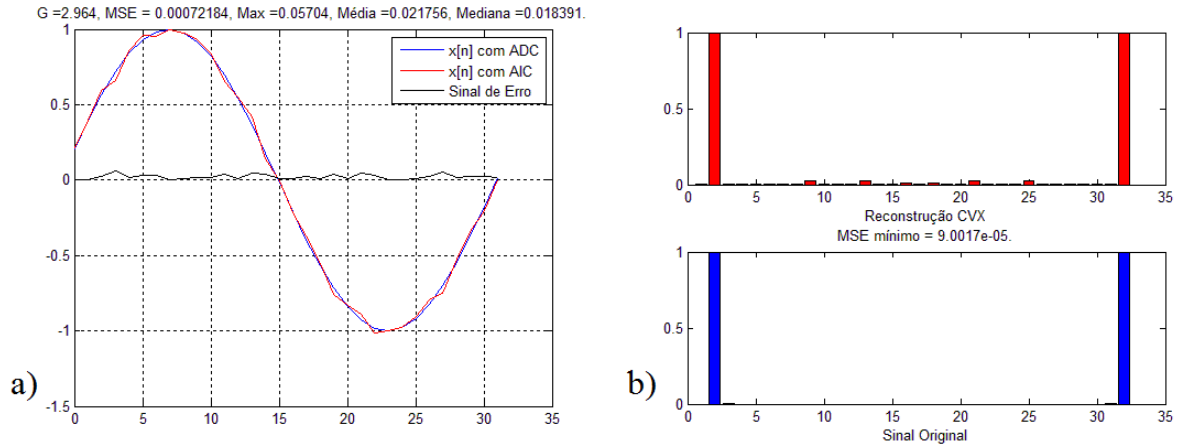


Figura 4.4: Reconstrução com CVX das Medidas do AIC em *Hardware* para 1 tom:
a) Domínio do Tempo e b) Domínio da Frequência:
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

Embora a reconstrução tenha ocorrido com sucesso, seu erro ficou maior. É possível atribuir esse aumento a vários fatores, dentre eles o ruído e a distorção ocorridos em *hardware*. Assim é feita uma nova reconstrução com essas mesmas medidas, considerando agora um determinado erro, descrita de forma geral na Equação (2.30) e especificamente na Equação (4.2), com um $\varepsilon = 0,14$, ou seja,

$$\min_{\tilde{s} \in \mathbb{C}^{32}} \|\tilde{s}\|_{l_1} \quad \text{sujeito a} \quad \|\Phi\Psi\tilde{s} - \mathbf{y}\|_{l_2} \leq 0,14. \quad (4.2)$$

Os resultados dessa reconstrução podem ser vistos na Figura 4.5, com um erro menor que o anterior. O valor de ε é determinado de forma empírica.

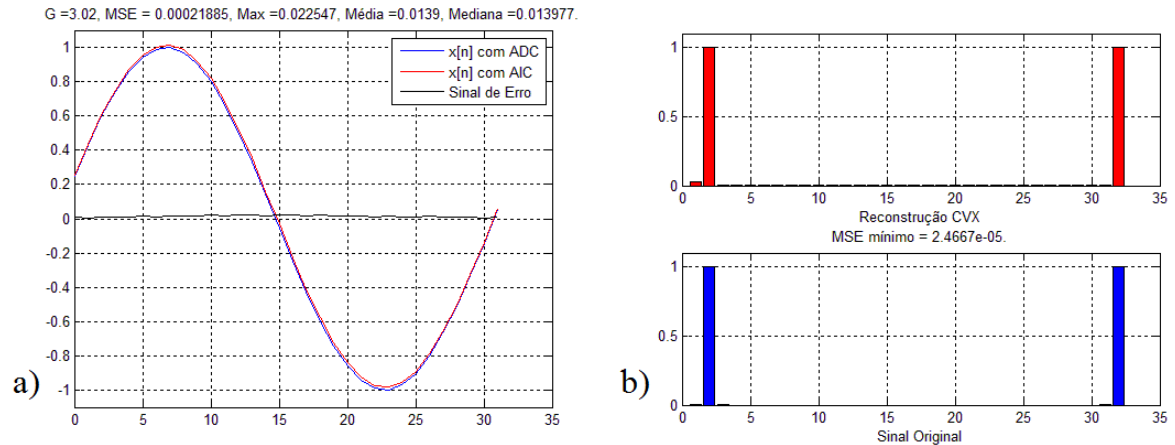


Figura 4.5: Reconstrução com CVX usando ε , das Medidas do AIC em *Hardware* para 1 tom:
a) Domínio do Tempo e b) Domínio da Frequência:
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

No Quadro 4.2 é possível fazer um comparativo numérico dessas três reconstruções. Não apenas pelos gráficos, mas numericamente é possível constatar que os valores

reconstruídos com as medidas do AIC em *hardware* ficaram próximos do modelo do AIC em *software*.

	AIC Software	AIC Hardware	AIC Hardware com ε
MSE de $\hat{s}[n]$	8,7538e-6	9,0017e-5	2,4667e-5
MSE de $\tilde{x}[n]$	7,8343e-5	0,00072184	0,00021885
Máximo do erro de $\tilde{x}[n]$	0,018361	0,05704	0,022547
Média do erro de $\tilde{x}[n]$	0,0074439	0,021756	0,0139
Mediana do erro de $\tilde{x}[n]$	0,0067108	0,018391	0,013977

Quadro 4.2: Comparativos das 3 Reconstruções para uma Senóide (1 tom).

A repetitividade (repetibilidade) foi realizada com 10 conjuntos de medidas sobre as mesmas condições, os resultados podem ser visto na Figura 4.6, em: a) as medidas $y[n]$ realizadas no AIC em *hardware*, b) os sinais $x[n]$ reconstruídos e c) os sinais $s[n]$ também reconstruídos. Ao lado direito de cada gráfico é possível visualizar uma ampliação de um determinado conjunto de amostra, idealmente todos esses pontos deveriam ter os mesmos valores. Contudo os resultados têm boa precisão, conforme pode ser visto em detalhe para alguns pontos na Figura 4.6.

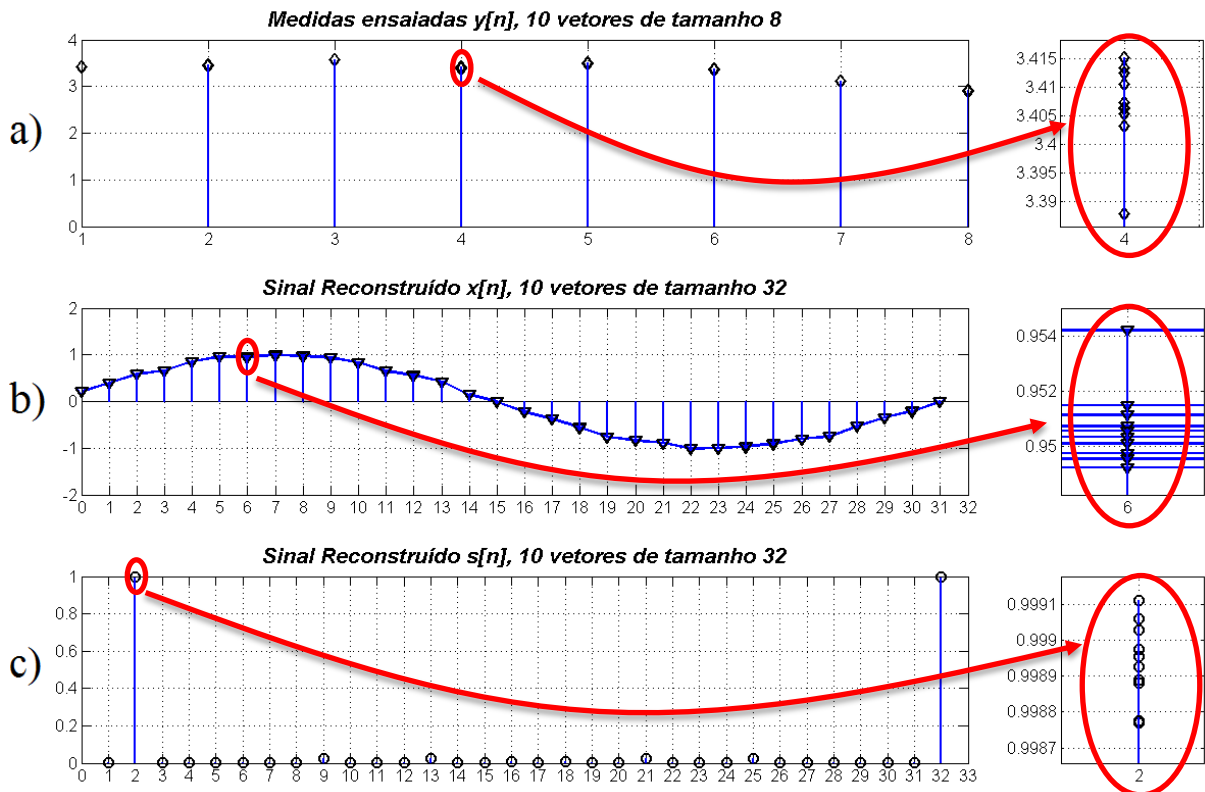


Figura 4.6: Reconstrução com CVX de 10 conjuntos de Medidas com AIC em *Hardware* para 1 tom: a) As 10 Medidas $y[n]$ de 8 valores, b) Os 10 Sinais $x[n]$ reconstruídos e c) Os 10 Sinais $s[n]$ Reconstruídos.

4.1.2 Sinal com dois tons

As simulações e os ensaios realizados para sinais com dois tons foram feitos da mesma forma que os realizados para sinais com um tom, com as mesmas configurações do AIC conforme Quadro 4.1. Sendo que amplitude do primeiro tom é de 0,5 V e do segundo tom é de 1 V, com frequências de 1 kHz e 3 kHz respectivamente, e ambos com fase zero.

Inicialmente são apresentados na Figura 4.7 os sinais internos do AIC em *Software*. Embora alguns desses sinais sejam diferentes dos vistos na simulação anterior (pois agora o sinal de entrada é outro), suas descrições são as mesmas, já descritas na Seção 4.1.1, lembrando que nesta figura foram omitidos os sinais de controle e sincronismo.

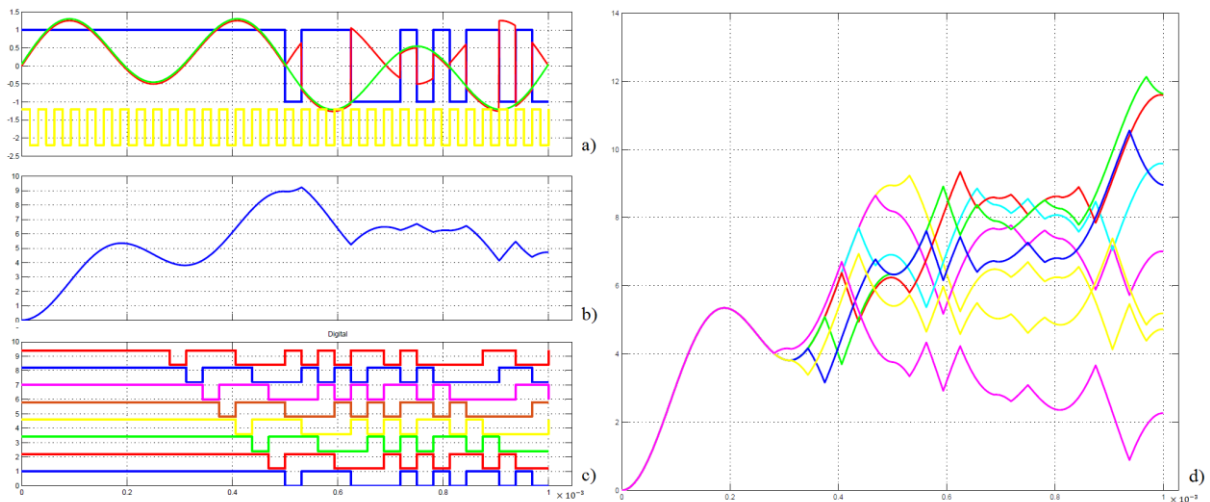


Figura 4.7: Sinais Internos do AIC em *Software* (Simulado) para 2 tons:
a) Sinais Referentes ao Canal 1, b) Na Saída do Filtro do Canal 1, c) Nas Saídas do LFSR e d) Na saída dos 8 filtros

A reconstrução feita com o CVX, conforme Equação (4.1), pode ser visualizada na Figura 4.8, sendo que em a) é possível visualizar o sinal de erro (em preto), além do sinal de referência (em azul) e o sinal reconstruído $\tilde{x}[n]$ (em vermelho), e em b) o sinal de referência (em azul) com suas dois tons e o sinal reconstruído (em vermelho) com alguns erros nos valores das amplitude e pequenas distorções. Nesta figura é escrita sua principal métrica o $MSE = 0,0052073$ no domínio temporal e o $MSE = 0,00048268$ no domínio espectral.

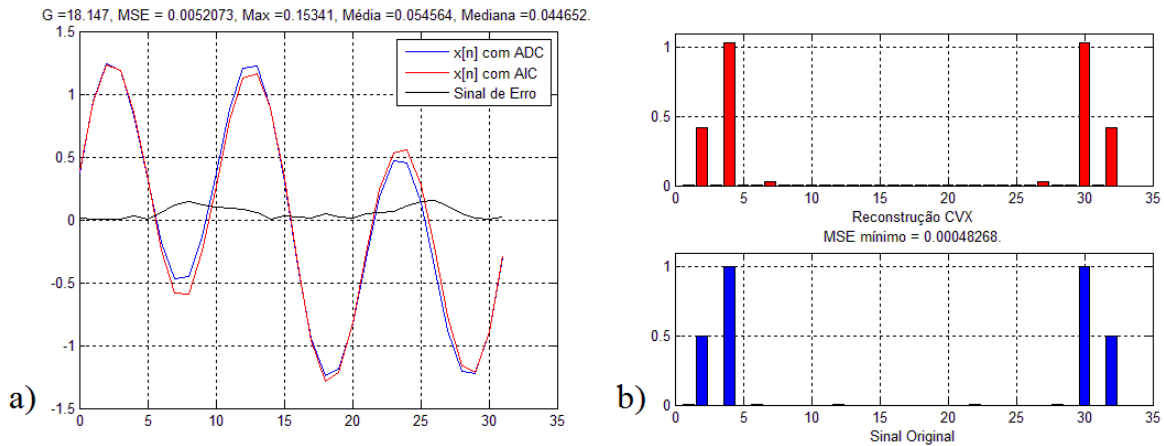


Figura 4.8: Reconstrução com CVX das Medidas do AIC em *Software* (Simulado) para 2 tons:
a) Domínio do Tempo e b) Domínio da Frequência:
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

A partir desses resultados simulados são feitos vários ensaios no AIC em *Hardware* para validação e comparações dos resultados, Na Figura 4.9 é possível visualizar alguns sinais internos neste AIC, por meios de dois osciloscópios. Como já descritos anteriormente, nesta figura são expostos também os sinais de controle e sincronismo.

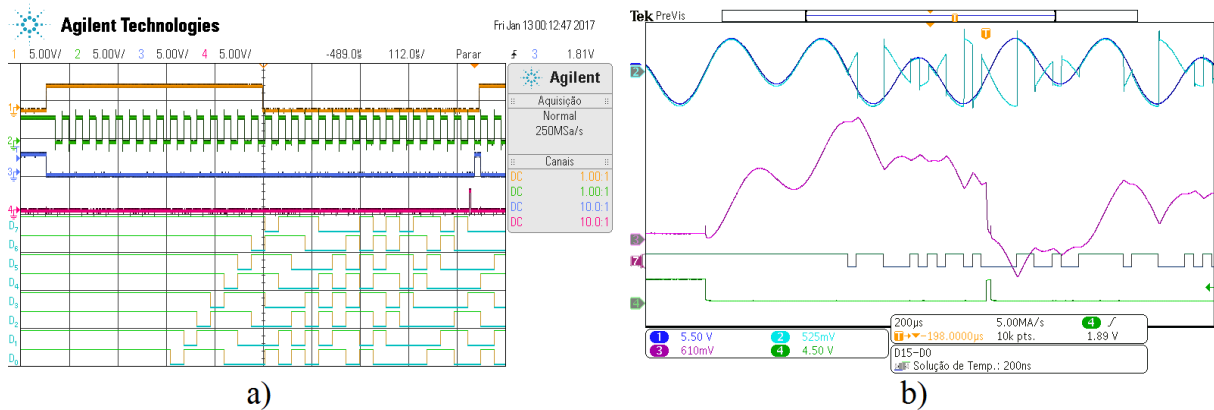


Figura 4.9: Sinais Internos do AIC em *Hardware* para 2 tons :
a) Sinais de Controle (dentre outros) e b) Sinais do Canal 1

É importante comparar as semelhanças dos sinais em comum mostrados na Figura 4.7 e na Figura 4.9, bem como os comportamentos de *reset*, gatilhos e posicionamento desses sinais, consequência direta dos circuitos de controle e de sincronismo internos a FPGA.

Com as medidas $y[n]$ obtidas do AIC em *Hardware* é feita a reconstrução, exatamente igual as feitas com as medidas simuladas anteriormente. Os resultados podem ser visualizados na Figura 4.10 e comparados com os da Figura 4.8. Um detalhe importante e não claramente perceptível é que o sinal de referência (em azul) não é exatamente igual para

as reconstruções das medidas simuladas e ensaiadas ($x[n]$ com ADC), seja no domínio temporal e espectral. O motivo dessa diferença é que as amostras idealizadas (do ADC idealizado) são ajustadas ao atraso (ajuste de fase) do sinal reconstruído (em vermelho), esse atraso é inerente ao sistema como um todo e de modelamento não trivial.

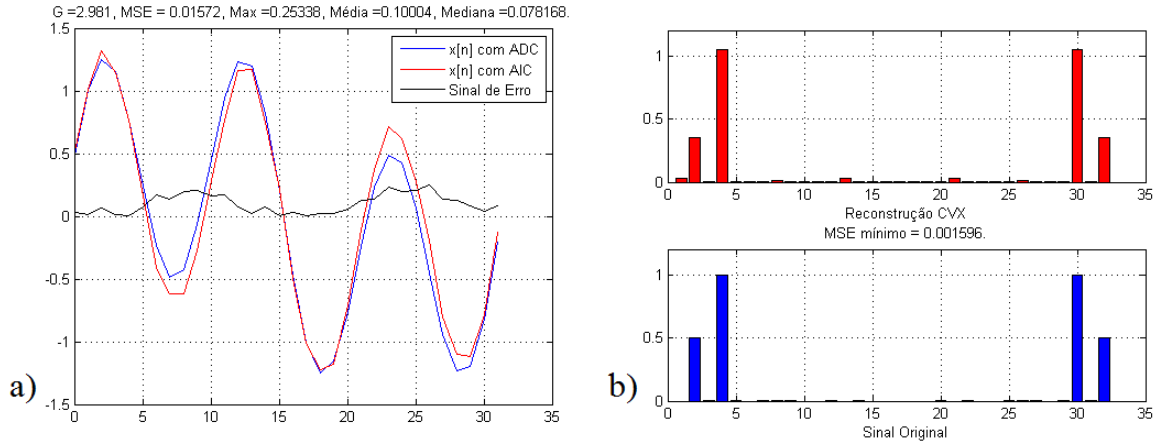


Figura 4.10: Reconstrução com CVX das Medidas do AIC em *Hardware* para 2 tons:
a) Domínio do Tempo e b) Domínio da Frequência:
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

A reconstrução seguinte é realizada utilizando as mesmas medidas $y[n]$ anteriores, obtidas no AIC em *Hardware*, todavia na reconstrução é considerado um erro de $\varepsilon = 0,6$, o resultado dessa nova busca é representado na Equação (4.3), e ilustrado na Figura 4.11.

$$\min_{\tilde{s} \in \mathbb{C}^{32}} \|\tilde{s}\|_{l_1} \quad \text{sujeito a } \|\Phi\Psi\tilde{s} - y\|_{l_2} \leq 0,6 \quad (4.3)$$

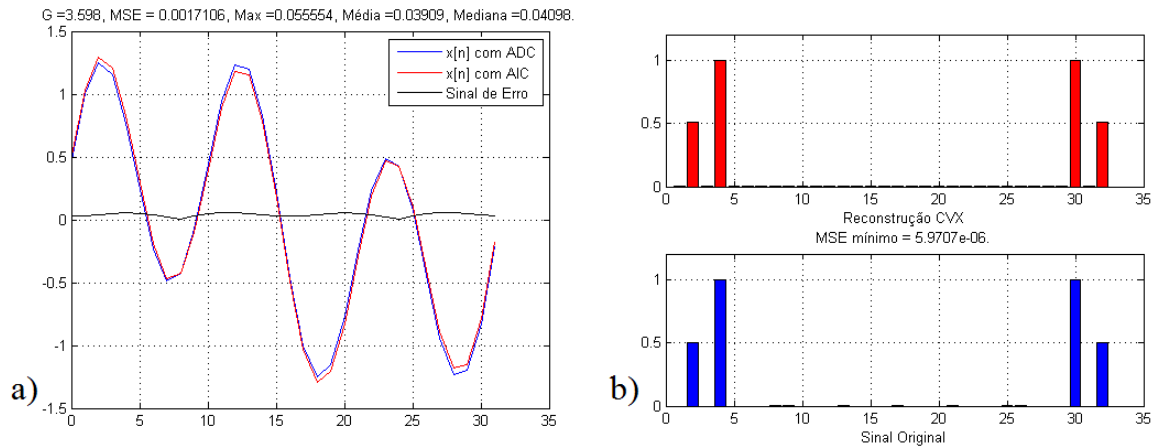


Figura 4.11: Reconstrução com CVX usando ε , das Medidas do AIC em *Hardware* para 2 tons:
a) Domínio do Tempo e b) Domínio da Frequência:
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

No Quadro 4.2 é feito um comparativo numérico dessas três reconstruções, sendo verificado que a melhor reconstrução foi a que utilizou o valor de ε e não a reconstrução feita com as medidas simuladas, fato não ocorrido anteriormente.

	<i>AIC Software</i>	<i>AIC Hardware</i>	<i>AIC Hardware com ε</i>
MSE de $\tilde{s}[n]$	0,00048268	0,001592	5,9707e-6
MSE de $\tilde{x}[n]$	0,0052073	0,01572	0,0017106
Máximo do erro de $\tilde{x}[n]$	0,15341	0,25338	0,055554
Média do erro de $\tilde{x}[n]$	0,054564	0,10004	0,03909
Mediana do erro de $\tilde{x}[n]$	0,044652	0,078168	0,04098

Quadro 4.3: Comparativos das 3 Reconstruções para 2 tons.

Para verificar a repetitividade (repetibilidade) foram realizados 10 conjuntos de medidas $y[n]$ no AIC em *Hardware* com a mesma configuração, e para cada um desses conjuntos de medidas foram feitas as reconstruções, na Figura 4.12 é possível visualizar em: a) os 10 conjuntos de 8 valores de $y[n]$ e em detalhe a direita o 7º valor de cada conjunto, b) os 10 conjuntos de 32 valores reconstruídos de $x[n]$ e em detalhe a direita o 13º valor de cada conjunto e c) os 10 conjuntos de 32 valores reconstruídos de $s[n]$ e em detalhe a direita o 2º valor de cada conjunto.

Dessa forma é demonstrada a precisão não apenas do AIC em *Hardware* como também das reconstruções realizadas.

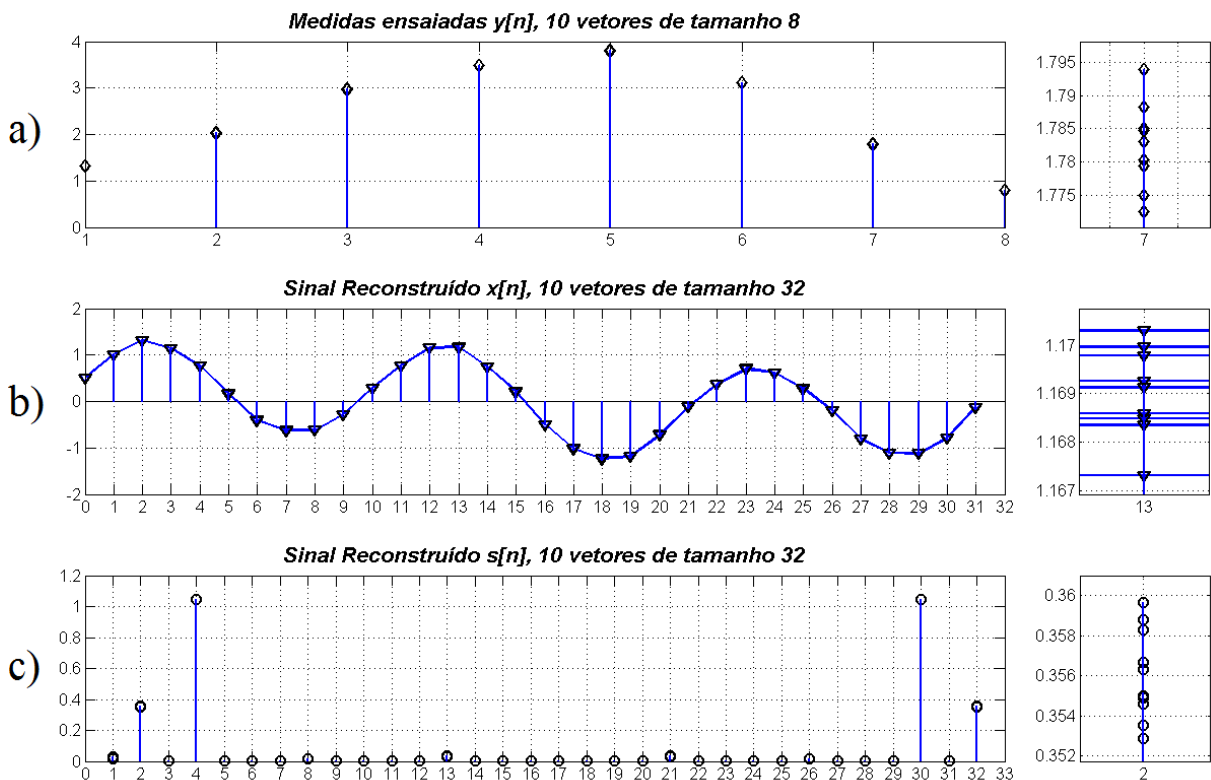


Figura 4.12: Reconstrução com CVX de 10 conjuntos de Medidas com AIC em *Hardware* para 2 tons:

a) As 10 Medidas $y[n]$ de 8 valores, b) Os 10 Sinais $x[n]$ reconstruídos e c) Os 10 Sinais $s[n]$ Reconstruídos.

4.1.3 Sinal com três tons

As simulações e os ensaios realizados para três tons foram feitos da mesma forma que os realizados para uma e dois tons, com as mesmas configurações do AIC conforme Quadro 4.1. Sendo que agora amplitude do primeiro, segundo e do terceiro tons são de 1 V, 0,75 V e 0,375V, com frequências de 1 kHz, 6 kHz e 8 kHz, respectivamente.

Seguindo a mesma sequência, inicialmente são apresentados na Figura 4.13 os sinais internos do AIC em *Software*, mais uma vez lembrando que nesta figura foram omitidos os sinais de controle e sincronismo.

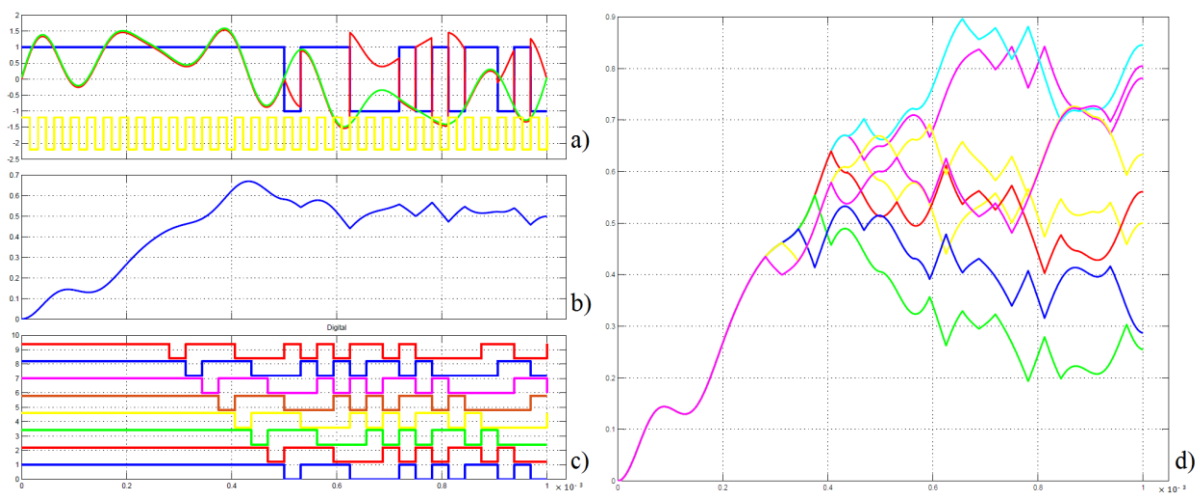


Figura 4.13: Sinais Internos do AIC em *Software* (Simulado) para 3 tons:
a) Sinais Referentes ao Canal 1, b) Na Saída do Filtro do Canal 1, c) Nas Saídas do LFSR e d) Na saída dos 8 filtros

A reconstrução feita com o CVX conforme Equação (4.1) pode ser visualizada na Figura 4.14, sendo que em a) é possível visualizar o sinal de erro (em preto), além do sinal de referência (em azul) e o sinal reconstruído (em vermelho) no domínio temporal, e em b) o sinal de referência (em azul) com seus três tons e o sinal reconstruído (em vermelho) no domínio espectral. Os erros do $MSE = 0,0052073$ no domínio temporal e o $MSE = 0,00048268$ no domínio espectral podem ser visualizados também nesta figura.

Um detalhe importante desses três sinais reconstruídos até agora é o fato do MSE do $s[n]$ (domínio da frequência ou esparso) ser sempre menor que o MSE do $x[n]$ (domínio da tempo ou não esparso), há duas razões principais para essa diferença, a primeira razão é que a busca sempre é feita no domínio esparso (no caso a frequência) e o sinal de referência nesse domínio é muito bem definido, a segunda razão está na dificuldade de se conseguir um bom sinal de referência no domínio do tempo, esse sinal de referência depende de alguns ajustes de ganho e de fase.

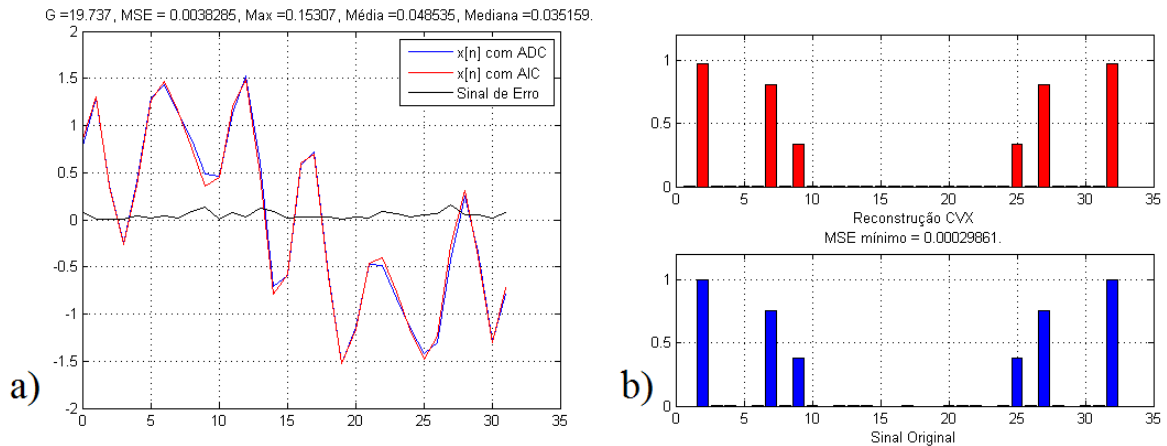


Figura 4.14: Reconstrução com CVX das Medidas do AIC em *Software* (Simulado) para 3 tons:
a) Domínio do Tempo e b) Domínio da Frequência:
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

Após a simulação, são feitos os ensaios no AIC em *Hardware*, sendo possível visualizar na Figura 4.15 alguns sinais internos do AIC, conforme já descritos anteriormente, sendo em: a) principalmente os sinais de controle e sincronismo e b) principalmente os efeitos do processamento do sinal de entrada $x(t)$ até obter as medidas $y[n]$.

Neste momento é importante fazer as comparações desses sinais ensaiados com os sinais simulados.

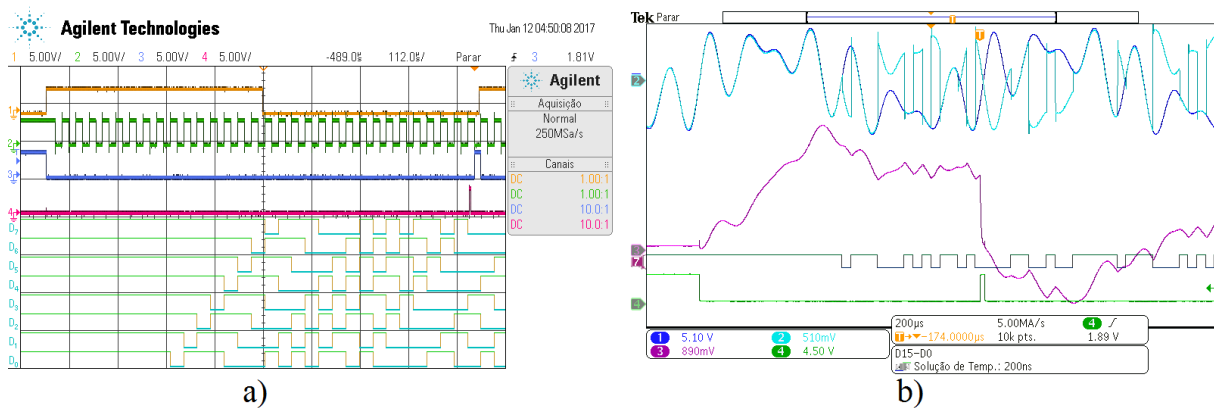


Figura 4.15: Sinais Internos do AIC em *Hardware* para 3 tons :
a) Sinais de Controle (dentre outros) e b) Sinais do Canal 1

A primeira reconstrução das medidas ensaiadas é feita de forma idêntica a reconstrução das medidas simuladas, ambos de acordo com a Equação (4.1), seus resultados podem ser visto na Figura 4.16 e comparados com os da Figura 4.14.

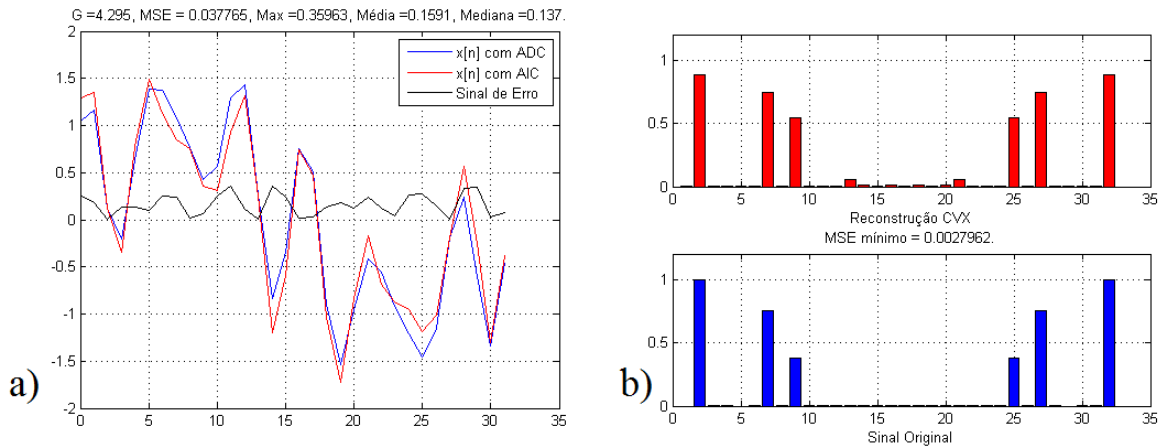


Figura 4.16: Reconstrução com CVX das Medidas do AIC em *Hardware* para 3 tons:
a) Domínio do Tempo e b) Domínio da Frequência:
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

A segunda reconstrução das medidas ensaiadas é feita de forma diferente das duas reconstruções anteriores, de acordo com a Equação (4.3), pois é considerado um $\varepsilon = 0,6$. Os resultados podem ser vistos na Figura 4.17 e comparados com os da Figura 4.16 e da Figura 4.14. Claramente é possível visualizar algumas melhorias nos sinais, referentes à reconstrução anterior.

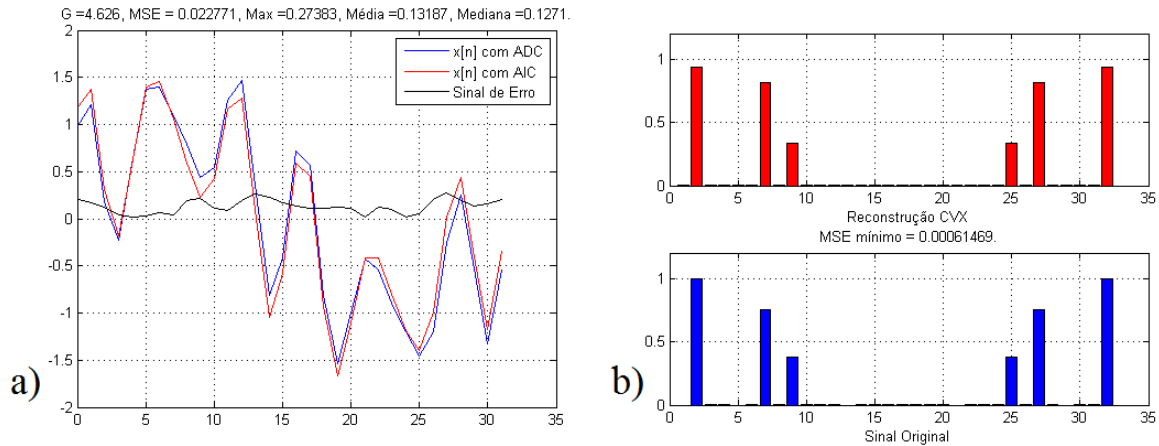


Figura 4.17: Reconstrução com CVX usando ε , das Medidas do AIC em *Hardware* para 3 tons:
a) Domínio do Tempo e b) Domínio da Frequência:
em: Azul o Sinal Original, Vermelho o Sinal Reconstruído e Preto o Sinal de Erro.

Para avaliação e comparação menos subjetivas são mostrados no Quadro 4.4 os resultados numéricos dessas três reconstruções.

	<i>AIC Software</i>	<i>AIC Hardware</i>	<i>AIC Hardware com ε</i>
MSE de $\hat{s}[n]$	0,00029861	0,0027962	0,00061469
MSE de $\hat{x}[n]$	0,0038285	0,037765	0,022771
Máximo do erro de $\hat{x}[n]$	0,15307	0,35963	0,27383
Média do erro de $\hat{x}[n]$	0,048535	0,1591	0,13187
Mediana do erro de $\hat{x}[n]$	0,035159	0,137	0,1271

Quadro 4.4: Comparativos das 3 Reconstruções para 3 tons.

Por fim, para avaliar a repetitividade (repetibilidade) foram realizados 10 conjuntos de medidas no AIC em *Hardware* sobre as mesmas condições, e para cada conjunto desses foram reconstruídos os sinais de acordo com a Equação (4.1). Os resultados podem ser vistos na Figura 4.18, sendo que em: a) as medidas $y[n]$ e em detalhe a direita o 2º valor de cada conjunto, b) os sinais $x[n]$ reconstruídos e em detalhe a direita o 6º valor de cada conjunto, e c) os sinais $s[n]$ e em detalhe a direita o 9º valor de cada conjunto.

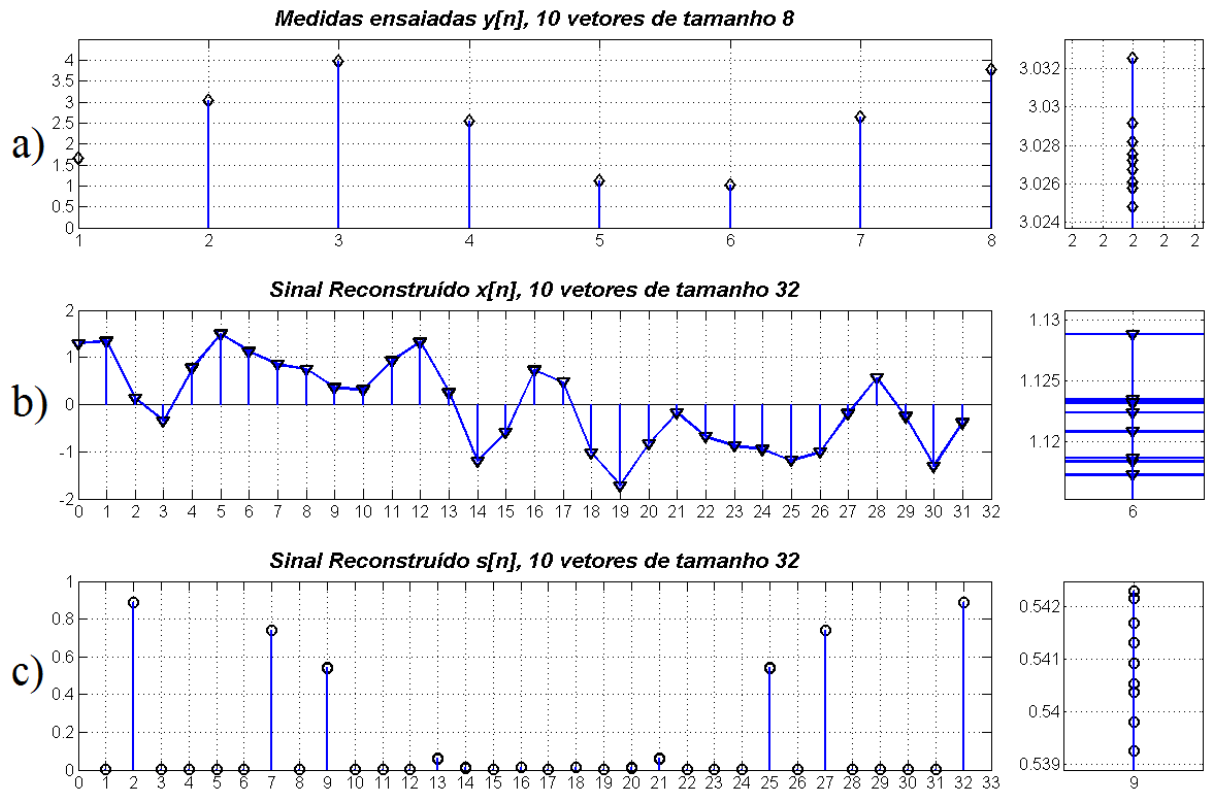


Figura 4.18: Reconstrução com CVX de 10 conjuntos de Medidas com AIC em *Hardware* para 3 tons: a) As 10 Medidas $y[n]$ de 8 valores, b) Os 10 Sinais $x[n]$ reconstruídos e c) Os 10 Sinais $s[n]$ Reconstruídos.

4.2 Medidas no Domínio Esparso

O sinal usado para esse ensaio é esparso no mesmo domínio que são realizadas as medidas, i.e. é esparso no tempo. Sinais semelhantes a esses foram usados em alguns AIC desenvolvidos, e podem servir como um sinal de referência entre os AIC, e por consequência possibilitar algumas comparações entre esses *hardware*.

O sinal é uma sequência aleatória de pulsos retangulares, um pulso retangular* pode ser descrito usando a função degrau unitário $u(t)$, conforme

$$p(t) = u(t) - u(t - L), \quad (4.4)$$

sendo L a largura do pulso.

Uma sequência desses pulsos deslocados no tempo descreve o sinal utilizado, denotado $s(t)$ e descrito por

$$s(t) = \sum_{i=1}^n a_i p(t - i \cdot L), \quad (4.5)$$

sendo a_i a amplitude de cada pulso e n o número total de pulsos utilizados. Em particular o sinal usado foi construído com $n = 20$ tendo $a_2 = 1,4$ e $a_7 = 0,36$ e os outros $a_i = 0$, caracterizando um sinal com esparsidade $\|s\|_0 = 2$. A largura do pulso é $50 \mu s$.

As configurações usadas no AIC estão no Quadro 4.5 e diferem em duas situações das usadas nas outras medidas, embora o polinômio primitivo seja o mesmo, as saídas do LFSR para cada canal estão invertidas e a matriz de medida agora é $\Theta_{8 \times 20}$, tendo em vista que agora o sinal $s(t)$ tem 20 pulsos retangulares, i.e., $n = 20$.

Parâmetros Gerais	AIC <i>Software</i> e <i>Hardware</i>
Polinômio para LFSR	$x^5 + x^3 + x^2 + x + 1$
Ordem da Matriz de Medida	8×20
Frequência de corte do filtro	$\approx 50\text{Hz}$
Número de bits do ADC	16
Taxa de amostragem do ADC por canal	1ksps

Quadro 4.5: Configurações Realizadas nas Medidas no AIC em *Software* e *Hardware*.

Os resultados são apresentados, na seguinte sequência: 1) verificação de alguns sinais do AIC em *Software*, 2) reconstrução das medidas simuladas, 3) verificação de alguns sinais do AIC em *Hardware*, 4) reconstrução das medidas ensaiadas, 5) reconstrução das medidas ensaiadas com um erro ε , 6) um quadro com um comparativo numérico do MSE dos três tipos

* Esse pulso é conhecido também como função porta $\text{ret}(t/L)$ e sua transformada de Fourier é $L \cdot \text{sinc}\left(\frac{\omega L}{2}\right)$, sendo $\text{sinc}(x) = \frac{\text{sen}(\pi x)}{\pi x}$ com largura de banda infinita.

de reconstrução e 7) um demonstrativo da repetitividade (repetibilidade) das medidas ensaiadas bem como das suas reconstruções.

Seguindo essa sequência, na Figura 4.19 é possível visualizar os sinais do AIC em *software*, exceto pelos sinais de controle, que foram omitidos. Sendo em: a) o sinal de relógio (*clock*) do LFSR (em verde), o sinal do canal 1 do LFSR (em azul) e o sinal na saída do filtro do canal 1 (em vermelho); b) o sinal $s(t)$ (em vermelho) e o sinal da saída do filtro do canal 1 (em azul), c) os 8 sinais usados nas saídas do LFSR e d) os 8 sinais na saída de cada um dos 8 filtros.

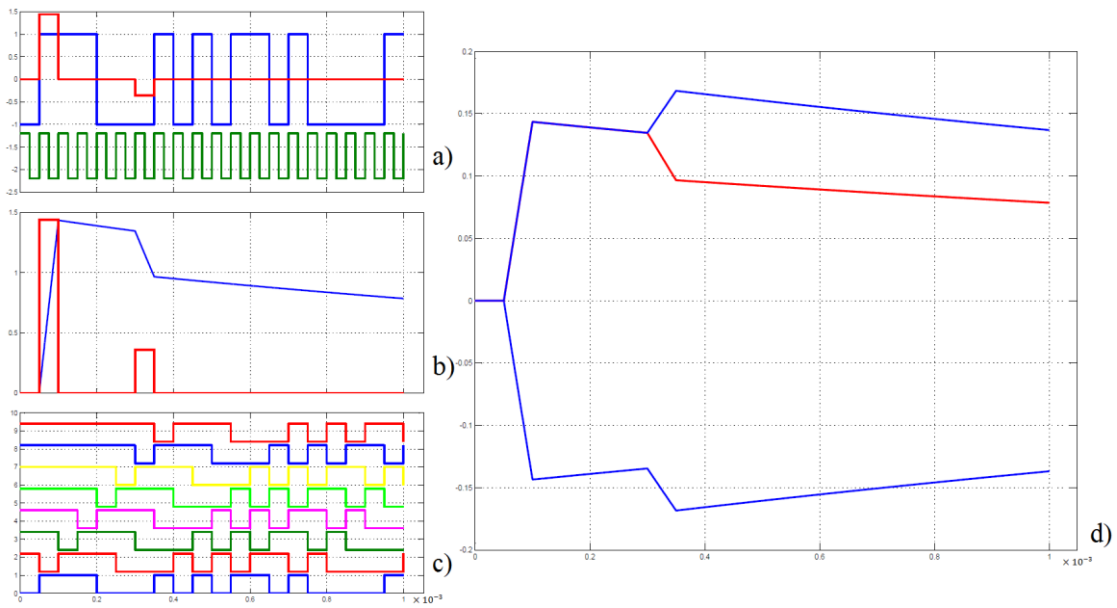


Figura 4.19: Sinais Internos do AIC em *Software* (Simulado) para o sinal esparso $x(t)$. a) PRBS (em azul), clock (em verde) e saída do multiplicador (em vermelho), b) $x(t)$ (em vermelho) e na saída do filtro canal 1 (em azul), c) Nas Saídas do LFSR e d) Na saída dos 8 filtros de cada canal.

O efeito de integração (acumulação) pode ser observado na Figura 4.19 b). É certo que idealmente o sinal deveria ser uma constante fora dos pulsos retangulares de $a_2 = 1,4$ e $a_7 = 0,36$, contudo a simulação foi configurada para modelar o comportamento dinâmico do filtro usado no AIC em *Hardware*. É possível melhorar esse comportamento aumentando a constante de tempo dos filtros utilizados, contudo isso é o mesmo que diminuir a frequência de corte e, por consequência, ter uma atenuação maior do sinal medido pelo ADC. É possível compensar isso com o aumento do ganho do filtro, mas a SNR tende a diminuir, pois esse ganho também afeta o ruído. Dessa forma, é estabelecida uma relação de compromisso discutida em [58].

Na reconstrução foi usado o CVX e o OMP, sendo os valores reconstruídos normalizados conforme Equação (4.6) e com a notação $\llbracket * \rrbracket^\dagger$.

$$\llbracket \tilde{s}[n] \rrbracket = \max \left(\frac{s[n]}{\tilde{s}[n]} \right) \cdot \tilde{s}[n]. \quad (4.6)$$

Sendo $s[n]$ o sinal original discretizado e $\tilde{s}[n]$ o sinal reconstruído.

Para o cálculo do MSE são utilizados os sinais $s[n]$ e $\text{Superios}(\tilde{s}[n])$. Cabe ressaltar que apenas o sinal medido $y[n]$ passa por uma quantização do ADC (16 bits).

A reconstrução das medidas simuladas e já normalizadas pode ser visualizada na Figura 4.20 juntamente com o sinal original (em azul). Neste caso, a reconstrução pelo CVX (em verde) foi realizada de acordo com a Equação (4.7) e a reconstrução usando o OMP (em vermelho) teve o valor da esparsidade $\|s\|_0 = 2$, além da matriz de medida $\Theta_{8 \times 20}$ e o vetor de medidas y .

$$\min_{\tilde{s} \in \mathbb{R}^{20}} \|\tilde{s}\|_{l_1} \text{ sujeito a } \Theta \tilde{s} = y \in \mathbb{R}^8. \quad (4.7)$$

Ainda na Figura 4.20 é possível observar os valores do MSE, tanto do sinal reconstruído pelo CVX como pelo OMP.

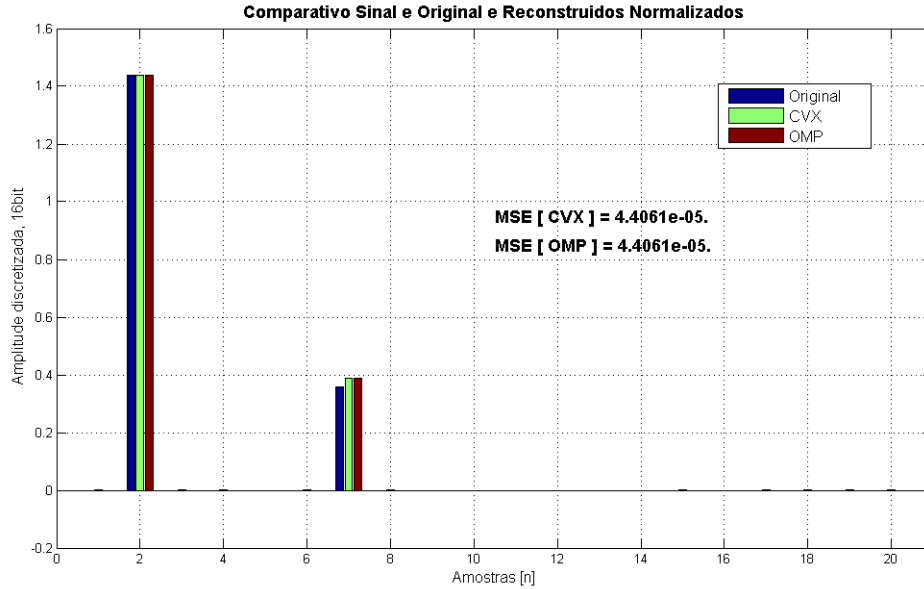


Figura 4.20: Reconstrução com CVX e OMP das Medidas do AIC em *Software* (Simulado) para o sinal $s(t)$.

[†] É definido aqui o operador $\llbracket * \rrbracket$, que multiplica todos os elementos sinal reconstruído $\tilde{s}[n]$ pelo maior quociente entre os elementos do sinal de referência $s[n]$ e do sinal reconstruído $\tilde{s}[n]$, i.e. é igualado um elemento dos vetores para comparar a diferença entre os demais elementos.

Na Figura 4.21 é possível observar vários dos sinais da Figura 4.19, sendo que agora esses sinais foram capturados do AIC em Hardware, sendo que em: a) o sinal de gatilho do gerador para o AIC (em laranja), o sinal de relógio (em verde), o sinal de *reset* (em azul), o sinal de gatilho do AIC para o ADC (DAQ) e por meio das entradas digitais do osciloscópio os sinais das saídas do LFSR, e b) os sinais referentes ao canal 1, semelhantes aos da Figura 4.19 exceto pelo sinal de *reset* (em verde).

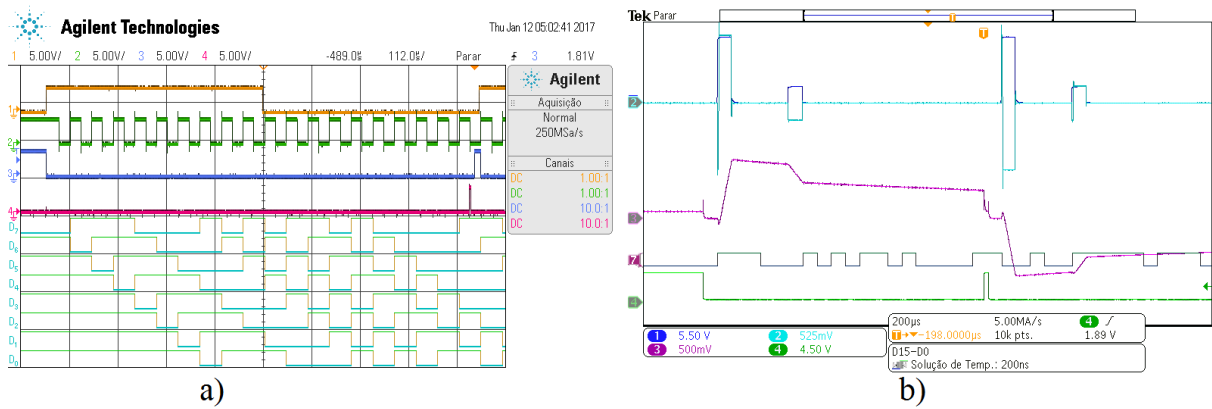


Figura 4.21: Sinais Internos do AIC em *Hardware* para $s(t)$:
a) Sinais de Controle (dentre outros) e b) Sinais do Canal 1.

A reconstrução das medidas em *hardware* (AIC HW) foram realizada de forma idêntica às medidas realizadas em *software* (AIC SW), sendo possível visualizar na Figura 4.22 que, além do MSE ter aumentado, houve algumas pequenas distorções do sinal reconstruído com o CVX, essas distorções não ocorrem com o OMP, pois a este é fornecida a esparsidade, $\|s\|_0 = 2$.

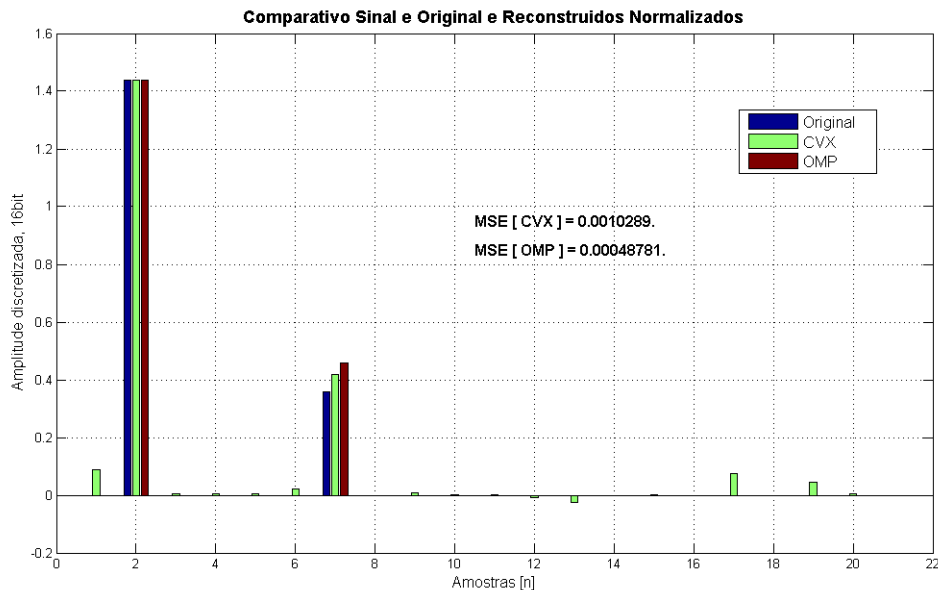


Figura 4.22: Reconstrução com CVX e OMP das Medidas do AIC em *Hardware* para o sinal $s(t)$.

A próxima reconstrução é feita considerando um erro $\varepsilon = 0,3$ apenas com o CVX, conforme descrito na Equação (4.8). Não há mudança alguma na reconstrução com o OMP.

$$\min_{\tilde{\mathbf{s}} \in \mathbb{R}^{20}} \|\tilde{\mathbf{s}}\|_{l_1} \quad \text{sujeito a } \|\Theta\tilde{\mathbf{s}} - \mathbf{y}\|_{l_2} \leq 0,3. \quad (4.8)$$

Na Figura 4.23 é possível visualizar essas reconstruções e observar que o MSE [CVX] ficou muito próximo dos resultados simulados, mesmo assim é possível observar as distorções, mesmo que menores que as anteriores.

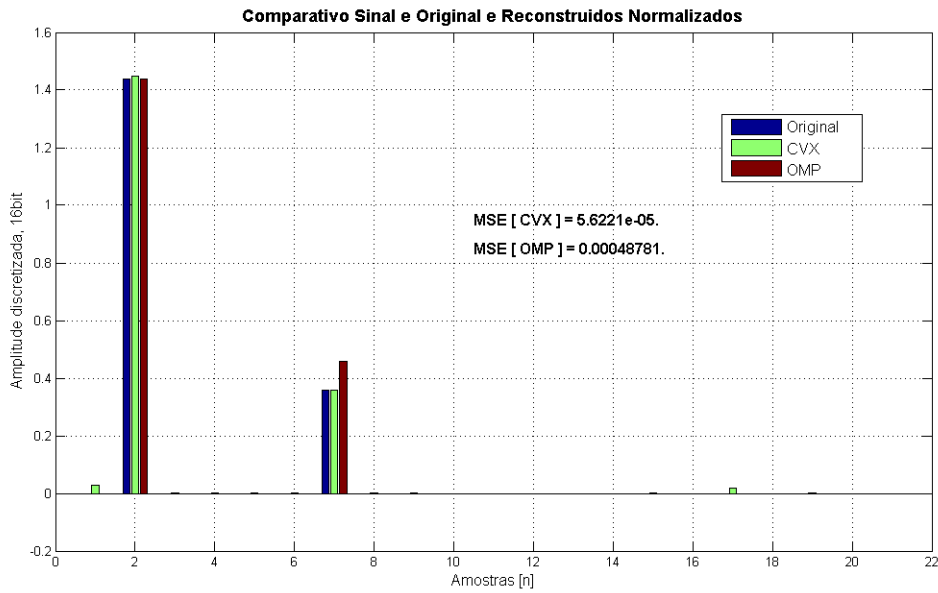


Figura 4.23: Reconstrução com CVX (com ε) e OMP das Medidas do AIC em *Hardware* para o sinal $s(t)$.

No Quadro 4.6 é possível observar o MSE das três reconstruções, com o CVX e com o OMP, e constatar que: 1) no AIC em *Software*, embora as duas reconstruções tenham o mesmo valor, o OMP estava com um dado a mais que o CVX, que é o valor da esparsidade; 2) no AIC em *hardware* o OMP teve uma construção mais robusta, dentre outras razões a esparsidade e 3) continuando com as mesmas medidas do AIC em *hardware*, o CVX com o ε teve um resultado muito superior ao OMP, da mesma ordem das medidas feitas no AIC em *software*.

	AIC <i>Software</i>	AIC <i>Hardware</i>	AIC <i>Hardware</i> com ε
MSE de $\tilde{\mathbf{s}}[n]$ com CVX	4,4061e-5	0,0010289	5,6221e-5
MSE de $\tilde{\mathbf{s}}[n]$ com OMP	4,4061e-5	0,00048781	0,00048781

Quadro 4.6: Comparativos das 3 Reconstruções para o sinal $s(t)$.

Para verificar a repetitividade (repetibilidade) do AIC em *hardware*, foram realizados 10 conjuntos de medidas (cada uma com 8 valores) sobre as mesmas condições, em seguida para cada conjunto desses foram feitas reconstruções usando o CVX e o OMP, da mesma forma que foram mostradas na Figura 4.22.

Na Figura 4.24 é possível visualizar esses resultados, sendo que em: a) os 10 conjuntos das medidas $y[n]$ de 8 valores, realizadas no AIC em *hardware* e em detalhe a direita os 10 valores da 7ª. posição (i.e. $y[7]$), b) os 10 conjuntos de 20 valores dos sinais $\tilde{s}[n]$ reconstruídos com o CVX e em detalhe a direita os 10 valores da 7ª. posição (i.e. $\tilde{s}[7]$) e c) os 10 conjuntos de 20 valores dos sinais $\tilde{s}[n]$ reconstruídos com o OMP e em detalhe a direita os 10 valores da 7ª. posição (i.e. $\tilde{s}[7]$), idealmente todos esses pontos deveriam ser iguais. Mesmo assim, os resultados demonstram boa precisão.

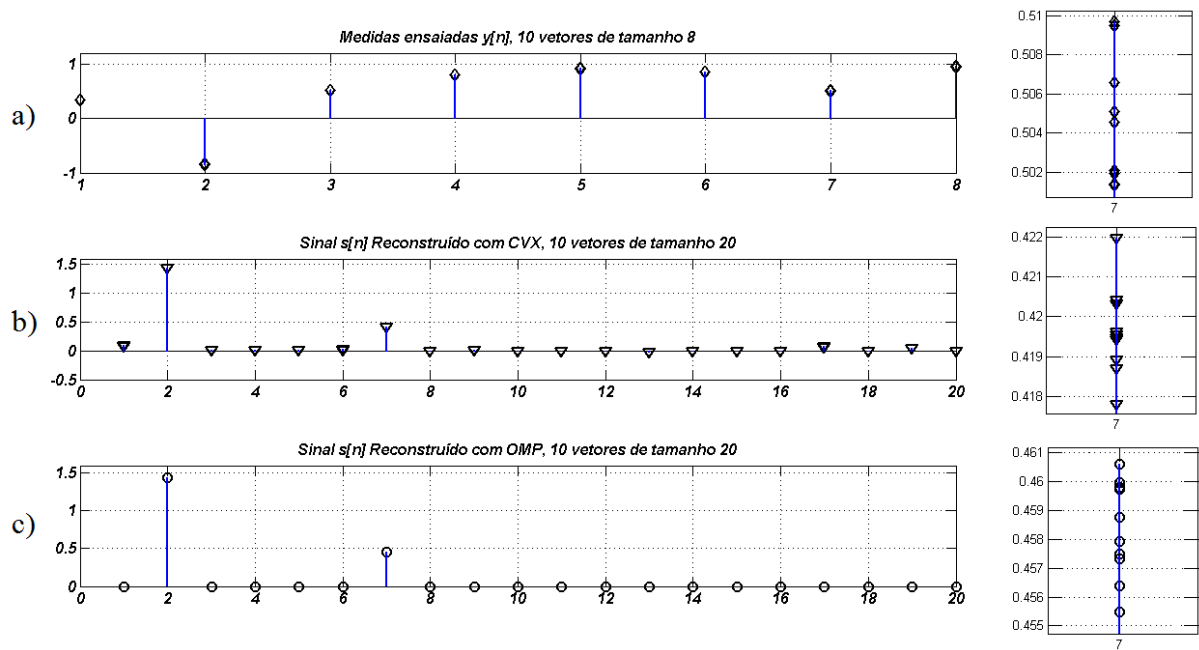


Figura 4.24: Repetitividade do AIC em *Hardware* para o sinal $s(t)$, para 10 conjuntos de Medidas:
a) As 10 Medidas $y[n]$, b) Os 10 Sinais $\tilde{s}[n]$ reconstruídos com CVX e c) Os 10 Sinais $\tilde{s}[n]$ reconstruídos com OMP.

5 CONCLUSÃO E TRABALHOS FUTUROS

5.1 Conclusão

No contexto em que os avanços tecnológicos propiciam o aumento exponencial na quantidade de dados gerados e que se busca formas eficientes de armazenar, transmitir e processar esses dados, baseando-se na ideia de projeções aleatórias, foi proposta a Amostragem Compressiva (*Compressed Sampling*) para realizar a aquisição de sinais de forma a se obter a representação apenas da informação neles contida.

A Amostragem Compressiva permite tirar proveito de uma característica do sinal (a esparsidade) para realizar as medidas com taxas inferiores as de Nyquist. A reconstrução do sinal original, por meio dessas medidas, é possível e computacionalmente viável pela convexidade da norma l_1 .

As implementações em *hardware* da Amostragem Compressiva, denominadas de Conversor Analógico para Informação (AIC, do inglês *Analog to Information Converter*), normalmente têm sido realizadas considerando-se apenas um tipo de sinal esparsos, por exemplo, pulso de radar (esparsos no tempo), sinais tonais (esparsos em frequência), dentre outros. Neste trabalho foi proposta uma arquitetura que podia ser modificada de forma a tornar o *hardware* configurável e possibilitar o seu uso com diversos tipos de sinais esparsos, apenas modificando os seus parâmetros.

O trabalho foi iniciado pela análise matemática dos fundamentos da Amostragem Compressiva, e em especial dos AIC. Após a realização de uma modelagem matemática, foi feita uma busca na bibliografia, percebendo-se que os AIC propostos são dedicados a uma classe específica de sinais e que normalmente eles não podem ser modificados para outras classes. Dessa forma foi proposta uma arquitetura flexível, que podia ser utilizada em mais de uma classe de sinais, apenas com modificação de seus parâmetros.

O segundo passo foi realizar a simulação do AIC proposto, o que permitiu a análise do seu comportamento dinâmico e a obtenção de parâmetros a serem utilizados no *hardware*. De posse do simulador e dos parâmetros por ele fornecidos, partiu-se para a implementação em *hardware*. Em seguida foi desenvolvido um cenário (*setup*) de medição de forma a permitir a comparação do sinal original com o recuperado a partir das medidas adquiridas pelo *hardware*. O *setup* proposto era genérico, configurável e permitia realizar medições de diversos parâmetros do *hardware* de forma síncrona.

Em seguida, foi implementado o *hardware* proposto. A implementação realizada foi dividida em duas partes, uma analógica e outra digital. Na parte analógica foram projetados filtros para funcionarem como integradores, e a implementação realizada permitiu a escolha de componentes para se obter filtros com diversas frequências de corte. A parte digital foi implementada em FPGA, podendo ser configurada por parâmetros enviados por *software*. Essa parte foi de grande importância para a configurabilidade do AIC proposto, pois gerava sinais de sincronismo ajustáveis aos sinais que estavam sendo medidos.

A partir da arquitetura configurável definiu-se a matriz de medida, frequência de corte e ganho do filtro, pulsos de sincronismo, número de canais e frequência do PRBS dentre outros parâmetros.

Com o AIC simulado e implementado, foram feitas medições de dois tipos de sinais: um deles composto de soma de cossenos (tonais), cuja esparsidade está no domínio da frequência, e o outro de sinais pulsados, cuja esparsidade está no domínio do tempo. Em ambos os casos a figura de mérito foi o erro médio quadrático entre as amostras do sinal original obtidas usando um conversor baseado no critério de Nyquist, e as amostras recuperadas a partir das medidas adquiridas pelo AIC proposto. Dessa forma, pode-se fazer comparação entre os resultados simulados e medidos.

Os erros encontrados têm valores baixos para as classes de sinais usadas como teste. Portanto, o AIC proposto consegue medir bem duas classes de sinais com domínios de esparsidade distintos.

5.2 Trabalhos Futuros

Com esse AIC configurável vislumbrou-se diversas outras pesquisas que podem ser realizadas como trabalhos futuros.

O primeiro é a ampliação da faixa de sinais que podem ser medidos, pela implementação de um novo protótipo do AIC Configurável com maior frequência de operação, hoje limitada em 1MHz devido ao multiplicador utilizado (AD633), maior número de canais e controle digital da frequência de corte dos filtros e do ganho do amplificador.

Há ainda outras características que precisam ser mais bem analisadas, como o efeito do ruído no desempenho do AIC proposto, visto que na literatura já foi demonstrado que esses conversores têm seu desempenho bastante degradado quando há presença de ruído.

Foram utilizadas matrizes cujos valores são obtidos de sequências pseudoaleatórias, e observou-se que essas matrizes influenciam diretamente no desempenho do sistema, assim há a necessidade de um estudo mais aprofundado sobre essas matrizes, principalmente considerando o fato que elas precisam ser geradas em *hardware*.

A implementação do AIC proposto em chip é outro ponto que merece análise, visto que a integração irá, no mínimo, reduzir a complexidade de manuseio do *hardware*.

6 REFERÊNCIAS

- [1] D. L. Donoho, “Compressed sensing”, *IEEE Transactions on Information Theory*, vol. 52, n° 4, p. 1289–1306, abr. 2006.
- [2] L. Floridi, Org., *The philosophy of information quality*. Cham: Springer, 2014.
- [3] “Executive Summary: Data Growth, Business Opportunities, and the IT Imperatives | The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things”. [Online]. Disponível em: <https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm>. [Acessado: 19-mar-2017].
- [4] R. D. Regazzi e A. de Campos, *Soluções Práticas de Instrumentação e Automação - LabVIEW*, 3R.KWG. Rio de Janeiro, Brasil: KWG, 2005.
- [5] W. Kester e Analog Devices, inc, Orgs., *Data conversion handbook*. Amsterdam ; Boston: Elsevier ; Newnes, 2005.
- [6] D. L. Donoho, “Compressed sensing”, *IEEE Transactions on Information Theory*, vol. 52, n° 4, p. 1289–1306, abr. 2006.
- [7] E. J. Candes e T. Tao, “Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?”, *IEEE Transactions on Information Theory*, vol. 52, n° 12, p. 5406–5425, dez. 2006.
- [8] M. Mangia, R. Rovatti, e G. Setti, “Rakeness in the Design of Analog-to-Information Conversion of Sparse and Localized Signals”, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, n° 5, p. 1001–1014, maio 2012.
- [9] T. K. Moon, *Mathematical methods and algorithms for signal processing*. Upper Saddle River, NJ: Prentice Hall, 2000.
- [10] E. J. Candès, “The restricted isometry property and its implications for compressed sensing.”, *Compte Rendus de l’Academie des Sciences, Paris, Serie I*, 346, p. 589–592.
- [11] E. J. Candes e T. Tao, “Decoding by linear programming”, *IEEE Transactions on Information Theory*, vol. 51, n° 12, p. 4203–4215, dez. 2005.
- [12] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. New York: Springer, 2010.
- [13] “Introduction — CVX Users’ Guide”. [Online]. Disponível em: <http://web.cvxr.com/cvx/doc/intro.html>. [Acessado: 24-maio-2015].
- [14] J. A. Tropp e S. J. Wright, “Computational Methods for Sparse Solution of Linear Inverse Problems”, *Proceedings of the IEEE*, vol. 98, n° 6, p. 948–958, jun. 2010.
- [15] S. Kirolos *et al.*, “Analog-to-Information Conversion via Random Demodulation”, in *2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, 2006, p. 71–74.
- [16] T. Ragheb, J. N. Laska, H. Nejati, S. Kirolos, R. G. Baraniuk, e Y. Massoud, “A prototype hardware for random demodulation based compressive analog-to-digital conversion”, in *51st Midwest Symposium on Circuits and Systems, 2008. MWSCAS 2008*, 2008, p. 37–40.
- [17] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, e R. G. Baraniuk, “Beyond Nyquist: Efficient Sampling of Sparse Bandlimited Signals”, *IEEE Transactions on Information Theory*, vol. 56, n° 1, p. 520–544, jan. 2010.
- [18] S. Kirolos *et al.*, “Analog-to-Information Conversion via Random Demodulation”, in *2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, 2006, p. 71–74.
- [19] Y. Massoud, S. Smaili, e V. Singal, “Efficient realization of random demodulator-based analog to information converters”, in *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2011, p. 133–136.

- [20] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, e Y. Massoud, "Theory and Implementation of an Analog-to-Information Converter using Random Demodulation", in *IEEE International Symposium on Circuits and Systems, 2007. ISCAS 2007*, 2007, p. 1959–1962.
- [21] S. A. Varma e K. M. M. Prabhu, "A new approach to near-theoretical sampling rate for modulated wideband converter", in *2014 International Conference on Signal Processing and Communications (SPCOM)*, 2014, p. 1–5.
- [22] S. Liu, M. Zhang, e W. Jiang, "Design and exploration of low-power sub-nyquist processing with Modulated Wideband Converter", in *2014 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, 2014, p. 1–3.
- [23] L. Chen, J. Jin, e Y. Gu, "A calibration system and perturbation analysis for the Modulated Wideband Converter", in *2010 IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, p. 78–81.
- [24] Y. Zhou-wei, L. Qi-Qin, Z. Yu, e F. Jian, "Compressed sensing in array signal processing based on modulated wideband converter", in *General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI*, 2014, p. 1–4.
- [25] M. Mishali e Y. C. Eldar, "Expected RIP: Conditioning of The modulated wideband converter", in *IEEE Information Theory Workshop, 2009. ITW 2009*, 2009, p. 343–347.
- [26] M. Mishali e Y. C. Eldar, "From Theory to Practice: Sub-Nyquist Sampling of Sparse Wideband Analog Signals", *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, n° 2, p. 375–391, abr. 2010.
- [27] E. Israeli *et al.*, "Hardware calibration of the modulated wideband converter", in *2014 IEEE Global Communications Conference (GLOBECOM)*, 2014, p. 948–953.
- [28] Y. Chen, M. Mishali, Y. C. Eldar, e A. O. Hero, "Modulated wideband converter with non-ideal lowpass filters", in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, p. 3630–3633.
- [29] G. Veld, "On a new compressed sensing paradigm in the modulated wideband converter", in *2013 IEEE EUROCON*, 2013, p. 2140–2145.
- [30] Y. Wang, L. Chen, e Y. Gu, "Quantization reference voltage of the Modulated Wideband Converter", in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, p. 3681–3684.
- [31] S. Zheng e X. Yang, "Wideband spectrum sensing in modulated wideband converter based cognitive radio system", in *2011 11th International Symposium on Communications and Information Technologies (ISCIT)*, 2011, p. 114–119.
- [32] D. E. Bellasi, L. Bettini, C. Benkeser, T. Burger, Q. Huang, e C. Studer, "VLSI Design of a Monolithic Compressive-Sensing Wideband Analog-to-Information Converter", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, n° 4, p. 552–565, dez. 2013.
- [33] M. Trakimas, R. D'Angelo, S. Aeron, T. Hancock, e S. Sonkusale, "A Compressed Sensing Analog-to-Information Converter With Edge-Triggered SAR ADC Core", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, n° 5, p. 1135–1148, maio 2013.
- [34] M. Wakin *et al.*, "A Nonuniform Sampler for Wideband Spectrally-Sparse Environments", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, n° 3, p. 516–529, set. 2012.
- [35] E. Allier, G. Sicard, L. Fesquet, e M. Renaudin, "Asynchronous level crossing analog to digital converters", *Measurement*, vol. 37, n° 4, p. 296–309, jun. 2005.
- [36] M. Ben-Romdhane, C. Rebai, A. Ghazel, P. Desgreys, e P. Loumeau, "Pseudorandom clock signal generation for data conversion in a multistandard receiver", in *3rd*

- International Conference on Design and Technology of Integrated Systems in Nanoscale Era, 2008. DTIS 2008*, 2008, p. 1–4.
- [37] P. Maechler, N. Felber, e A. Burg, “Random sampling ADC for sparse spectrum sensing”, in *Signal Processing Conference, 2011 19th European*, 2011, p. 1200–1204.
 - [38] J. Yoo, S. Becker, M. Monge, M. Loh, E. Candes, e A. Emami-Neyestanak, “Design and implementation of a fully integrated compressed-sensing signal acquisition system”, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, p. 5325–5328.
 - [39] S. R. Becker, “Practical Compressed Sensing: modern data acquisition and signal processing”, California Institute of Technology, California - USA, 2011.
 - [40] J. Yoo *et al.*, “A Compressed Sensing Parameter Extraction Platform for Radar Pulse Signal Acquisition”, *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, n° 3, p. 626–638, set. 2012.
 - [41] M. Mangia, R. Rovatti, G. Setti, e P. Vandergheynst, “Combining Spread Spectrum Compressive Sensing with rakeness for low frequency modulation in RMPI architecture”, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, p. 4146–4150.
 - [42] M. Mangia, F. Pareschi, R. Rovatti, G. Setti, e G. Frattini, “Coping with saturating projection stages in RMPI-based Compressive Sensing”, in *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2012, p. 2805–2808.
 - [43] H. Mamaghanian, N. Khaled, D. Atienza, e P. Vandergheynst, “Design and Exploration of Low-Power Analog to Information Conversion Based on Compressed Sensing”, *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, n° 3, p. 493–501, set. 2012.
 - [44] M. Mishali, Y. C. Eldar, e A. J. Elron, “Xampling: Signal Acquisition and Processing in Union of Subspaces”, *IEEE Transactions on Signal Processing*, vol. 59, n° 10, p. 4719–4734, out. 2011.
 - [45] C. Choudhuri, A. Ghosh, U. Mitra, e S. Pamarti, “Robustness of xampling-based RF receivers against analog mismatches”, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, p. 2965–2968.
 - [46] M. Mishali e Y. C. Eldar, “Xampling: Analog Data Compression”, in *Data Compression Conference (DCC), 2010*, 2010, p. 366–375.
 - [47] M. Mishali, Y. C. Eldar, O. Dounaevsky, e E. Shoshan, “Xampling: Analog to digital at sub-Nyquist rates”, *IET Circuits, Devices Systems*, vol. 5, n° 1, p. 8–20, jan. 2011.
 - [48] T. Michaeli e Y. C. Eldar, “Xampling at the Rate of Innovation”, *IEEE Transactions on Signal Processing*, vol. 60, n° 3, p. 1121–1133, mar. 2012.
 - [49] “Cyclone IV Device Handbook, Volume 1, 2, and 3 - cyclone4-handbook.pdf”, 14-jul-2015. [Online]. Disponível em: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf. [Acessado: 15-jul-2015].
 - [50] T. Technologies, “Terasic - All FPGA Main Boards - Cyclone IV - Altera DE2-115 Development and Education Board”, 11-dez-2016. [Online]. Disponível em: <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=139&No=502&PartNo=2>. [Acessado: 12-dez-2016].
 - [51] V. A. Pedroni, *Digital electronics and design with VHDL*. Amsterdam: Boston : Elsevier Morgan Kaufmann Publishers, 2008.
 - [52] C. Maxfield, *The design warrior's guide to FPGAs: devices, tools, and flows*. Boston: Newnes/Elsevier, 2004.

- [53] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Fourth. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.
- [54] A. Molina-Rueda, F. Uceda-Ponga, e C. F. Uribe, “Extended Period LFSR Using Variable TAP Function”, in *CONIELECOMP 2008, 18th International Conference on Electronics, Communications and Computers*, 2008, 2008, p. 129–132.
- [55] V. L. Reis, E. C. Gurjão, e R. C. S. Freire, “Using synchronism pulse to improve A2I implementations”, in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2015, p. 13–17.
- [56] “CVX: MATLAB Software for Disciplined Convex Programming | CVX Research, Inc.”, 11-dez-2016. [Online]. Disponível em: <http://cvxr.com/cvx/>. [Acessado: 11-dez-2016].
- [57] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar, e P. S. Krishnaprasad, “Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition”, in *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, 1993, p. 40–44.
- [58] V. de L. Reis, P. C. Lobo, E. C. Gurjão, e R. C. S. Freire, “Influence of integrators in the performance of analog-to-information converters”, in *2016 1st International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)*, 2016, p. 118–121.

Apêndice A – AIC em Software.

