

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELÉTRICA
CURSO DE ENGENHARIA ELÉTRICA

MARCO TULIO LOPES GUERREIRO

**ESTUDO E ANÁLISE COMPARATIVA DAS
TÉCNICAS DE *COMPRESSED SENSING* E
SUBAMOSTRAGEM**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2017

MARCO TULIO LOPES GUERREIRO

**ESTUDO E ANÁLISE COMPARATIVA DAS
TÉCNICAS DE *COMPRESSED SENSING* E
SUBAMOSTRAGEM**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Engenharia Elétrica da Coordenação de Engenharia Elétrica - COELT - da Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Fábio Luiz Bertotti

Coorientador: Prof. Dr. Giovanni Guarneri

PATO BRANCO

2017

TERMO DE APROVAÇÃO

O Trabalho de Conclusão de Curso intitulado **ESTUDO E ANÁLISE COMPARATIVA DAS TÉCNICAS DE *COMPRESSED SENSING* E SUBAMOSTRAGEM** do acadêmico **Marco Tulio Lopes Guerreiro** foi considerado **APROVADO** de acordo com a ata da banca examinadora N° 150 de 2017.

Fizeram parte da banca examinadora os professores:

Prof. Dr. Fábio Luiz Bertotti

Prof. Dr. Giovanni Guarneri

Prof. Dr. Fernando José Avancini Schenatto

Prof. Dr. Kleiton de Moraes Sousa

AGRADECIMENTOS

Esse trabalho não seria possível sem a orientação do professor Fábio e do professor Giovanni. Agradeço a ambos pela orientação, paciência e disposição.

Superar os momentos difíceis não seria possível sem a companhia dos meus amigos, Yascara e Augusto. Os momentos de descontração e debates infinitos sempre ajudaram a superar as barreiras encontradas.

Realizar o curso, o qual possibilitou a elaboração deste trabalho de conclusão, não seria possível sem a ajuda e apoio da minha família. Agradeço aos meus pais pelo sacrifício que possibilitou todos esses anos de estudo. Também agradeço aos meus tios e tias, os quais sempre se dispuseram a ajudar em momentos difíceis. Por último, agradeço a minha avó, pelas inúmeras vezes em que ela se dispôs a fazer um café. Sem dúvidas, a escrita do código das simulações seriam diferentes sem esse apoio importante!

RESUMO

A resposta em frequência da bioimpedância tem sido aplicada para determinação da qualidade da carne para consumo e diagnóstico de câncer de pele. No entanto, a aquisição e análise da bioimpedância multifrequencial apresenta desafios, em particular devido as elevadas taxas de amostragem impostas ao conversor analógico-digital, o que pode inviabilizar aplicações de baixo custo devido a necessidade de conversores mais caros. Este trabalho apresenta um estudo e uma análise comparativa entre duas técnicas de aquisição e processamento de sinais, que visam reduzir a taxa de amostragem. A técnica de *compressed sensing* possui uma abordagem diferente dos métodos convencionais, sendo que projeções do sinal são amostradas. A partir do pressuposto que o sinal é esparso no domínio da frequência, a técnica propõe realizar a amostragem do sinal com uma forma de onda que possua uma representação densa no domínio da frequência, sendo o processamento das amostras realizado por algoritmos de otimização. A técnica de subamostragem propõe aplicar uma versão modificada da transformada discreta de Fourier nas amostras de um sinal analógico, em que a frequência do sinal de entrada é conhecida e, dessa forma, é possível a aquisição de fase e magnitude do sinal amostrado com taxas menores do que o estabelecido pelo teorema da amostragem de Nyquist. A comparação entre as técnicas mostra que, dentro dos parâmetros e condições estipuladas para a aquisição de bioimpedância, o sistema de subamostragem leva vantagem em relação a erros de amplitude e fase obtidos em função da frequência do sinal de entrada e da resolução do conversor analógico-digital. Para amplitude, a técnica de subamostragem mostra que a maior parte dos erros concentra-se entre $\pm 1\%$, enquanto que a técnica de *compressed sensing* apresenta erros de até 4%. O trabalho também apresenta uma análise de viabilidade de implementação de *hardware*, a partir da premissa que o sistema será implementado com o microcontrolador STM32F407VGT6. A técnica de subamostragem também apresenta vantagens no quesito de implementação, sendo o custo dos componentes necessários cerca de 10% do custo para implementar a técnica de *compressed sensing*. No entanto, o sistema proposto para a implementação da técnica de *compressed sensing*, o demodulador aleatório, apresenta vantagens para aumentar a frequência de amostragem, possibilitando a aquisição de sinais de frequências mais elevadas que o sistema de subamostragem.

Palavras-chave: Aquisição de Sinais, *Compressed Sensing*, Subamostragem, Transformada Discreta de Fourier.

ABSTRACT

Multi-Frequency Bioelectrical Impedance Analysis has been applied to determine meat quality and skin cancer diagnosis. However, acquisition and characterization of multi-frequency bioimpedance presents challenges, in particular due to the high sampling rates imposed on the analog-to-digital converter, which can lead to impractical low-cost implementations due to the need of more expensive converters. This work presents a study and a comparative analysis between two signal acquisition and processing techniques, which aim to reduce the sampling rate. The compressed sensing technique has an approach that differs from conventional methods, where signal projections are sampled. From the assumption that the signal is sparse in the frequency domain, the technique proposes signal acquisition with a waveform that have a dense representation in the frequency domain and the samples processing is performed by optimization algorithms. The undersampling system proposes applying a modified version of the discrete Fourier transform, where the input signal's frequency is known, which allows acquiring the phase and amplitude of the sampled signal with sampling rates below the Nyquist rate. Comparison between the two techniques shows that, inside the parameters and conditions assumed for bioimpedance acquisition, the undersampling technique has advantages with respect to magnitude and phase errors as a function of input signal frequency and analog-to-digital converter resolution. With respect to magnitude, the undersampling technique shows the majority of the errors lie inside $\pm 1\%$, while the compressed sensing technique presents errors up to 4%. This work also an hardware implementation analysis, based on the assumption that the system will be implemented with the STM32F407VGT6 microcontroller. The undersampling technique also shows advantages, where the costs of the components necessary are about 10% of the compressed sensing technique's implementation cost. However, the hardware for implementation of the compressed sensing technique, the random demodulator, shows advantages to increase the sampling rate, allowing the acquisition of signals with frequencies higher than the undersampling system.

Keywords: Signal Acquisition, Compressed Sensing, Undersampling, Discrete Fourier Transform.

LISTA DE FIGURAS

Figura 1 –	Esquema para a aquisição de sinais analógicos.	13
Figura 2 –	Representação do processo de quantização.	15
Figura 3 –	Sinal $x(t)$, amostras e sinal $s(t)$	17
Figura 4 –	Sistema de subamostragem.	17
Figura 5 –	Sinal $x[t]$ (esquerda) e sua representação no domínio da frequência (direita).	20
Figura 6 –	Função $\delta[t - t_0]$ no domínio do tempo (esquerda) e no domínio da frequência (direita).	21
Figura 7 –	Sistema de aquisição utilizando demodulação aleatória.	23
Figura 8 –	IDLE, ambiente de desenvolvimento utilizado.	27
Figura 9 –	Bloco de simulação para o sistema de subamostragem.	29
Figura 10 –	Esquema para simulação do demodulador aleatório.	29
Figura 11 –	Bloco de simulação para o demodulador aleatório.	30
Figura 12 –	Bloco de reconstrução para o demodulador aleatório.	30
Figura 13 –	Resultados da simulação do sistema de subamostragem.	33
Figura 14 –	Resultados da simulação do sistema de subamostragem, para sinais com frequências múltiplas da frequência de amostragem.	34
Figura 15 –	Erro de amplitude em função da quantidade de amostras.	35
Figura 16 –	Erro de fase em função da quantidade de amostras.	36
Figura 17 –	Resultados da simulação do demodulador aleatório.	37
Figura 18 –	Coefficientes de Fourier obtidos (superior) e sinal reconstruído no tempo (inferior).	37
Figura 19 –	Comparação entre sinal de entrada (azul), reconstrução ideal (laranja) e reconstrução obtida (verde).	38
Figura 20 –	Diferença de fase entre entrada e reconstrução, em função da frequência do sinal de entrada.	39

Figura 21 –	Diferença de fase em função da frequência do sinal de entrada, para frequências de reconstrução diferentes.	40
Figura 22 –	Diferença de fase em função da razão entre frequência do sinal de entrada (f_{in}) e a frequência de reconstrução (f_r).	41
Figura 23 –	SNR obtido, em função da frequência do sinal de entrada.	42
Figura 24 –	Simulação do demodulador aleatório para (a) $\epsilon = 0,001$ e (b) $\epsilon = 0,2$	43
Figura 25 –	SNR obtido em função do erro permitido ϵ	43
Figura 26 –	Comparação do erro de amplitude entre (a) o sistema de subamostragem e (b) o demodulador aleatório.	44
Figura 27 –	Comparação do erro de fase entre (a) o sistema de subamostragem e (b) o demodulador aleatório.	45
Figura 28 –	Erro de magnitude em função da frequência do sinal de entrada e ADC, para (a) o sistema de subamostragem e (b) o demodulador aleatório.	47
Figura 29 –	Erro de fase em função da frequência do sinal de entrada e ADC, para (a) o sistema de subamostragem e (b) o demodulador aleatório.	48
Figura 30 –	<i>Kit</i> de desenvolvimento com o microcontrolador STM32F407VGT6.	50
Figura 31 –	<i>Hardware</i> proposto para o demodulador aleatório.	51
Figura 32 –	Sinal com SNR de aproximadamente 40 dB.	62
Figura 33 –	Histograma (superior) e espectro de frequência (inferior) do sinal de ruído.	63
Figura 34 –	Simulação do ADC.	64
Figura 35 –	Validação do modelo do ADC.	64
Figura 36 –	Sinal (esquerda) e coeficientes de Fourier (direita).	65
Figura 37 –	Resposta do filtro ao impulso.	66

LISTA DE TABELAS

Tabela 1 –	Parâmetros para a simulação do sistema de subamostragem. . . .	32
Tabela 2 –	Resultados obtidos para a simulação do sistema de subamostragem.	33
Tabela 3 –	Resultados da simulação com uma frequência de amostragem de 2,35 MHz.	34
Tabela 4 –	Parâmetros para a simulação do demodulador aleatório.	36
Tabela 5 –	Componentes propostos para o demodulador aleatório.	52

SUMÁRIO

1	INTRODUÇÃO	10
1.1	JUSTIFICATIVA	10
1.2	OBJETIVO GERAL	11
1.2.1	Objetivos específicos	11
1.3	ORGANIZAÇÃO DO TRABALHO	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	AQUISIÇÃO DE SINAIS	13
2.1.1	Ruído em sinais	14
2.1.2	Conversor Analógico-Digital	15
2.2	SUBAMOSTRAGEM	17
2.2.1	Sistema de subamostragem	17
2.2.2	Modelagem do sistema de subamostragem	18
2.2.2.1	Etapa de subamostragem	18
2.2.2.2	Processamento	19
2.3	<i>COMPRESSED SENSING</i>	19
2.3.1	Sinais compressíveis e incoerência	20
2.3.2	Reconstrução do sinal	21
2.3.3	Propriedade de isometria restrita	22
2.3.4	Demodulador aleatório	23
2.3.4.1	Demodulador	23
2.3.4.2	Filtro passa-baixas	24
2.3.4.3	Conversor Analógico-Digital	25
2.3.4.4	Transformação de base	25
2.3.4.5	Matriz A	25
3	MATERIAIS E MÉTODOS	26
3.1	MATERIAIS	26
3.1.1	Python e ambiente de desenvolvimento	27
3.1.1.1	Pacotes adicionais	27

3.2	MÉTODOS	28
3.2.1	Validação dos modelos	28
3.2.2	Sistema de subamostragem	28
3.2.3	Demodulador aleatório	29
3.2.4	Análise de desempenho	31
4	RESULTADOS E DISCUSSÃO	32
4.1	SIMULAÇÕES	32
4.1.1	Sistema de subamostragem	32
4.1.1.1	Erro de amplitude e fase	35
4.1.2	Demodulador aleatório	36
4.1.2.1	Diferença de fase	38
4.1.2.2	Qualidade da reconstrução	41
4.1.2.3	Parâmetro de regularização	42
4.2	COMPARAÇÕES	44
4.2.1	Erro em função do sinal de entrada	44
4.2.2	Erro em função da resolução do ADC	46
4.2.3	Tempo de processamento	49
4.3	VIABILIDADE DE HARDWARE	49
4.3.1	<i>Hardware</i> para o sistema de subamostragem	50
4.3.2	<i>Hardware</i> para o demodulador aleatório	51
4.3.3	Comparação	52
5	CONCLUSÕES	54
5.1	TRABALHOS FUTUROS	55
	REFERÊNCIAS	57
	APÊNDICE A - VALIDAÇÃO DOS MODELOS	62
A.1	RUÍDO	62
A.2	CONVERSOR ANALÓGICO-DIGITAL	63
A.3	BLOCO DE PROCESSAMENTO DO SISTEMA DE SUBAMOSTRAGEM ..	65
A.4	RESPOSTA DO FILTRO	65

1 INTRODUÇÃO

1.1 JUSTIFICATIVA

As aplicações de bioimpedância abrangem desde a determinação de qualidade do leite e qualidade de carne para consumo humano à diagnóstico de câncer (BERTEMES-FILHO; NEGRI; PATERNO; 2012, BERTOTTI, 2010, AYLLÓN; SEOANE; GIL-PITA, 2009). No entanto, a medição da bioimpedância apresenta desafios. A análise multifrequencial de bioimpedância (MF-BIA, do inglês *Multi-Frequency Bioimpedance Analysis*) requer a sua medição em diversos pontos do espectro (LING et al., 2011), sendo que a caracterização de bioimpedância consiste em analisá-la em frequências que variam entre 1 kHz e 1 MHz.

O procedimento básico para a medição de uma impedância consiste em inserir um sinal de corrente, com magnitude e frequência conhecida, e medir o sinal de tensão resultante. O sinal inserido também poder ser um sinal de tensão. Nesse caso, a corrente é medida. O processamento do sinal pode ser feito digitalmente, utilizando a Transformada Discreta de Fourier (DFT, do inglês *Discrete Fourier Transform*) para analisar a magnitude e fase dos sinais da impedância sob teste. Porém, antes de realizar o processamento digital, o sinal deve ser amostrado.

Uma das abordagens clássicas utilizadas em processamento digital de sinais é baseada no teorema de Shannon-Nyquist, que define a taxa de amostragem mínima para obter toda a informação do sinal amostrado. Nessa abordagem, se F_s é a maior componente de frequência contida em um sinal, então, é necessário amostrá-lo com uma frequência superior a $2F_s$ (LATHI, 2000).

Um dos principais componentes na aquisição de sinais é o conversor analógico-digital ou ADC (*Analog to Digital Converter*), sendo responsável por realizar a amostragem do sinal analógico e a conversão de cada amostra em uma palavra digital (LE et al., 2005). A aquisição de sinais das frequências mais altas, necessárias para caracterizar a bioimpedância, requer taxas de amostragem elevadas. Conversores com alta taxa de amostragem e com resolução aceitável para aplicações como a aquisição de bioimpedância apresentam um custo elevado, o que torna aplicações em dispositivos de baixo custo inviáveis.

Trabalhos iniciados na última década tem buscado técnicas alternativas de aquisição, visando reduzir a taxa de amostragem (DAPONTE et al., 2017, MARVASTI et al., 2012, DUARTE; ELDAR, 2011, MARAVIC; VETTERLI, 2005, AMARAL, 2004). Embora ADCs com alta taxa de amostragem ainda apresentam custo elevado, ADCs com largura de banda elevada tem possibilitado o uso da técnica de subamostragem (AMARAL, 2004), que propõe a amostragem do sinal com taxas abaixo da taxa de Nyquist. No entanto, a técnica da subamostragem requer um conversor com um tempo de aquisição reduzido. Além disso, recentemente, uma técnica de amostragem chamada *compressed sensing* tem revolucionado o campo de processamento digital de sinais (DUARTE; ELDAR, 2011). A técnica, inicialmente proposta para sinais discretos, possibilita a redução da taxa de amostragem a partir de um método de aquisição diferenciado. Porém, a técnica requer um processamento de sinais mais elaborado se comparado a técnica de subamostragem, além de apresentar desafios para a implementação em sinais analógicos.

Para a redução da taxa de amostragem dos conversores, ambas as técnicas apresentam vantagens e desvantagens. Dessa forma, neste trabalho, um estudo comparativo é realizado, visando identificar a técnica que apresenta mais vantagens para a implementação de um sistema para aquisição de bioimpedância. Para avaliar as técnicas, erros de amplitude e fase são comparados, em função da frequência do sinal de entrada e da resolução do conversor. Além disso, uma análise de viabilidade de *hardware* também é realizada, visando comparar a viabilidade de implementação prática dos sistemas que utilizam as técnicas.

1.2 OBJETIVO GERAL

O trabalho consiste em realizar uma análise comparativa por meio de simulação entre dois sistemas de aquisição de sinais, um deles empregando a técnica de *compressed sensing* e outro utilizando a técnica de subamostragem.

1.2.1 Objetivos específicos

- Caracterizar as técnicas de *compressed sensing* e subamostragem.
- Modelar o sistema de aquisição utilizando subamostragem em uma ferramenta de simulação.
- Modelar o sistema de aquisição utilizando *compressed sensing* em uma ferramenta de simulação.

- Simular os sistemas considerando sinais periódicos, com características diversas de amplitude e fase.
- Analisar os resultados, incluindo análise de erros de amplitude e fase em função da frequência do sinal de entrada e da resolução do conversor analógico-digital.
- Analisar a viabilidade da implementação dos sistemas, considerando quesitos como disponibilidade de materiais, custo e complexidade de placas de circuito impresso.

1.3 ORGANIZAÇÃO DO TRABALHO

A organização deste trabalho foi realizada da seguinte forma. O capítulo 1 apresentou uma introdução ao problema e as técnicas estudadas, bem como os objetivos. No capítulo 2, a fundamentação teórica do trabalho é feita, detalhando a técnica de subamostragem e a técnica de *compressed sensing*, apresentando os modelos de cada sistema considerado para a simulação. Em seguida, no capítulo 3, os materiais e métodos são apresentados, indicando as ferramentas utilizadas para o desenvolvimento e a metodologia utilizada para a implementação da simulação e comparação entre os sistemas em estudo. No capítulo 4, as simulações e comparação de desempenho dos sistemas são discutidos, bem como uma análise de viabilidade de *hardware*. Por fim, o capítulo 5 apresenta as principais conclusões deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo tem como objetivo apresentar a base teórica do trabalho. Primeiro, princípios de aquisição de sinais, comuns a ambos os métodos de amostragem, são discutidos. Em seguida, o conceito e a modelagem do sistema de subamostragem são abordados. Por último, a técnica de *compressed sensing* e suas restrições são discutidas, juntamente a modelagem do demodulador aleatório.

2.1 AQUISIÇÃO DE SINAIS

Sinais provenientes do mundo real tem natureza analógica. Como o sistema de subamostragem e a técnica de *compressed sensing* utilizam processamento digital de sinais, é necessário realizar o condicionamento e aquisição dos sinais analógicos. Um esquema genérico utilizado na aquisição de sinais pode ser visto na Figura 1.

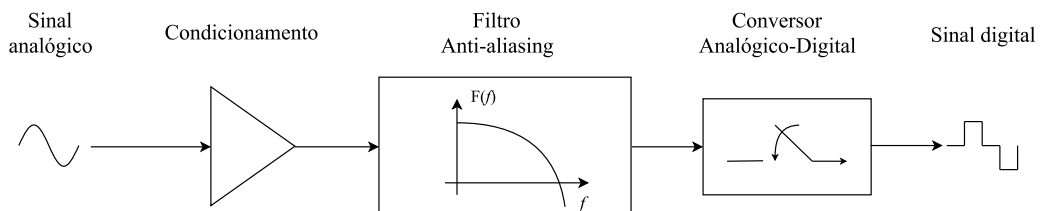


Figura 1 – Esquema para a aquisição de sinais analógicos.

Fonte: Autoria própria

No esquema indicado na Figura 1, o sinal analógico passa por um bloco de condicionamento, que é responsável por ações como aplicar um ganho ao sinal, somá-lo a outro sinal, entre outros. Essa etapa é necessária em casos em que o sinal analógico possui amplitude reduzida ou quando o sinal na entrada do conversor analógico digital não pode assumir valores negativos, para não danificá-lo.

Em seguida, um filtro *anti-aliasing* é responsável por limitar a largura de banda do sinal medido (WU; DEY; CHEN, 2016). Limitar a largura de banda do sinal de entrada é fundamental, uma vez que se o sinal contém frequências maiores do que o dobro da frequência de amostragem, o sinal amostrado aparece com características de frequência diferente do sinal analógico. Dessa forma, o filtro *anti-aliasing* é projetado baseado na frequência de amostragem do conversor, de forma que a frequência de corte

do filtro não seja maior do que metade da frequência de amostragem do conversor, afim de respeitar a taxa de Nyquist. Na Figura 1, o filtro é posicionado após o condicionamento e imediatamente antes do conversor analógico-digital. No entanto, a ordem das etapas poderia ser invertida, sendo o bloco de condicionamento imediatamente antes do conversor analógico-digital e após o filtro.

Após o filtro, um conversor analógico-digital realiza a amostragem e a quantização do sinal. No processo de amostragem, o conversor retira amostras do sinal, geralmente em intervalos regulares. Cada amostra é, então, quantizada. Nesse processo, a amostra tomada pelo conversor é convertida para uma palavra digital. Na saída do conversor analógico-digital, o sinal digital pode ser processado em um sistema computacional, utilizando algoritmos de processamento digital de sinais.

Cada uma das etapas de aquisição da Figura 1 introduz ruído no sinal sendo medido, alterando suas características. O ruído pode ser atribuído a diversos fatores, como não linearidades contidas no conversor analógico-digital, resposta em frequência de amplificadores operacionais, variação no comportamento dos componentes conforme a temperatura, entre outros.

A seguir, o efeito de ruídos no sinal analógico e o bloco do conversor analógico-digital são modelados. O bloco de condicionamento não é modelado, pois considera-se que o sinal na entrada dos sistemas em estudo já estão condicionados. Como o bloco do filtro de *anti-aliasing* não é incluído no sistema de subamostragem mas faz parte do sistema do demodulador aleatório, o seu modelo é apresentado apenas em seções posteriores.

2.1.1 Ruído em sinais

Para modelar a adição de ruído a sistemas, é comum utilizar o modelo AWGN (*Additive White Gaussian Noise*). Nesse modelo, considera-se que as diversas fontes de ruído presentes no sistema possuem natureza aleatória (VERTELLI et al., 2014). Dessa forma, o ruído que é somado ao sinal é proveniente de uma distribuição normal e sua energia é distribuída no espectro, não sendo concentrada em nenhuma frequência.

Para representar a quantidade de ruído presente no sinal, utiliza-se o parâmetro SNR (*Signal-to-Noise Ratio*). O SNR define a razão logarítmica entre a potência do sinal e a potência do ruído (PAULUZZI; BEAULIEU, 2000), sendo expresso por:

$$\text{SNR} = 10 \log_{10} \left(\frac{P_s}{P_r} \right), \quad (1)$$

em que P_s é a potência do sinal, P_r é a potência do ruído e o SNR é dado em decibel

(dB). A potência de um sinal discreto $s[k]$ é uma maneira de quantificar seu tamanho. No caso em que $s[k]$ pode ser representado de forma periódica, a potência do sinal pode ser obtida com:

$$P = \frac{1}{N} \sum_{n=0}^N (s[k])^2, \quad (2)$$

em que N é a dimensão do sinal.

O processo de introduzir um ruído com um determinado SNR ao sinal discreto consiste em obter um vetor cuja potência satisfaça a relação de (1). O sinal de ruído é obtido aplicando um fator de escala K em um vetor que é gerado com valores aleatórios de uma distribuição normal com média zero e desvio padrão igual a um. O fator de escala é obtido a partir de (3).

$$K = \sqrt{P_r}. \quad (3)$$

2.1.2 Conversor Analógico-Digital

O conversor analógico-digital tem como função realizar a conversão de um sinal analógico para uma palavra digital (LE et al., 2005). A quantidade de palavras digitais distintas que o conversor é capaz de produzir é dada pela sua resolução, fornecida em bits. Um ADC de N bits possui 2^N palavras digitais para a conversão. A referência do conversor serve como um fundo de escala e, idealmente, indica o valor do sinal na entrada que corresponde à última palavra de conversão.

Neste trabalho, a modelagem do ADC consiste em modelar o processo de conversão, chamado de quantização (PENG et al., 2004, WALDEN, 1999). O processo ocorre conforme indicado na Figura 2.

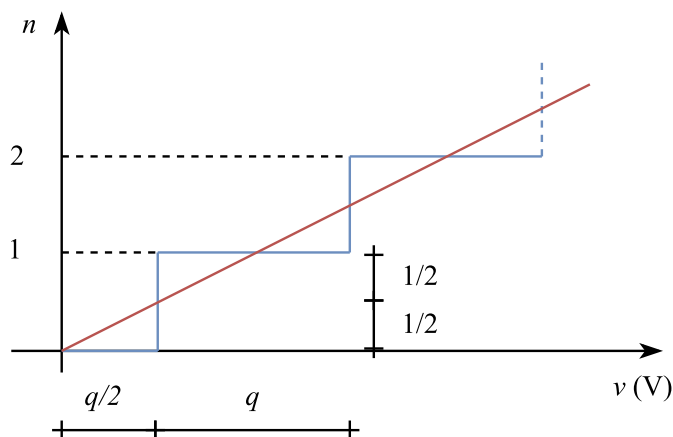


Figura 2 – Representação do processo de quantização.
Fonte: Adaptado de Peng et al., 2004.

Na Figura 2, a conversão realizada pelo conversor é indicada pela função em degrau, que representa um ADC de N bits, com uma tensão de referência v_{ref} . A função de transferência de um ADC com a mesma referência, mas com uma resolução infinita, é indicada pela reta. A equação da função que relaciona a tensão na entrada v e o resultado da conversão n é dada por:

$$n = \frac{2}{q}v, \quad (4)$$

em que:

$$q = \frac{v_{ref}}{2^N}. \quad (5)$$

A partir da Figura 2, verifica-se que o processo de conversão do sinal analógico para uma palavra digital ocorre da seguinte maneira. Calculando n a partir de (4), o resultado da conversão segue a seguinte forma:

$$\begin{aligned} 0 \leq n < 1/2 &\rightarrow 0 \\ 1/2 \leq n < 3/2 &\rightarrow 1 \\ 3/2 \leq n < 5/2 &\rightarrow 2 \\ &\vdots \end{aligned} \quad (6)$$

Multiplicando o resultado em (6) por 2, obtém-se:

$$\begin{aligned} 0 \leq 2n < 1 &\rightarrow 0 \\ 1 \leq 2n < 3 &\rightarrow 2 \\ 2 \leq 2n < 3 &\rightarrow 4 \\ &\vdots \end{aligned} \quad (7)$$

Dessa forma, é possível observar que, para a conversão, apenas a parte inteira do resultado $2n$ é necessária. A partir do resultado em (7), nota-se que, quando a parte inteira do resultado $2n$ for um número par, a parte inteira é o próprio resultado da conversão. Por outro lado, se o resultado $2n$ for ímpar, o valor imediatamente acima é o resultado da conversão. O resultado final é obtido dividindo os resultados anteriores por 2.

Embora existam outros processos de quantização, o processo indicado na Figura 2 é um dos métodos mais utilizados, uma vez que apresenta um erro de quantização de, no máximo, $\pm 1/2$ (TOCCI; WIDMER; MOSS, 2010).

2.2 SUBAMOSTRAGEM

A técnica de subamostragem baseia-se no fato de que o sinal resultante de uma amostragem com uma taxa menor do que a taxa de Nyquist possui as mesmas características de fase e amplitude do sinal original (DUDYKEYVYCH et al., 2001, AMARAL, 2004). A Figura 3 mostra um sinal $x(t)$ de 100 Hz amostrado a uma frequência de 90 Hz, indicado pelo sinal $s[k]$. É possível notar que, um terceiro sinal $s(t)$ com uma frequência de 10 Hz e mesma amplitude e fase de $x(t)$, se ajusta perfeitamente às amostras $s[k]$. O processamento do sinal amostrado tira vantagem do fato de que a frequência do sinal amostrado é conhecida. O processamento consiste em aplicar uma versão modificada da transformada discreta de Fourier diretamente nas frequências que são conhecidas. Por esse motivo, é fundamental conhecer a frequência do sinal amostrado.

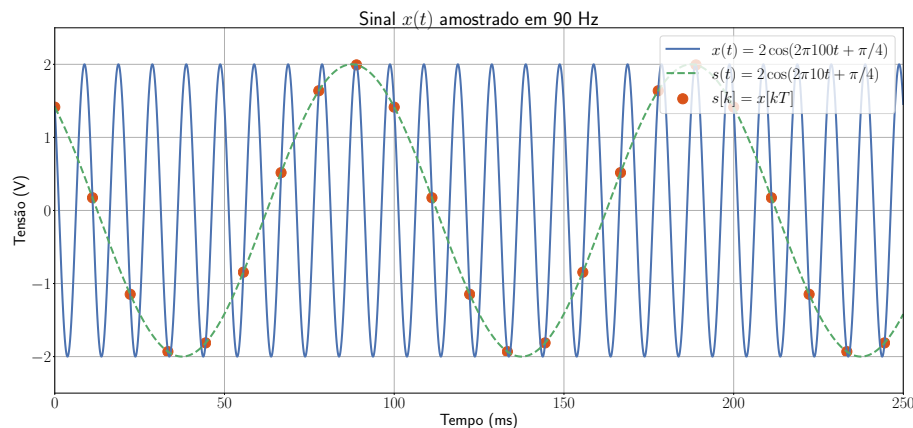


Figura 3 – Sinal $x(t)$, amostras e sinal $s(t)$.

Fonte: Autoria própria

2.2.1 Sistema de subamostragem

O sistema de subamostragem é composto pelo sinal de entrada $x(t)$, ruído $r(t)$, etapa de subamostragem, ADC e etapa de processamento, conforme indicado na Figura 4. Opta-se por realizar a amostragem e a quantização dos sinais em blocos diferentes para prover uma maior flexibilidade no sistema de amostragem.

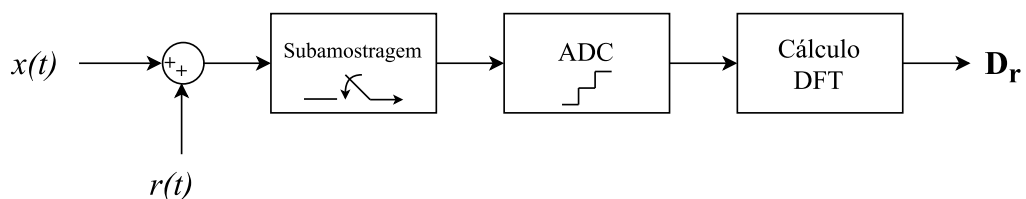


Figura 4 – Sistema de subamostragem.

Fonte: Autoria própria

O bloco de subamostragem é responsável por tomar as amostras do sinal, conforme uma frequência de amostragem. Em seguida, o sinal passa por um ADC, que é responsável por realizar a quantização do sinal amostrado, enquanto que o bloco do cálculo da DFT é responsável por aplicar uma versão modificada da DFT nas amostras. Na saída, o bloco de processamento fornece os coeficientes de Fourier (\mathbf{D}_r), com módulo e fase, nas frequências desejadas.

2.2.2 Modelagem do sistema de subamostragem

Esta seção apresenta os modelos dos blocos de subamostragem e DFT do sistema de subamostragem. O bloco do ADC que realiza a quantização do sinal e a adição de ruído foram discutidos previamente.

2.2.2.1 Etapa de subamostragem

O processo de amostragem consiste em retirar algumas amostras de um sinal de entrada. No caso em que a frequência de amostragem é fixa, as amostras do sinal são tomadas em intervalos regulares. Em um sistema discreto, o sinal resultante pode ser definido por:

$$y[k'] = x[Fk], \quad (8)$$

em que F é a taxa de amostragem. Como os índices k' e k devem ser números inteiros, a taxa de amostragem também deve ser um número inteiro. Se f_d é a frequência de discretização do sinal x e f_s é a frequência de amostragem, a taxa de amostragem é dada por:

$$F = \frac{f_d}{f_s}. \quad (9)$$

Dessa forma, o sinal discreto $y[k']$ consiste em amostras do sinal $x[k]$, tomadas em intervalos uniformes, sendo o intervalo entre amostras definido pela relação entre f_d e f_s . A título de exemplo, considere uma frequência de discretização de 100 Hz e uma frequência de amostragem de 10 Hz. Nesse caso, a relação f_d/f_s é 10 e, portanto, o vetor $y[k']$ é dado por:

$$\begin{aligned} y[0] &= x[0] \\ y[1] &= x[10] \\ y[2] &= x[20] \\ &\vdots \end{aligned} \quad (10)$$

2.2.2.2 Processamento

O processamento do sinal consiste em aplicar uma versão modificada da DFT nas amostras adquiridas, calculando os coeficientes de Fourier somente nas frequências do sinal que são conhecidas. As partes real e imaginária dos coeficientes de Fourier podem ser calculados a partir de (11) e (12) (AMARAL, 2004).

$$x = \sum_{n=0}^N s[n] \cos(2\pi n f / f_s), \quad (11)$$

$$y = \sum_{n=0}^N s[n] \sin(2\pi n f / f_s). \quad (12)$$

em que x e y são as componentes reais e imaginárias do coeficiente de Fourier, $s[n]$ são as amostras do sinal, f é a frequência do sinal e f_s é a frequência de amostragem. A partir das componentes reais e imaginárias, o módulo ($|X|$) e a fase (θ) dos coeficientes de Fourier são obtidos por:

$$|X| = \sqrt{x^2 + y^2}, \quad (13)$$

$$\tan \theta = -\frac{y}{x}. \quad (14)$$

2.3 COMPRESSED SENSING

O princípio da técnica de *compressed sensing* consiste em obter m amostras de um sinal e em seguida estimar um sinal de dimensão n que possa ter gerado as amostras observadas. Dessa forma, a princípio, é possível amostrar um sinal com uma taxa menor que a taxa de Nyquist.

A técnica baseia-se em duas premissas acerca do sinal sendo medido e do método de aquisição utilizado (RAGHEB et al., 2008). A primeira premissa é que o sinal sendo inferido é compressível, enquanto que a segunda premissa é que o método de aquisição utilizado é incoerente com o sinal sendo medido. Nesta seção, os conceitos básicos de *compressed sensing* são discutidos, bem como o processo de reconstrução do sinal e algumas restrições e requisitos. Em seguida, a técnica do demodulador aleatório e sua modelagem são apresentadas.

2.3.1 Sinais compressíveis e incoerência

A técnica de *compressed sensing* baseia-se no fato de que o sinal a ser medido possui uma representação esparsa em alguma base. Além disso, a técnica não trabalha diretamente com as amostras do sinal; a técnica processa projeções do sinal, tomadas a partir de um processo de amostragem incoerente com a base esparsa do sinal.

Sinais compressíveis são aqueles que possuem uma representação esparsa em alguma base (CANDÈS; WAKING, 2009). Por exemplo, considere o sinal discreto $x[t]$ representado no domínio do tempo e no domínio da frequência, indicado na Figura 5. O sinal $x[t]$ é uma onda senoidal de 60 Hz e amplitude 2, amostrada em 3000 Hz. A cada ciclo de $x[t]$ existem 500 pontos, sendo que muitos desses pontos possuem valores distintos. No entanto, considerando o sinal $x[t]$ no domínio da frequência, verifica-se que o sinal possui apenas um ponto diferente de zero, que representa a sua magnitude. Dessa forma, o sinal $x[t]$ pode ser considerado esparsa no domínio da frequência, pois embora seja necessário muitos pontos para representá-lo no domínio do tempo, poucos coeficientes diferentes de zero são necessários para representá-lo no domínio da frequência.

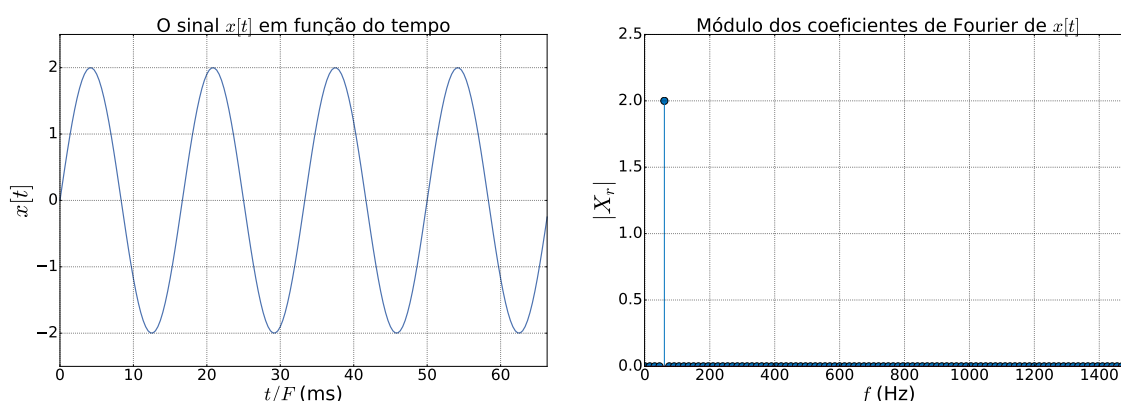


Figura 5 – Sinal $x[t]$ (esquerda) e sua representação no domínio da frequência (direita).

Fonte: Autoria própria

A **incoerência** entre o processo de amostragem e a base esparsa do sinal significa que a correlação entre as formas de onda do processo de amostragem e as formas de onda da base esparsa do sinal é baixa (CANDÈS; WAKING, 2009). Em outras palavras, incoerência significa que a forma de onda de amostragem possui uma representação densa na base esparsa do sinal. A Figura 6 mostra a função delta de Dirac ($\delta[t - t_0]$) e sua representação no domínio da frequência. É possível observar que o sinal $\delta[t - t_0]$ possui uma representação densa no domínio da frequência, pois a maioria de seus coeficientes possuem valor diferente de zero.

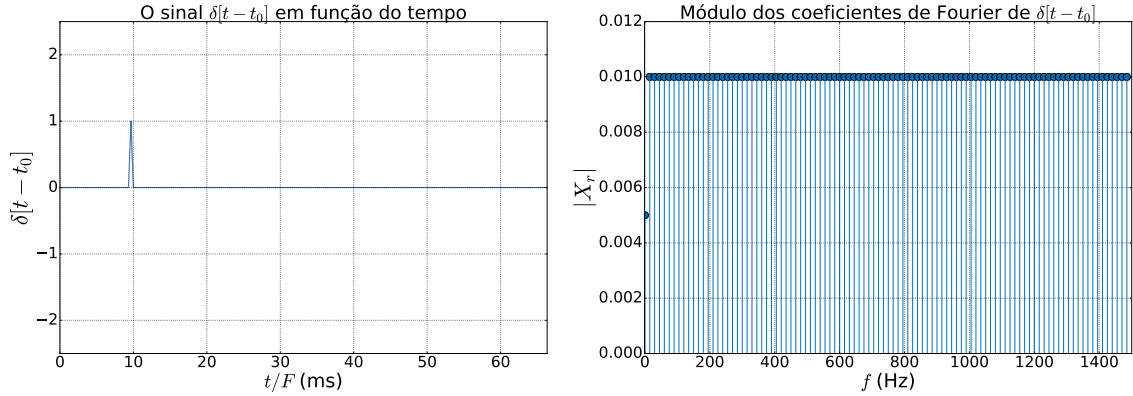


Figura 6 – Função $\delta[t - t_0]$ no domínio do tempo (esquerda) e no domínio da frequência (direita).

Fonte: Autoria própria

2.3.2 Reconstrução do sinal

Considerando um sinal $x[t]$ que é esparso, a sua reconstrução consiste em estimar um sinal $\tilde{x}[t]$ que possa ter gerado as amostras observadas, a partir de um processo de aquisição conhecido e incoerente com o sinal $x[t]$. O processo de reconstrução (estimação) do sinal resume-se em resolver um problema de otimização, envolvendo os conceitos de norma ℓ_1 e ℓ_2 de um vetor.

As normas ℓ_1 e ℓ_2 possuem papel importante no erro e na esparsidade do resultado (YIN; ESSER; XIN, 2014). A norma ℓ_2 de um vetor \mathbf{x} de dimensão n é mais conhecida como sendo o seu módulo (ou comprimento). A norma ℓ_2 de um vetor de dimensão n é dada por:

$$\|\mathbf{x}\|_{\ell_2} = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}. \quad (15)$$

A norma ℓ_1 de um vetor \mathbf{x} é definida como sendo a soma dos valores absolutos:

$$\|\mathbf{x}\|_{\ell_1} = |x_1| + |x_2| + \dots + |x_n|. \quad (16)$$

O processo de reconstrução do sinal consiste em encontrar um vetor que possua a menor norma ℓ_1 e que se ajuste as amostras observadas. Seja \mathbf{x} o vetor que representa o sinal $x[t]$, com dimensão n ($\mathbf{x} \in \mathbb{R}^n$) e \mathbf{y} um vetor com algumas amostras do sinal, com dimensão m ($\mathbf{y} \in \mathbb{R}^m$). O processo de amostragem, que gera o vetor \mathbf{y} a partir do vetor \mathbf{x} , pode ser representado por uma matriz Φ de dimensão $m \times n$:

$$\mathbf{y} = \Phi \mathbf{x}, \quad (17)$$

de tal forma que $n > m$, uma vez que n representa a dimensão do sinal e m representa a quantidade de amostras que são obtidas.

O sinal \mathbf{x} pode ser representado em outra base. Por exemplo, se \mathbf{x} representa um sinal no tempo, é possível representá-lo no domínio da frequência com os coeficientes de Fourier \mathbf{D} ($\mathbf{D} \in \mathbb{C}^n$). Dessa forma, o vetor \mathbf{x} pode ser reescrito como:

$$\mathbf{x} = \mathbf{\Psi}\mathbf{D}, \quad (18)$$

em que a matriz $\mathbf{\Psi}$ ($\mathbf{\Psi} \in \mathbb{C}^{n \times n}$) representa a transformada inversa de Fourier.

O problema consiste em obter \mathbf{x} a partir de \mathbf{y} , ou seja, a partir de (17), o problema consiste em determinar \mathbf{x} conhecendo \mathbf{y} e $\mathbf{\Phi}$, o que resulta na solução de um sistema linear indeterminado (uma vez que $n > m$). Como o sistema é indeterminado, existe uma quantidade infinita de soluções $\tilde{\mathbf{x}}$, que satisfazem a relação $\mathbf{y} = \mathbf{\Phi}\tilde{\mathbf{x}}$. A técnica de *compressed sensing* explora o fato que se o sinal é compressível e o processo de amostragem é incoerente com o sinal, é possível recuperar sua representação esparsa $\tilde{\mathbf{D}}$ que satisfaça a relação $\mathbf{y} = \mathbf{\Phi}\tilde{\mathbf{D}}$, resolvendo o seguinte problema (CANDÈS et al., 2006):

$$\min_{\tilde{\mathbf{D}} \in \mathbb{C}^n} \|\tilde{\mathbf{D}}\|_{\ell_1} \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\tilde{\mathbf{D}}, \quad (19)$$

em que $\mathbf{A} = \mathbf{\Phi}\mathbf{\Psi}$. Em outras palavras, entre as possíveis soluções do sistema indeterminado proposto, a solução desejada é aquela que possui a menor norma ℓ_1 , respeitando a igualdade $\mathbf{y} = \mathbf{A}\tilde{\mathbf{D}}$. Resolvendo o problema de minimização, é possível encontrar uma solução exata com alta probabilidade.

No caso em que existe ruído no sistema, de tal forma que o sinal \mathbf{x} é corrompido, o processo de reconstrução é diferente. Ao invés de usar uma restrição em que $\mathbf{y} = \mathbf{A}\tilde{\mathbf{D}}$, utiliza-se uma restrição que permite um erro. Dessa forma, quando o sinal \mathbf{x} for corrompido durante a aquisição, sua reconstrução consiste em minimizar a norma ℓ_1 de sua representação esparsa sujeito a um erro (CANDÈS et al., 2006):

$$\min_{\tilde{\mathbf{D}} \in \mathbb{C}^n} \|\tilde{\mathbf{D}}\|_{\ell_1} \quad \text{s.t.} \quad \|\mathbf{A}\tilde{\mathbf{D}} - \mathbf{y}\|_{\ell_2} \leq \epsilon. \quad (20)$$

2.3.3 Propriedade de isometria restrita

A matriz \mathbf{A} , que representa o processo de aquisição e a transformação de base do sinal, possui restrições quanto a sua forma.

A propriedade de isometria restrita (RIP, do inglês *Restricted Isometry Property*) é uma restrição na matriz \mathbf{A} . A RIP pode ser entendida como sendo a ortogonalidade entre as colunas da matriz \mathbf{A} (VIDYASAGAR, 2016). Como a matriz \mathbf{A} possui mais colunas do que linhas, as colunas não podem ser exatamente ortogonais. A RIP re-

quer que qualquer sub conjunto de colunas de \mathbf{A} seja aproximadamente ortogonais. Nesse caso, é possível recuperar um vetor $\tilde{\mathbf{D}}$ a partir das amostras \mathbf{y} , resolvendo o problema de otimização apresentado em (20).

2.3.4 Demodulador aleatório

A técnica de *compressed sensing* é formulada apenas para sinais discretos. No entanto, é possível utilizar a técnica para aquisição de sinais contínuos. A técnica estudada neste trabalho é a técnica do demodulador aleatório, que possui o esquema representado na Figura 7 (PANKIEWICZ et al., 2011, LASKA et al., 2007). O demodulador aleatório consiste em multiplicar o sinal de entrada, $x(t)$, por uma sequência aleatória que varia entre +1 e -1. Em seguida, o sinal é filtrado com um filtro passa-baixas e amostrado por um ADC, resultando em um vetor de amostras $y[k]$.

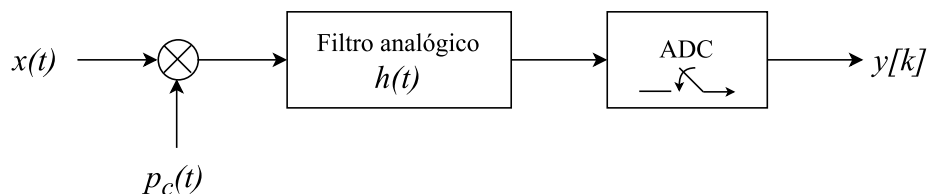


Figura 7 – Sistema de aquisição utilizando demodulação aleatória.

Fonte: Adaptado de Laska et al., 2007.

A matriz \mathbf{A} utilizada na reconstrução é composta por diversas matrizes: uma matriz que representa a multiplicação por valores que alternam aleatoriamente entre +1 e -1, a matriz que representa resposta do filtro, a matriz que representa o processo de aquisição do ADC e a matriz que representa a transformação de base do sinal. Em seguida, a modelagem da matriz de cada etapa é discutida.

2.3.4.1 Demodulador

Nesta etapa, o sinal $x(t)$ é multiplicado por uma sequência que alterna aleatoriamente entre +1 e -1. O processo de multiplicação tem como objetivo produzir uma matriz que atenda a RIP. Além disso, como consequência, o processo de multiplicação espalha o conteúdo do sinal $x(t)$ no espectro de frequência (RAGHEB et al; 2008).

Considerando um sinal discreto $x[t]$ e um vetor $p_c[t]$, ambos de dimensão n , o processo de multiplicação pode ser escrito como:

$$\tilde{\mathbf{x}} = \mathbf{P}_c \mathbf{x}, \quad (21)$$

em que:

$$\mathbf{P}_c = \text{diag}(\mathbf{p}_c). \quad (22)$$

2.3.4.2 Filtro passa-baixas

O filtro utilizado no demodulador aleatório é um filtro passa-baixas, que tem como função evitar o fenômeno de *aliasing* antes do processo de amostragem. Dessa forma, a frequência de corte do filtro é pelo menos a metade da frequência de amostragem do ADC. Por exemplo, se o sinal na saída do filtro será amostrado a 100 kHz, a frequência de corte do filtro não deve ser maior do que 50 kHz.

O filtro empregado na simulação do demodulador aleatório é um filtro passa-baixa de quarta ordem, seguindo os coeficientes normalizados de Butterworth (MANCINI, 2002). A função de transferência do filtro é dada por (LUDWIG; BOGDANOV, 2008):

$$\frac{Y(s)}{X(s)} = \frac{1}{(s^2 + 0.765s + 1)(s^2 + 1.848s + 1)}. \quad (23)$$

O sinal na saída do filtro pode ser obtido com a convolução entre o sinal de entrada e a resposta do filtro ao impulso (LATHI, 2000), sendo que a resposta ao impulso pode ser obtida aplicando a transformada inversa de Laplace em (23), com $X(s) = 1$ (para desenvolvimento e validação, ver apêndice A.4). Dessa forma, conhecendo a resposta do filtro ao impulso, é possível simular sua resposta para um sinal arbitrário.

Para sinais discretos, o sinal de saída do filtro pode ser implementado com um cálculo matricial. Seja $h[t]$ a resposta discretizada do filtro e $x[t]$ o sinal de entrada, a resposta $y[t]$ do filtro pode ser obtida por:

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \quad (24)$$

em que \mathbf{H} representa a ação do filtro e é definida como:

$$\mathbf{H} = \begin{bmatrix} h[0] & 0 & 0 & \dots & 0 \\ h[1] & h[0] & 0 & \dots & 0 \\ h[2] & h[1] & h[0] & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ h[N-1] & h[N-2] & h[N-3] & \dots & h[0] \end{bmatrix}. \quad (25)$$

2.3.4.3 Conversor Analógico-Digital

O ADC tem como função amostrar e quantizar o sinal, fornecendo as amostras digitais em sua saída. As amostras, por sua vez, são utilizadas no algoritmo para a reconstrução do sinal.

Para a modelagem do demodulador aleatório, considera-se que o ADC realiza apenas a amostragem do sinal. O processo de quantização é deconsiderado, visando simplificar o modelo matricial do ADC. A matriz do processo de amostragem do sinal é modelado como o sinal sendo amostrado, $x[t]$, multiplicado por um impulso no instante da amostragem. Dessa forma, a matriz Δ que representa o processo de amostragem é definida como:

$$\Delta = \begin{bmatrix} \dots & \delta[t - t_0] & \dots \\ \dots & \delta[t - t_1] & \dots \\ \dots & \delta[t - t_2] & \dots \\ \vdots & \ddots & \vdots \\ \dots & \delta[t - t_{m-1}] & \dots \end{bmatrix}. \quad (26)$$

2.3.4.4 Transformação de base

Neste trabalho, estuda-se a reconstrução de sinais medidos no tempo. Portanto, como base esparsa do sinal, a transformada de Fourier é utilizada. A matriz que realiza a mudança do domínio da frequência para o domínio do tempo é a matriz da transformada inversa de Fourier, que pode ser obtida como sendo o conjugado complexo da transformada de Fourier de uma matriz identidade. Dessa forma, a matriz Ψ que representa a transformada inversa de Fourier é dada por:

$$\Psi = \mathcal{F}(\mathbf{I}_{n \times n})^*. \quad (27)$$

2.3.4.5 Matriz \mathbf{A}

A matriz \mathbf{A} que representa o modelo do demodulador aleatório é dada pelo produto entre as matrizes do modelo, conforme indicado em (28) (PANKIEWICZ et al., 2011).

$$\mathbf{A} = \Delta \mathbf{H} \mathbf{P}_c \Psi. \quad (28)$$

A matriz \mathbf{A} contém todas as etapas da aquisição do sinal (demodulação, filtro e amostragem) e a representação esparsa do sinal.

3 MATERIAIS E MÉTODOS

De maneira geral, a simulação dos sistemas estudados neste trabalho consiste na simulação de sistemas lineares, processamento de arquivos, criação, manipulação e operações com vetores, matrizes e números complexos, além da necessidade do uso de algoritmos de otimização e ferramentas para a criação de gráficos. Uma ferramenta capaz de atender todos os requisitos citados é o *software* Matlab, que possui o seu uso disseminado no meio acadêmico. No entanto, optou-se por utilizar a linguagem de programação Python para a implementação das simulações, uma linguagem de programação de propósito geral, lançada em 1991 (ROSSUM; DRAKE, 2003).

A escolha de utilizar Python ao invés de usar uma ferramenta mais específica, como o Matlab, baseia-se em alguns fatores. Python é uma linguagem que possui seu desenvolvimento em código aberto, sendo mantida pela *Python Software Foundation*¹, uma organização sem fins lucrativos. Por outro lado, o *software* Matlab é pago, sendo que sua licença possui um custo elevado. Além disso, o algoritmo de otimização que é utilizado na simulação da técnica de *compressed sensing* pode ser utilizado em uma implementação prática. Dessa forma, um possível protótipo se torna livre do uso de uma ferramenta paga, o que contribui para o conceito de ferramenta de baixo custo. Um outro ponto importante é o fato que o uso de Python tem aumentado de forma significativa nos últimos anos. A revista *IEEE Spectrum*, que anualmente publica a lista das melhores linguagens de programação baseada em diversos fatores, posicionou Python em primeiro lugar em 2017, a frente de outras linguagens como C, C++ e Java (CASS; 2017).

3.1 MATERIAIS

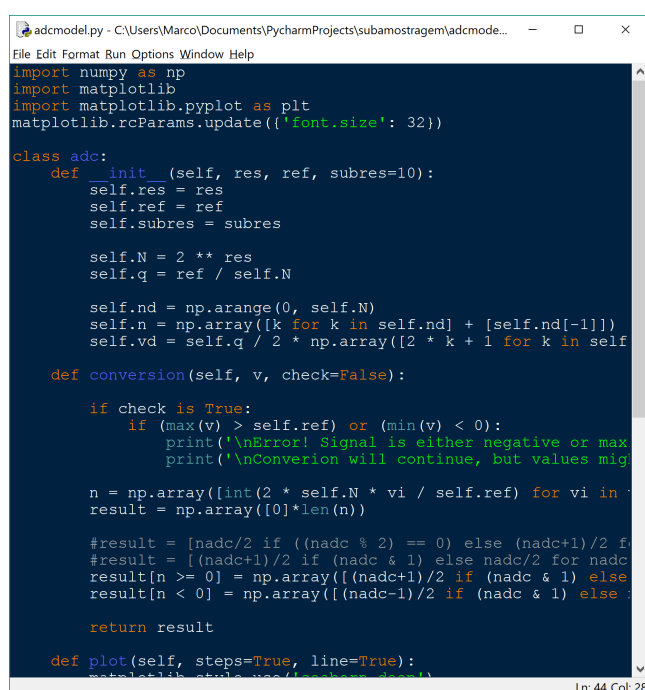
O material necessário para o desenvolvimento do trabalho resume-se a um computador com capacidade de executar códigos escritos em Python e com acesso a internet, para pesquisas bibliográficas.

¹<https://www.python.org/psf/>

3.1.1 Python e ambiente de desenvolvimento

O Python conta com diversas versões, sendo as versões 2.7, 3.5 e 3.6 as versões mais utilizadas. Para o desenvolvimento do trabalho, optou-se por utilizar a versão 3.6.2.

Para a escrita do código, optou-se por utilizar o IDLE, um ambiente de desenvolvimento que é fornecido junto a distribuição dos arquivos de instalação do Python. Embora o IDLE seja uma ferramenta simples, apresenta recursos suficientes para o desenvolvimento do código para a simulação. A Figura 8 mostra a janela de edição de código do IDLE.



```
adcmodel.py - C:\Users\Marco\Documents\PycharmProjects\subamostragem\adcmode... - □ ×
File Edit Format Run Options Window Help
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
matplotlib.rcParams.update({'font.size': 32})

class adc:
    def __init__(self, res, ref, subres=10):
        self.res = res
        self.ref = ref
        self.subres = subres

        self.N = 2 ** res
        self.q = ref / self.N

        self.nd = np.arange(0, self.N)
        self.n = np.array([k for k in self.nd] + [self.nd[-1]])
        self.vd = self.q / 2 * np.array([2 * k + 1 for k in self.nd])

    def conversion(self, v, check=False):
        if check is True:
            if (max(v) > self.ref) or (min(v) < 0):
                print('\nError! Signal is either negative or max')
                print('\nConversion will continue, but values mig

        n = np.array([int(2 * self.N * vi / self.ref) for vi in v])
        result = np.array([0]*len(n))

        #result = [nadc/2 if ((nadc % 2) == 0) else (nadc+1)/2 for nadc in n]
        #result = [(nadc+1)/2 if (nadc & 1) else nadc/2 for nadc in n]
        result[n >= 0] = np.array([(nadc+1)/2 if (nadc & 1) else nadc/2 for nadc in n])
        result[n < 0] = np.array([(nadc-1)/2 if (nadc & 1) else nadc/2 for nadc in n])

        return result

    def plot(self, steps=True, line=True):
        matplotlib.pyplot.plot(self.n, self.vd, 'o', label='Step Response')
        if line:
            matplotlib.pyplot.plot(self.n, self.vd, 'b-', label='Linear Approximation')
```

Figura 8 – IDLE, ambiente de desenvolvimento utilizado.
Fonte: Autoria própria.

3.1.1.1 Pacotes adicionais

Para a implementação das simulações, utilizou-se pacotes adicionais disponíveis para Python. Os seguintes pacotes foram utilizados para o desenvolvimento do trabalho:

- *NumPy*²: pacote padrão para uso científico. Possui funções de álgebra linear, geração de números aleatórios, implementação da transformada de Fourier, convolução, entre outros. Além disso, possibilita a criação de matrizes multidimensionais.

²<http://www.numpy.org>

- *Matplotlib*³: pacote padrão para a criação de gráficos de alta qualidade, em diversos formatos. Possibilita a criação de gráficos 2D, 3D, gráficos com barra de erro, entre outros.
- *Spgl1*⁴: pacote que conta com algoritmos para a solução de problemas de otimização baseados na norma ℓ_1 .

3.2 MÉTODOS

A simulação do sistema foi implementada utilizando uma abordagem modular. A partir do modelo de blocos dos sistemas, cada bloco foi implementado de maneira independente. Os blocos recebem um vetor em sua entrada, parâmetros e fornecem um vetor na saída. Dessa forma, os blocos podem ser facilmente conectados para formar o sistema.

3.2.1 Validação dos modelos

Quando possível, os modelos descritos e a sua respectiva implementação em Python foram comparados com resultados obtidos com funções disponíveis pelo NumPy ou simulações realizadas com o *software* Matlab. Por exemplo, a resposta ao impulso, obtida para o filtro de ordem 4, foi comparada com a resposta ao impulso gerada pelo *software* Matlab. As comparações realizadas podem ser encontradas no Apêndice A.

3.2.2 Sistema de subamostragem

A implementação da simulação do sistema de subamostragem foi baseada na Figura 9. A simulação do sistema consiste em informar a quantidade de amostras desejadas (A), a frequência de discretização do sinal a ser amostrado (f_d), a frequência de amostragem (f_s), a resolução do ADC (N), a referência do ADC (v_{ref}) e um vetor com as frequências em que se deseja obter os coeficientes de Fourier (\mathbf{f}).

Internamente, os parâmetros são utilizados pelos blocos individuais, conforme indicado na Figura 9. Os dados fornecidos aos blocos possuem a seguinte forma. O bloco Subamostragem recebe um vetor em sua entrada, já corrompido por ruído, e fornece

³<https://matplotlib.org>

⁴https://github.com/drrelyea/SPGL1_python_port

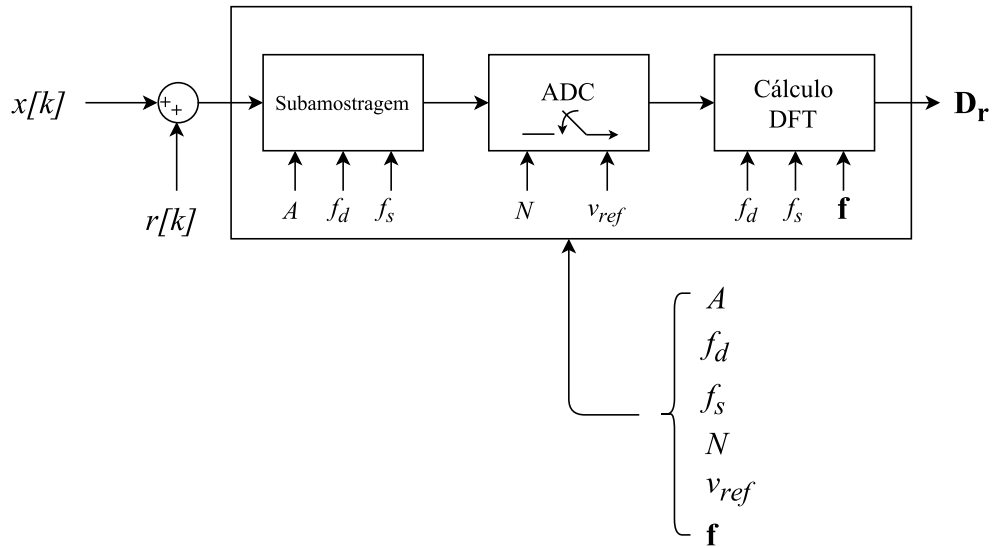


Figura 9 – Bloco de simulação para o sistema de subamostragem.

Fonte: Autoria própria

amostras do vetor em sua saída, conforme os parâmetros informados. Em seguida, a saída do bloco Subamostragem é fornecida ao bloco ADC, que recebe um vetor em sua entrada e realiza a sua quantização. Finalmente, os dados quantizados pelo bloco ADC são fornecidos ao bloco de cálculo da DFT, que realiza o cálculo dos coeficientes de Fourier nas frequências conhecidas.

3.2.3 Demodulador aleatório

O modelo considerado para a simulação do demodulador aleatório pode ser visto na Figura 10. O sistema difere ligeiramente do modelo apresentado na seção 2.3.4 (Figura 7). A diferença encontra-se no modelo do ADC. Na seção 2.3.4, considera-se que o ADC realiza a amostragem e a quantização do sinal. No modelo da simulação, considera-se que a amostragem e a quantização são realizadas por blocos diferentes. Dessa forma, há uma maior flexibilidade na aplicação da amostragem, uma vez que os blocos de amostragem e quantização podem ser modificados de maneira independente.

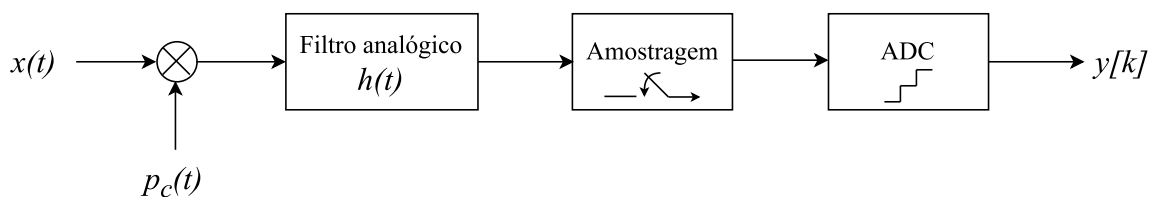


Figura 10 – Esquema para simulação do demodulador aleatório.

Fonte: Adaptado de Laska et al., 2007.

Para implementar a simulação, utilizou-se o esquema indicado na Figura 11. Para simular o sistema, é necessário informar a frequência de discretização utilizada para gerar o sinal de entrada (f_d), a frequência de reconstrução desejada (f_r), a frequência de amostragem (f_{adc}), a frequência de corte do filtro (f_c), a quantidade de pontos a serem simuladas (n), a resolução do ADC (N) e a tensão de referência utilizada pelo ADC (v_{ref}). Internamente, a simulação utiliza os parâmetros fornecidos nos blocos individuais de cada etapa.

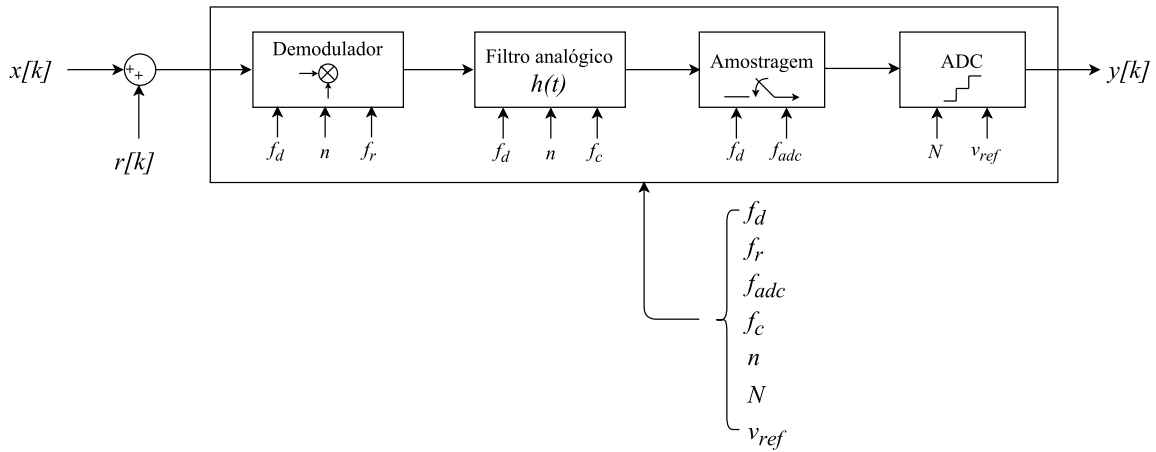


Figura 11 – Bloco de simulação para o demodulador aleatório.

Fonte: Autoria própria

Após a simulação do sistema, as amostras $y[k]$ e o modelo do sistema (matrizes Φ e Ψ) são fornecidas a um algoritmo, que resolve o problema de minimização para reconstrução do sinal (conforme descrito na seção 2.3.2).

O problema, como formulado, é conhecido como *basis pursuit denoising* (VIDYA-SAGAR, 2016). O pacote disponível para Python, *Spgl1*, utilizado no desenvolvimento da simulação, possui algoritmos para a execução do *basis pursuit denoising*. Por fim, como saída, o algoritmo fornece um vetor esparso $\tilde{\mathbf{D}}$, conforme ilustrado na Figura 12.

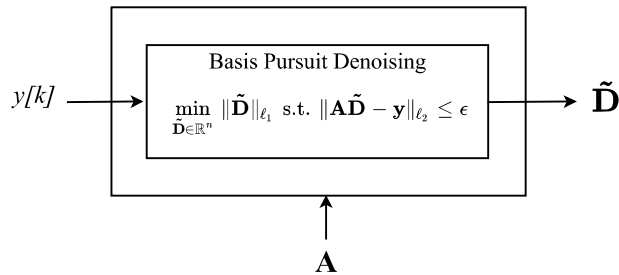


Figura 12 – Bloco de reconstrução para o demodulador aleatório.

Fonte: Autoria própria

3.2.4 Análise de desempenho

Para realizar a análise de desempenho dos sistemas, utilizou-se o erro de amplitude e o erro de fase. O erro é obtido a partir do sinal de entrada e do sinal obtido com cada um dos sistemas.

Para determinar o erro de amplitude, utilizou-se a seguinte expressão:

$$\text{erro}_{\text{mag}}(f) = \frac{A(f) - \tilde{A}(f)}{A(f)} \cdot 100\%, \quad (29)$$

em que f é a frequência do sinal, A é a amplitude da componente na frequência f e \tilde{A} é a amplitude obtida, na frequência f .

Para obter o erro de fase, a seguinte expressão foi utilizada:

$$\text{erro}_{\text{fase}}(f) = \angle A(f) - \angle \tilde{A}(f). \quad (30)$$

4 RESULTADOS E DISCUSSÃO

Esse capítulo apresenta os principais resultados obtidos com as simulações do sistema de amostragem e do demodulador aleatório, bem como as comparações e análise de *hardware* de ambos.

4.1 SIMULAÇÕES

As simulações dos sistemas foram feitas com parâmetros similares para possibilitar a posterior comparação de desempenho e erro de cada sistema. Os parâmetros da simulação foram determinados a partir de uma aplicação: aquisição de bioimpedância. Nesse contexto, definiu-se que o sistema deve ser capaz de realizar a aquisição de sinais de até 5 MHz.

A seguir, a simulação de cada sistema e os parâmetros utilizados são apresentados junto a uma discussão das particularidades encontradas.

4.1.1 Sistema de subamostragem

Para a simulação do sistema de subamostragem, os parâmetros indicados na Tabela 1 foram utilizados.

Tabela 1 – Parâmetros para a simulação do sistema de subamostragem.

Parâmetro	Valor	Parâmetro	Valor
f_d	1 GHz	Amostras	100
f_s	2,5 MHz	Resolução ADC	12 bits
SNR	40 dB	Referência ADC	3,3 V
Pontos de simulação	40000		

A Figura 13 mostra o resultado obtido. A curva em azul mostra o sinal de entrada, que é composto por duas componentes de frequência: uma componente com magnitude 1 em 1 MHz e fase de 12° ; e uma componente com magnitude 0,5 em 2,5 MHz e fase de 47° . Também em azul, estão indicadas as amostras que são tomadas do sinal. A curva em laranja mostra as amostras quantizadas, que são utilizadas no processamento do algoritmo da subamostragem.

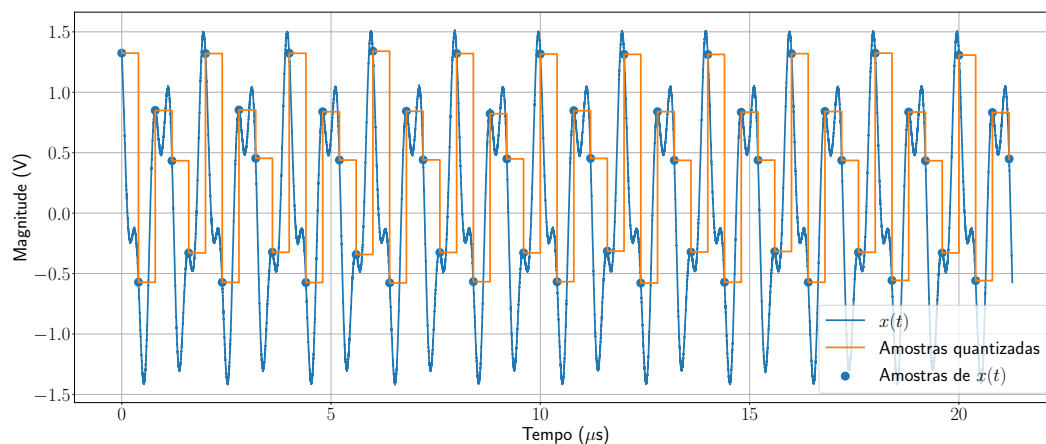


Figura 13 – Resultados da simulação do sistema de subamostragem.

Fonte: Autoria própria

Para o processamento das amostras, foi fornecido ao algoritmo as frequências de 1 MHz e 2,5 MHz. Os resultados obtidos podem ser vistos na Tabela 2.

Tabela 2 – Resultados obtidos para a simulação do sistema de subamostragem.

Frequência	Amplitude	Fase
1 MHz	0,9992 V	12,06°
2,5 MHz	0,6823 V	0,00°

A partir da Tabela 2, observa-se que o erro de magnitude e fase na frequência de 2,5 MHz é elevado. Em particular, o algoritmo encontrou fase zero para o sinal em 2,5 MHz. Isso se deve a uma particularidade do sistema de subamostragem, em que as frequências contidas no sinal não pode ser múltiplas da frequência de amostragem.

Para ilustrar o problema, considere o caso em que o sinal contém três frequências: uma componente em 2,5 MHz (o mesmo da frequência de amostragem), uma componente em 5 MHz (segunda harmônica da frequência de amostragem) e uma componente em 7,5 MHz (terceira harmônica da frequência de amostragem); todas as componentes possuem amplitude 1. A simulação do sistema pode ser visto na Figura 14. Nesse caso, as amostras coincidem com o período do sinal e o processamento falha, pois o sinal aparece como um sinal contínuo. Isso ocorre devido ao próprio efeito de *aliasing* (BENTSMAN, 2016). No entanto, o sinal é distorcido de tal forma que não é possível obter sua informação de frequência, pois a amostragem verifica um sinal contínuo. Mesmo que a componente de frequência do sinal não seja harmônica da frequência do sinal de amostragem, o algoritmo demonstra problemas. No caso em que a frequência do sinal é 1,5 vezes a frequência de amostragem, a amostragem apresenta problemas.

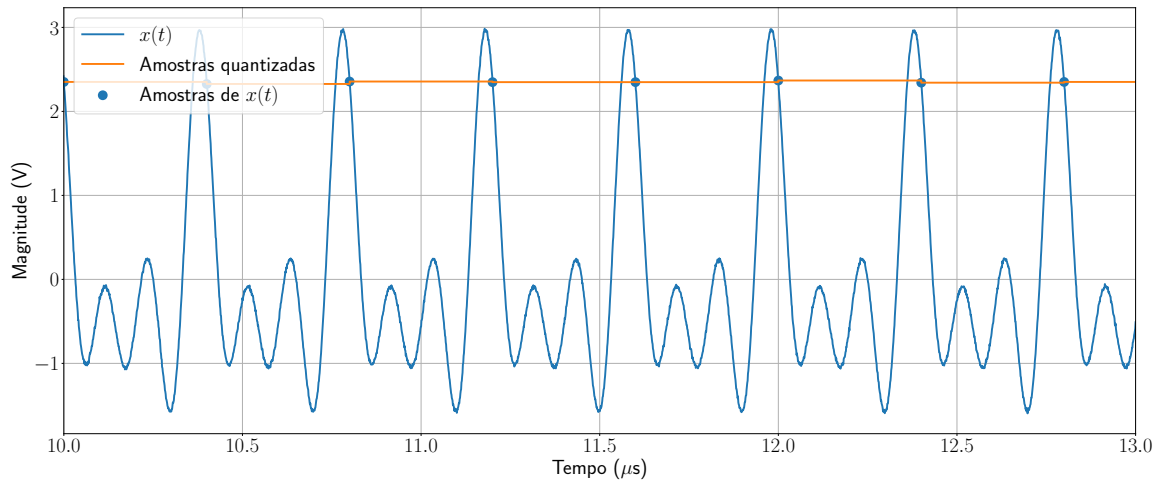


Figura 14 – Resultados da simulação do sistema de subamostragem, para sinais com frequências múltiplas da frequência de amostragem.

Fonte: Autoria própria

Para contornar esse problema, é necessário garantir que a frequência do sinal sendo subamostrado não seja um múltiplo da frequência de amostragem. Para isso, escolhe-se uma frequência de amostragem de forma a minimizar a possibilidade que o sinal sendo amostrado seja múltiplo.

A Tabela 3 mostra os resultados obtidos da simulação do sistema de subamostragem com uma frequência de 2,35 MHz, considerando que o sistema tem em sua entrada um sinal de 1 MHz com amplitude de 1 e fase de 13° e um sinal de 2,5 MHz com amplitude de 1 e fase de 25° . Os parâmetros para a simulação são os mesmos que apresentados na Tabela 1, exceto pela frequência de amostragem (de 2,5 MHz para 2,35 MHz) e a frequência de discretização (de 1 GHz para 0,94 GHz). Nota-se que a amplitude e fase obtidas nas frequências de 1 MHz e 2,5 MHz apresentam um erro menor do que se comparado aos resultados da Tabela 2.

Tabela 3 – Resultados da simulação com uma frequência de amostragem de 2,35 MHz.

Frequência	Amplitude	Fase
1 MHz	0,9990 V	$13,62^\circ$
2,5 MHz	0,9863 V	$24,32^\circ$

4.1.1.1 Erro de amplitude e fase

Durante as simulações do sistema de subamostragem, notou-se que os erros de amplitude e fase obtidos com o sistema de subamostragem variam com a quantidade de amostras. Em geral, um número maior de amostras resulta em um erro menor. No entanto, o erro apresenta oscilações conforme o número de amostras varia. A Figura 15 mostra o erro de amplitude obtido variando o número de amostras de 100 a 1000, em que a curva em azul representa o erro na componente de 1 MHz, enquanto que a curva em laranja representa o erro na componente de 2,5 MHz. Para a componente de 1 MHz, é possível notar que o erro é próximo de zero quando a quantidade de amostras é igual a 100. No entanto, ao aumentar o número de amostras para 110, o erro chega a 2%. É possível notar que o erro mostra um comportamento oscilatório conforme o número de amostras é alterado.

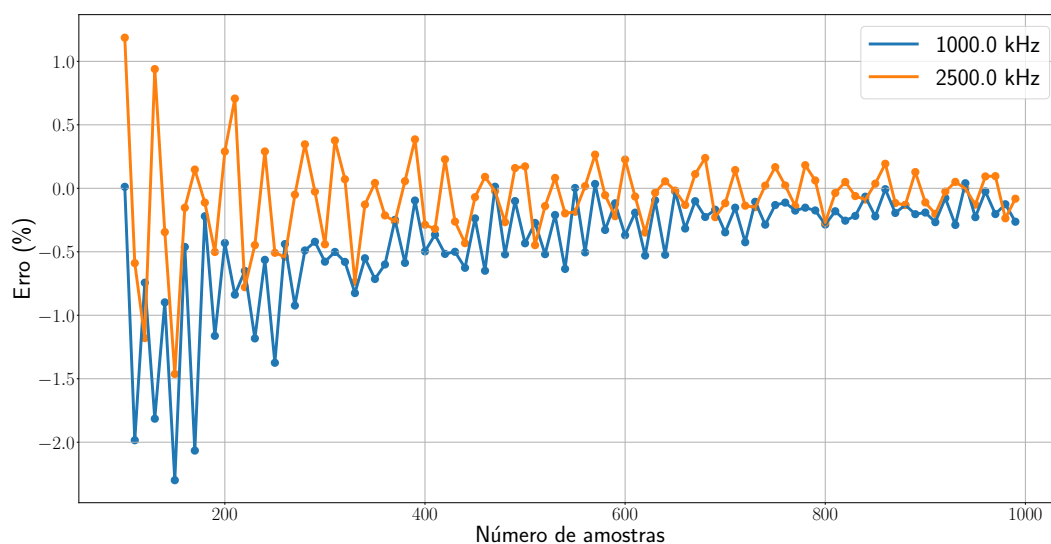


Figura 15 – Erro de amplitude em função da quantidade de amostras.

Fonte: Autoria própria

O mesmo comportamento é observado para o erro de fase, conforme indicado na Figura 16. Após alterar o número de amostras de 100 para 110, o erro de fase ambos os sistemas diminui. No entanto, ao aumentar de 110 para 120, o erro de fase aumenta. Para o erro de fase, nota-se um comportamento oscilatório similar ao de erro de amplitude.

Os comportamentos observados nas Figuras 15 e 16 podem estar associados ao fenômeno de espalhamento, relacionado com o cálculo dos coeficientes de Fourier (LYON, 2009). Se o erro for de fato proveniente do espalhamento, é possível, a princípio, determinar um número de amostras ótimo, de forma a minimizar o erro.

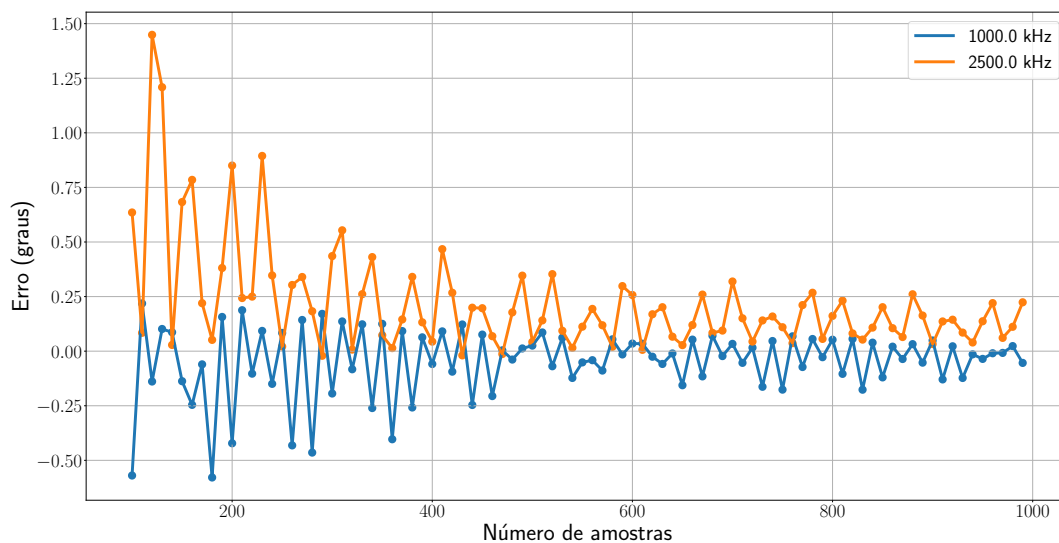


Figura 16 – Erro de fase em função da quantidade de amostras.
Fonte: Autoria própria

4.1.2 Demodulador aleatório

As simulações do demodulador aleatório foram realizadas baseadas nos parâmetros indicados na Tabela 4. Os parâmetros de frequência de amostragem, SNR do sinal de entrada, quantidade de amostras, resolução do ADC e referência do ADC são os mesmos que os utilizados na simulação do sistema de subamostragem.

Tabela 4 – Parâmetros para a simulação do demodulador aleatório.

Parâmetro	Valor	Parâmetro	Valor
f_d	1 GHz	Pontos de simulação	40000
f_r	50 MHz	Amostras	100
f_c	1,25 MHz	Pontos reconstruídos	2000
f_{adc}	2,5 MHz	Resolução ADC	12 bits
SNR	40 dB	Referência ADC	3,3 V

A Figura 17 mostra os resultados obtidos com a simulação do demodulador aleatório. A primeira curva mostra o sinal de entrada, $x(t)$. O sinal possui duas componentes de frequência: uma componente com magnitude 1 em 1 MHz e uma componente com magnitude 0,5 em 2,5 MHz. A segunda curva mostra o sinal $x(t)$ multiplicado pela sequência aleatória de +1 e -1, $p_c(t)$, que possui uma frequência de 50 MHz. A terceira curva mostra o resultado da multiplicação entre $x(t)$ e $p_c(t)$ após passar pelo filtro passa-baixas, que possui uma frequência de corte de 1,25 MHz. Além disso, a terceira curva também mostra as amostras que são tomadas pelo ADC, a uma taxa de 2,5 MHz.

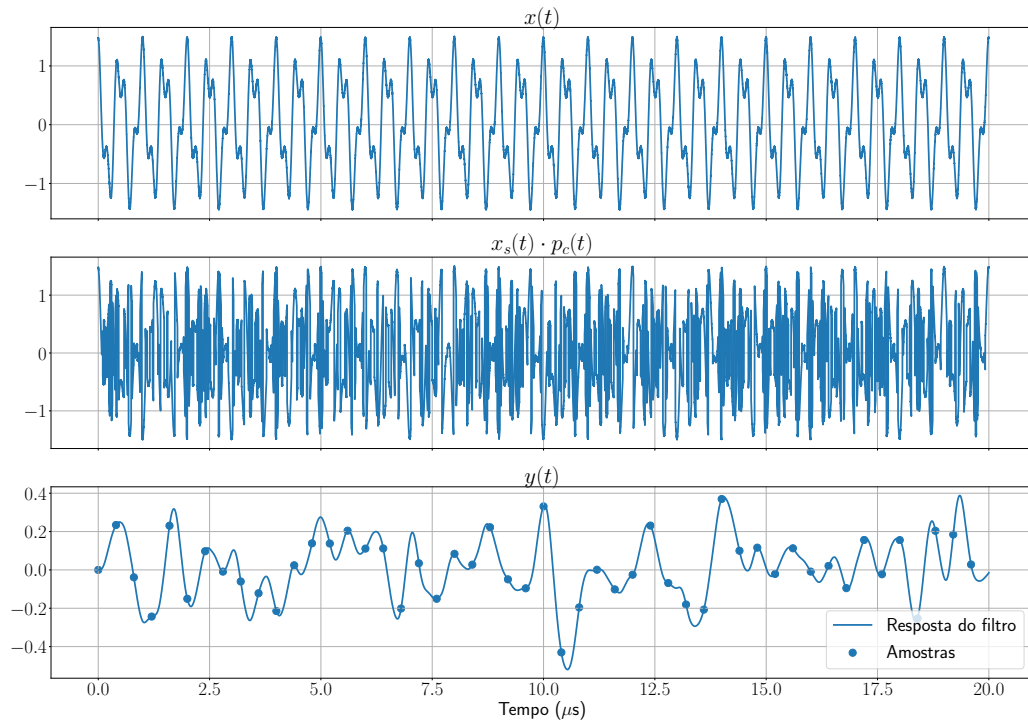


Figura 17 – Resultados da simulação do demodulador aleatório.

Fonte: Autoria própria

A partir da simulação, as amostras que são obtidas são fornecidas ao algoritmo para recuperar o sinal. O algoritmo recupera o sinal diretamente no domínio da frequência, uma vez que é nesse domínio que o sinal possui uma representação esparsa. Além disso, o algoritmo recupera as amostras na mesma frequência da variação da sequência aleatória, no caso, 50 MHz. A Figura 18 mostra o sinal recuperado no domínio da frequência e no domínio do tempo.

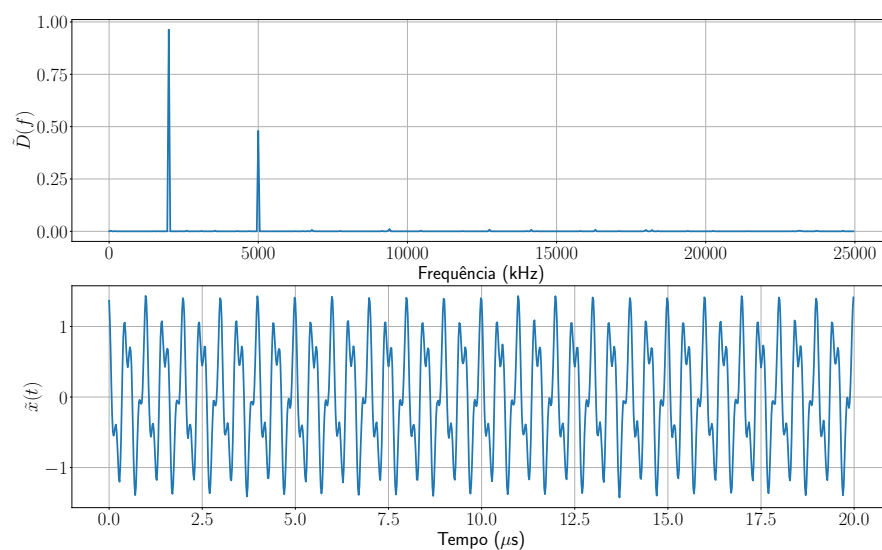


Figura 18 – Coeficientes de Fourier obtidos (superior) e sinal reconstruído no tempo (inferior).

Fonte: Autoria própria

A Figura 19 mostra uma comparação entre três sinais. A curva em azul mostra o sinal de entrada, que é discretizado em 1 GHz. Em laranja pontilhado, está indicado um sinal amostrado diretamente a 50 MHz. Em verde, a reconstrução obtida em 50 MHz a partir de amostras em 2,5 MHz está indicada. É possível notar que o sinal reconstruído possui uma diferença de fase em relação ao sinal de entrada, além de uma amplitude ligeiramente menor.

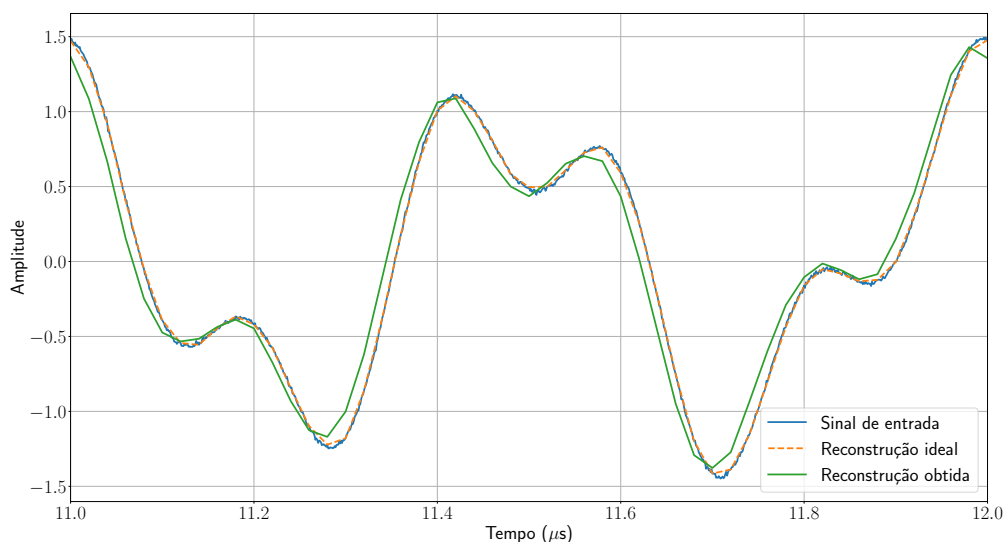


Figura 19 – Comparação entre sinal de entrada (azul), reconstrução ideal (laranja) e reconstrução obtida (verde).

Fonte: Autoria própria

4.1.2.1 Diferença de fase

Conforme indicado na Figura 19, verificou-se que o sinal reconstruído apresenta uma fase em relação ao sinal de entrada. A partir das simulações com sinais em outras frequências, também foi constatado que a diferença de fase observada varia conforme a frequência do sinal de entrada varia.

Para investigar esse fenômeno, o sistema foi simulado para sinais com frequências diferentes, variando de 50 kHz a 5 MHz. Como o modelo conta com uma sequência aleatória, os resultados obtidos variam de simulação para simulação. Para evitar uma conclusão errônea acerca do comportamento do demodulador aleatório e obter uma melhor estimativa sobre seu comportamento, as simulações foram repetidas 40 vezes para cada sinal na entrada, possibilitando obter um valor médio e um desvio padrão para os resultados em cada frequência. A diferença de fase entre o sinal reconstruído e o sinal de entrada pode ser visto na Figura 20.

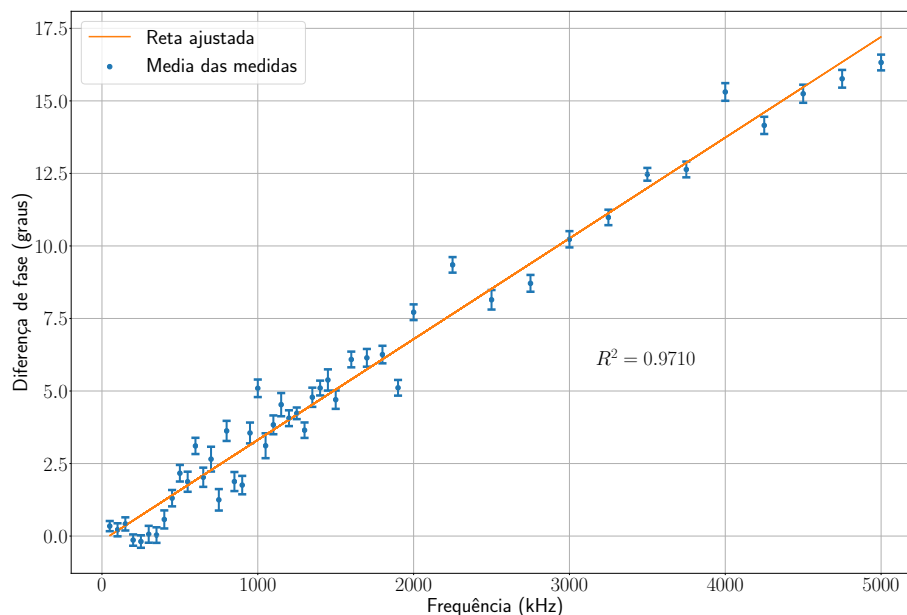


Figura 20 – Diferença de fase entre entrada e reconstrução, em função da frequência do sinal de entrada.

Fonte: Autoria própria

Nota-se que para frequências baixas (se comparadas a frequência de reconstrução do sinal, de 50 MHz), a diferença de fase é relativamente baixa. No entanto, não há uma relação linear entre a frequência do sinal e a diferença de fase. Por outro lado, para frequências maiores, a diferença de fase aparenta mostrar uma relação linear.

Conforme indicado na Figura 20, uma reta que melhor se ajusta aos pontos foi obtida, utilizando o método do *least-squares* (MILLER; 2006). A partir da reta, seria possível corrigir a fase do sinal após a sua reconstrução. No entanto, o erro obtido com esse método pode mostrar um erro elevado para pontos que estão muito distantes da reta (como é o caso dos pontos em 1 MHz, 2 MHz e 4 MHz, por exemplo).

Para avaliar o erro de fase produzido pelo demodulador aleatório, a fase obtida na reconstrução foi corrigida, baseada na Figura 20. Após simular o sistema, a fase obtida foi corrigida com o valor médio apresentado nas simulações indicadas na Figura 20. Isso é possível pois a diferença de fase apresentada pelo algoritmo de reconstrução aparenta ser sistemático, uma vez que a diferença obtida em cada frequência não é aleatória, e o desvio padrão é relativamente pequeno.

As simulações realizadas foram feitas considerando uma frequência de reconstrução de 50 MHz. A diferença de fase obtida é função da frequência do sinal de entrada. No entanto, questiona-se se a frequência de reconstrução pode também ser um fator determinante. Para verificar se essa diferença de fase apresentada na reconstrução do sinal varia

conforme a frequência de reconstrução, realizou-se simulações para quatro frequências de reconstrução, conforme indicado na Figura 21.

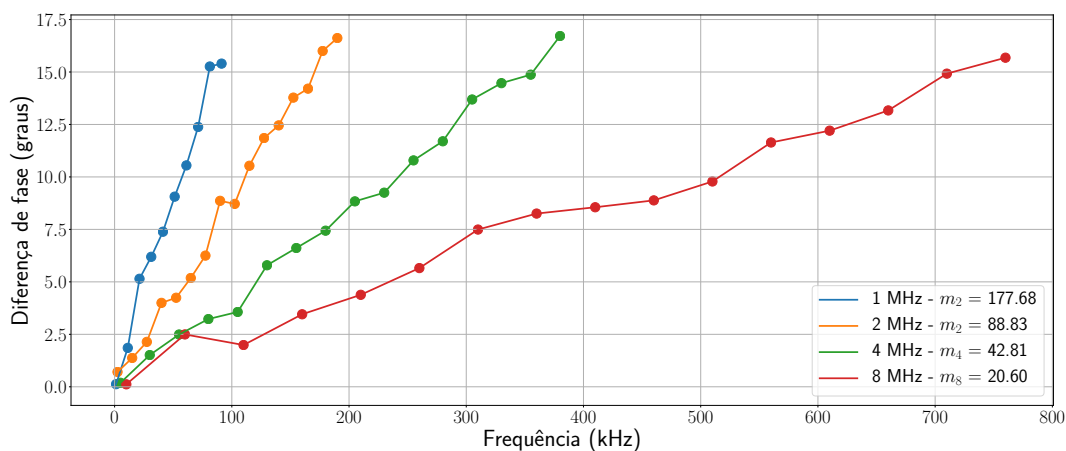


Figura 21 – Diferença de fase em função da frequência do sinal de entrada, para frequências de reconstrução diferentes.

Fonte: Autoria própria

Na Figura 21, nota-se que a diferença de fase obtida também varia com a frequência de reconstrução. Por exemplo, quando o sinal possui uma frequência de 100 kHz, a diferença de fase obtida é de aproximadamente 2,5 graus para uma frequência de reconstrução de 8 MHz e aproximadamente 9 graus para uma frequência de reconstrução de 2 MHz.

O fato da diferença de fase depender da frequência do sinal de entrada e da frequência de reconstrução é preocupante, uma vez que, se necessário alterar a frequência de reconstrução, o ensaio da diferença de fase deverá ser repetido. No entanto, é possível que exista uma relação entre a razão entrada/reconstrução e a diferença de fase. Na Figura 21, as simulações foram realizadas considerando que a frequência do sinal de entrada varia até aproximadamente um décimo da frequência de reconstrução. Próximo desses valores, a diferença de fase obtida é de cerca de 16 graus. Além disso, a legenda da Figura 21 indica os coeficientes angulares das retas que melhor se ajustam a cada uma das curvas. Nota-se que, conforme a frequência de reconstrução dobra, o coeficiente angular (ou inclinação da reta) cai pela metade, reforçando a hipótese.

Nesse caso, seria possível realizar um ensaio com uma frequência normalizada. Nesse ensaio, o resultado obtido seria a diferença de fase em função da razão entre a frequência de entrada e a frequência de reconstrução. A Figura 22 mostra a diferença de fase em função da razão entre frequência de entrada e a frequência de reconstrução. A Figura 22 possui exatamente os mesmos dados da Figura 21, com a única diferença que a

frequência do sinal foi dividida pela frequência de reconstrução.

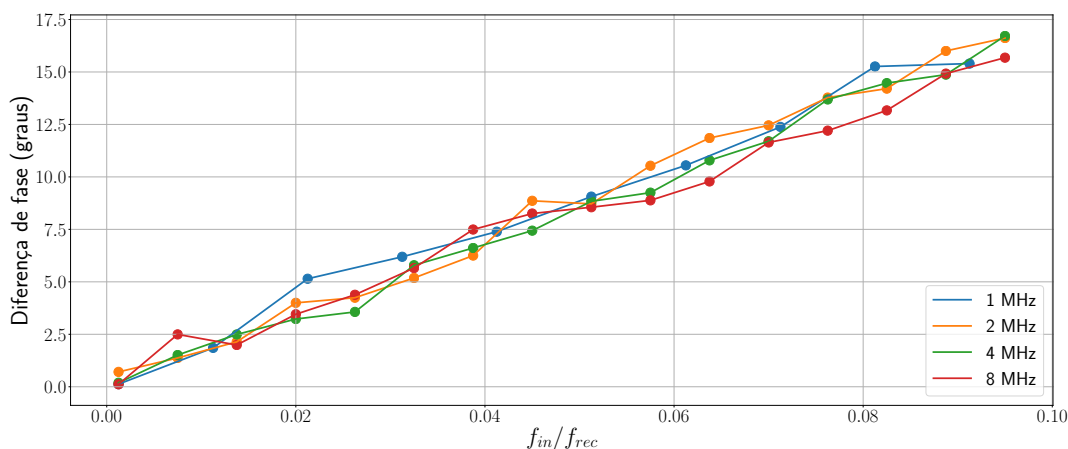


Figura 22 – Diferença de fase em função da razão entre frequência do sinal de entrada (f_{in}) e a frequência de reconstrução (f_r).

Fonte: Autoria própria

Conforme indicado na Figura 22, as curvas para as diferentes frequências de reconstrução estão próximas, reforçando uma possível relação entre diferença de fase e razão entre frequência do sinal de entrada e a frequência de reconstrução. Essa possível relação merece maiores investigações que, no momento, fogem ao escopo deste trabalho.

4.1.2.2 Qualidade da reconstrução

Para determinar a qualidade da reconstrução, utilizou-se a métrica do SNR, em que calcula-se a relação entre o sinal reconstruído e o erro obtido, em relação ao sinal de entrada. No caso, utilizou-se a relação entre os coeficientes de Fourier obtidos com o algoritmo ($\tilde{\mathbf{D}}$) e os coeficientes de Fourier do sinal de entrada (\mathbf{X}), sendo o SNR do erro dado por:

$$\text{SNR}_{\text{erro}} = 10 \log_{10} \left(\frac{\|\tilde{\mathbf{D}}\|^2}{\|\tilde{\mathbf{D}} - \mathbf{X}\|^2} \right). \quad (31)$$

A Figura 23 mostra o SNR obtido para as simulações de erro de magnitude e diferença de fase. Nota-se que o SNR se mantém em uma média acima de 30 dB para frequências relativamente baixas se comparada a frequência de reconstrução. No entanto, conforme a frequência do sinal de entrada aumenta, a qualidade da reconstrução começa a degradar.

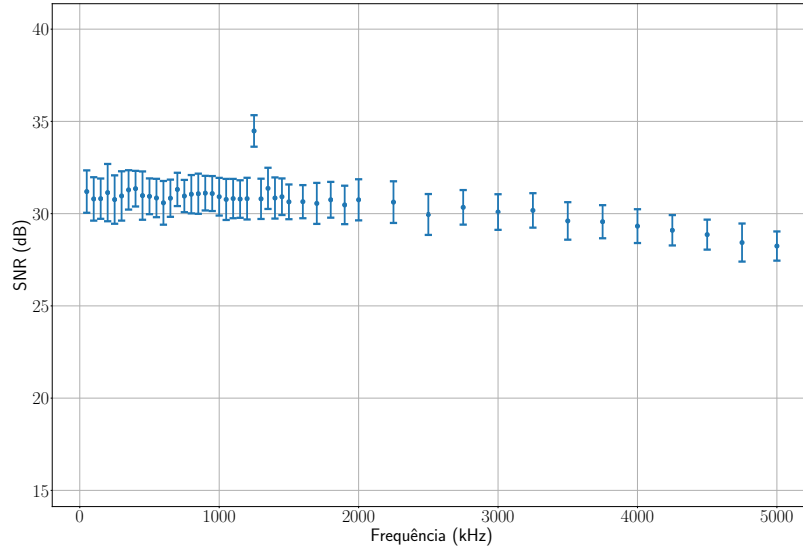


Figura 23 – SNR obtido, em função da frequência do sinal de entrada.

Fonte: Autoria própria

4.1.2.3 Parâmetro de regularização

O problema formulado para a reconstrução do sinal consiste em otimizar a norma ℓ_1 do vetor esparso (ver seção 2.3.2):

$$\min_{\tilde{\mathbf{D}} \in \mathbb{R}^n} \|\tilde{\mathbf{D}}\|_{\ell_1} \quad \text{s.t.} \quad \|\mathbf{A}\tilde{\mathbf{D}} - \mathbf{y}\|_{\ell_2} \leq \epsilon. \quad (32)$$

O parâmetro ϵ possui um papel importante na reconstrução do sinal pois, conforme seu valor aumenta, permite-se recuperar um sinal com um erro maior. Por outro lado, conforme seu valor diminui, o algoritmo busca um sinal que produza um erro menor. No entanto, o parâmetro ϵ também influencia na esparsidade da solução procurada: conforme seu valor é maior, permite-se um erro maior com a vantagem de encontrar uma solução com uma esparsidade menor. Da mesma forma, conforme o seu valor diminui, permite-se um erro menor mas o sinal procurado possui uma esparsidade maior.

Em suma, o parâmetro ϵ regula uma troca entre esparsidade e fidelidade. No contexto dos sinais analisados, a esparsidade se traduz em componentes de frequência obtidas. É possível recuperar um sinal com mais componentes de frequência e que produza um erro menor ou recuperar sinal com menos componentes de frequência mas com um erro maior.

A Figura 24 mostra a simulação do sistema, com os mesmos parâmetros indicados na Tabela 4, sendo que o sinal de entrada possui uma componente em 1 MHz com

amplitude 1 e uma componente em 2,5 MHz, com amplitude 0,5. No caso em que o valor de ϵ é de 0,001, o sinal recuperado contém diversas componentes de frequência, embora as componentes possuem magnitude reduzida. Por outro lado, quando o valor de ϵ é de 0,2, o algoritmo recuperou um sinal com apenas duas componentes de frequência. No entanto, o erro de magnitude é maior.

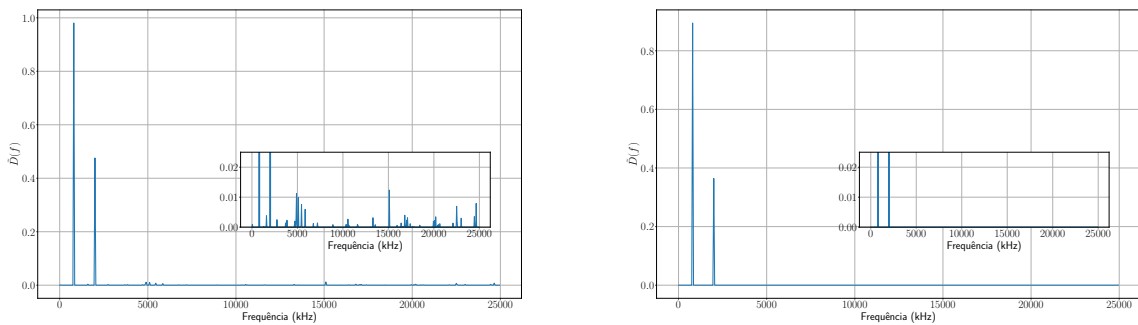
(a) $\epsilon = 0,001$ (b) $\epsilon = 0,2$

Figura 24 – Simulação do demodulador aleatório para (a) $\epsilon = 0,001$ e (b) $\epsilon = 0,2$
Fonte: Autoria própria

Essa característica do parâmetro ϵ motivou a investigação da variação da qualidade da reconstrução (SNR) em função do valor do parâmetro. A Figura 25 mostra os resultados obtidos. É possível notar que, quando o valor de ϵ é elevado, a qualidade da reconstrução cai. Conforme o parâmetro diminui, a qualidade da reconstrução aumenta até um certo ponto. Diminuindo o parâmetro a partir desse ponto resulta em uma degradação do SNR, uma vez que o algoritmo insere mais componentes de frequência no sinal para tentar ajustar o melhor ajuste possível da estimativa as amostras observadas.

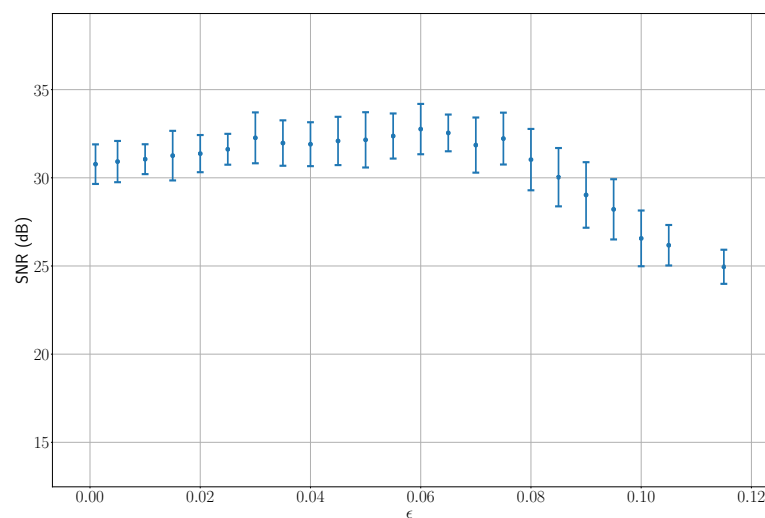


Figura 25 – SNR obtido em função do erro permitido ϵ .
Fonte: Autoria própria

Durante as simulações do demodulador aleatório, optou-se por utilizar ϵ de 0,01. Essa escolha foi baseada no SNR do sinal de entrada, que era de 40 dB. Para esse valor de SNR, o valor RMS do erro é de 0,01; dessa forma, o algoritmo permitiu um erro até o nível de ruído do sinal de entrada.

4.2 COMPARAÇÕES

Para as comparações, os parâmetros utilizados para as simulações são os mesmos que os indicados nas Tabelas 1 e 4. O único parâmetro alterado, posteriormente, é a resolução do ADC, que é alterada para comparar a resposta dos sistemas para diferentes valores de resolução.

4.2.1 Erro em função do sinal de entrada

Os erros de magnitude e fase para cada sistema foram investigados para diversos sinais de entrada com frequências diferentes. A Figura 26 mostra o erro de magnitude obtido para ambos os sistemas.

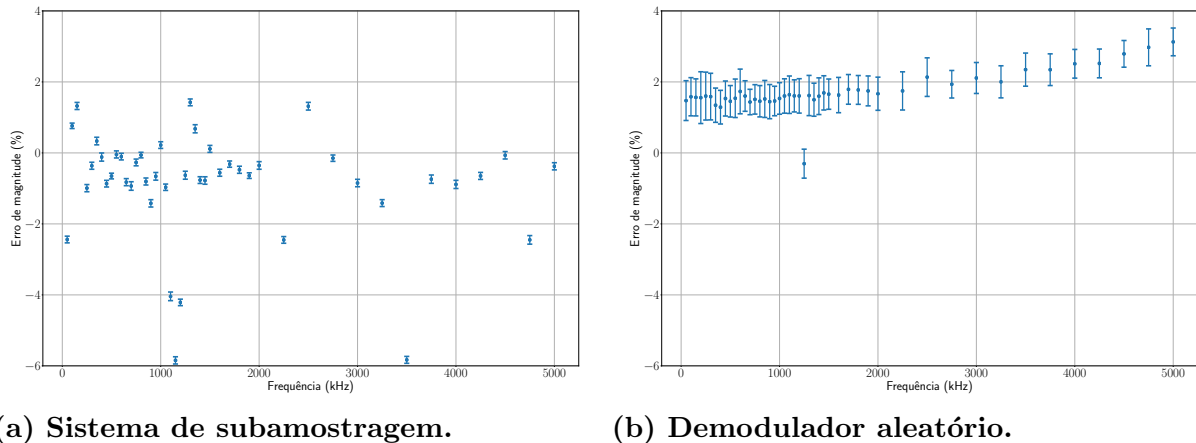


Figura 26 – Comparação do erro de amplitude entre (a) o sistema de subamostragem e (b) o demodulador aleatório.

Fonte: Autoria própria

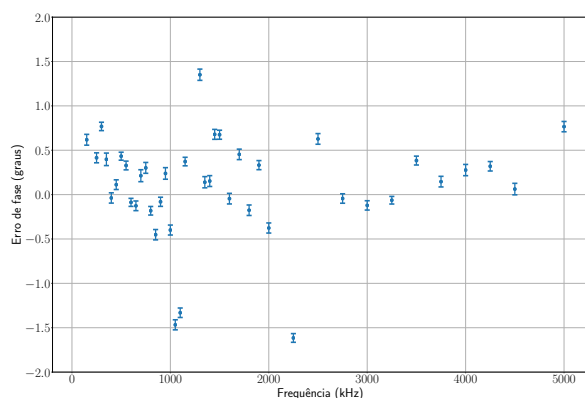
É possível notar algumas diferenças entre o comportamento dos sistemas. A primeira diferença é o fato de que o erro de magnitude aparenta ser independente da frequência do sinal de entrada para o sistema de subamostragem, enquanto é possível notar que, no demodulador aleatório, o erro tende a aumentar quando se tem frequências mais elevadas. Outra diferença notável é que a dispersão do erro é muito menor para o sistema de subamostragem, o que lhe atribui uma maior robustez.

No sistema de subamostragem, nota-se dois pontos no gráfico que possuem um erro de aproximadamente -6% , quando o sinal de entrada possui uma frequência de $1,15\text{ MHz}$ e $3,5\text{ MHz}$. Além disso, os pontos em torno de -4% possuem frequência de $1,1\text{ MHz}$ e $1,2\text{ MHz}$. O erro pode ser devido ao efeito de espalhamento, em que os sinais com essas frequências estão fora da resolução do espectro da transformada discreta de Fourier. Exceto os pontos nessas quatro frequências, os demais pontos apresentam erros em uma faixa de $-2,5\%$ e $1,5\%$.

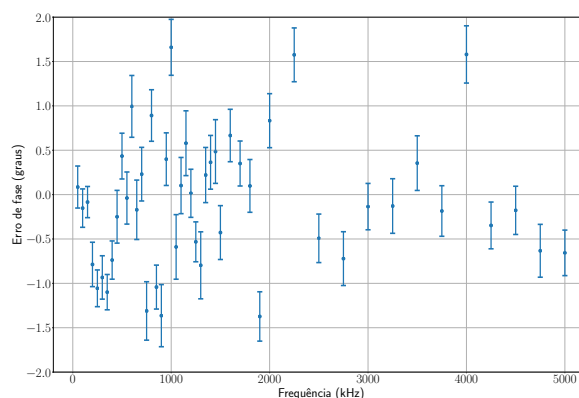
No sistema do demodulador aleatório, nota-se que a dispersão do erro é elevada. Por exemplo, para sinais com frequência até 2 MHz , o erro se mantém em uma média de $1,6\%$, com um desvio padrão de cerca de $0,7\%$. Considerando que 99% das medidas realizadas estão na faixa de até 3 vezes o desvio padrão, o sistema do demodulador aleatório pode apresentar um erro de até $3,7\%$. Para frequências maiores do que 2 MHz , a média do erro tende a aumentar, sendo cerca de $3,2\%$ para uma frequência de 5 MHz . Considerando um desvio padrão de $0,4\%$, o erro obtido pode ser até de $4,5\%$.

Em geral, o erro obtido é menor para o sistema de subamostragem, uma vez que grande parte dos pontos estão dentro de uma faixa de -1% e $+1\%$, além de apresentar uma dispersão menor. Uma dispersão menor confere uma maior repetibilidade ao sistema.

As simulações também produziram o erro de fase de cada sistema. Os erros podem ser vistos na Figura 27, que compara o erro em função da frequência do sinal de entrada.



(a) Sistema de subamostragem.



(b) Demodulador aleatório.

Figura 27 – Comparação do erro de fase entre (a) o sistema de subamostragem e (b) o demodulador aleatório.

Fonte: Autoria própria

Análogo ao erro de magnitude, o demodulador aleatório apresenta uma dispersão maior do que a do sistema de subamostragem. No entanto, diferentemente do

erro de magnitude, ambos os sistemas apresentam, em média, erros de fase comparáveis. Enquanto o sistema de subamostragem apresenta um erro de fase que varia entre cerca de -1,5% a 3%, o demodulador aleatório apresenta um erro que varia de cerca de -2% a 2%. Porém, como a dispersão do demodulador aleatório é maior, o erro também pode ser maior.

Outro fato interessante apontado pelo erro de fase da Figura 27 é que o erro do demodulador aleatório aparenta ser independente da frequência do sinal de entrada, ao contrário do erro de magnitude, que aumenta conforme o sinal de entrada aumenta após cerca de 2 MHz. Isso garante ao sistema uma maior robustez para a detecção de fase.

No quesito de erro de amplitude e fase em função da frequência do sinal de entrada, o sistema de subamostragem leva vantagem. Para o sistema de subamostragem, o erro de amplitude concentra-se em torno de 0%, variando de -2,5% a 1,5%, enquanto que para o demodulador aleatório o erro concentra-se em torno de 1,5%, e varia com a frequência. Além disso, a dispersão do erro é menor para o sistema de subamostragem, o que lhe confere uma maior repetibilidade. Para o erro de fase, o desempenho de ambos os sistemas são comparáveis. No entanto, a dispersão do erro do sistema de subamostragem é menor.

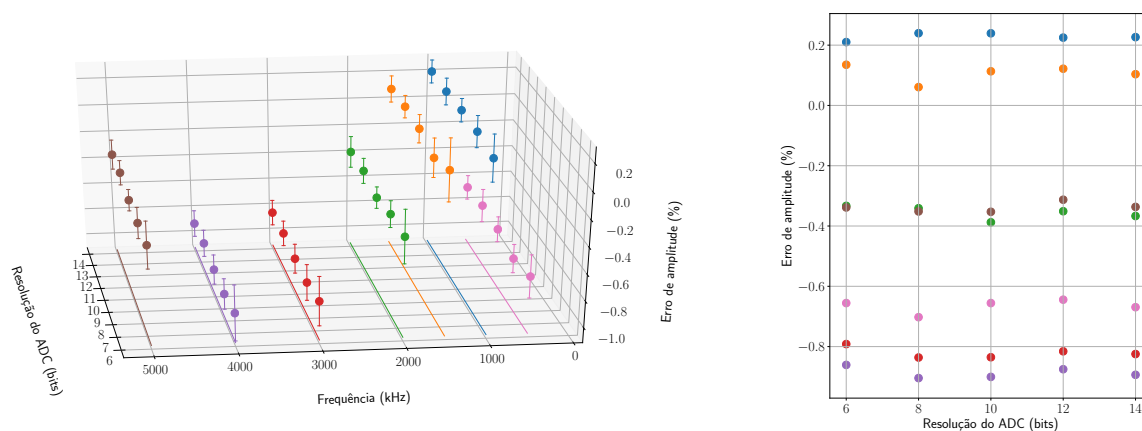
Os erros obtidos com ambos os sistemas podem ser melhorados. Nesse trabalho, compara-se o desempenho dos sistemas com parâmetros similares. No entanto, o sistema de subamostragem pode apresentar um desempenho melhor se o número de amostras for ajustado corretamente. Nesse caso, os pontos que apresentam um erro de -6% poderiam ser contornados. Para o demodulador aleatório, o mesmo é verdadeiro. Os erros poderiam ser melhorados ajustando o parâmetro de regularização ou até mesmo escolhendo um outro algoritmo para a reconstrução do sinal. O sistema de subamostragem leva vantagem no quesito de erro dentro dos parâmetros e limitações avaliadas nesse trabalho.

4.2.2 Erro em função da resolução do ADC

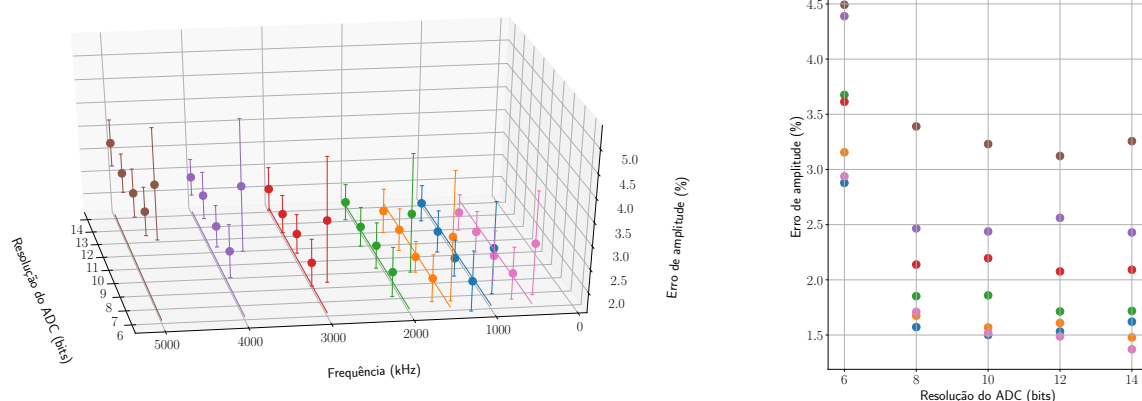
A resolução do ADC é um parâmetro importante a ser avaliado, uma vez que ADCs com maiores resoluções são mais caros e, em geral, possuem taxas de amostragem menores se comparados a ADC com resoluções menores.

A Figura 28 mostra o erro de magnitude em função da frequência do sinal de entrada e da resolução do ADC. No gráfico a esquerda, o erro de magnitude é mostrado simultaneamente em função da frequência e do ADC, enquanto que no gráfico a direita,

mostra-se o erro em função da resolução do ADC, em que cada conjunto de pontos da mesma cor representa uma frequência diferente para o sinal de entrada.



(a) Sistema de subamostragem.



(b) Demodulador aleatório.

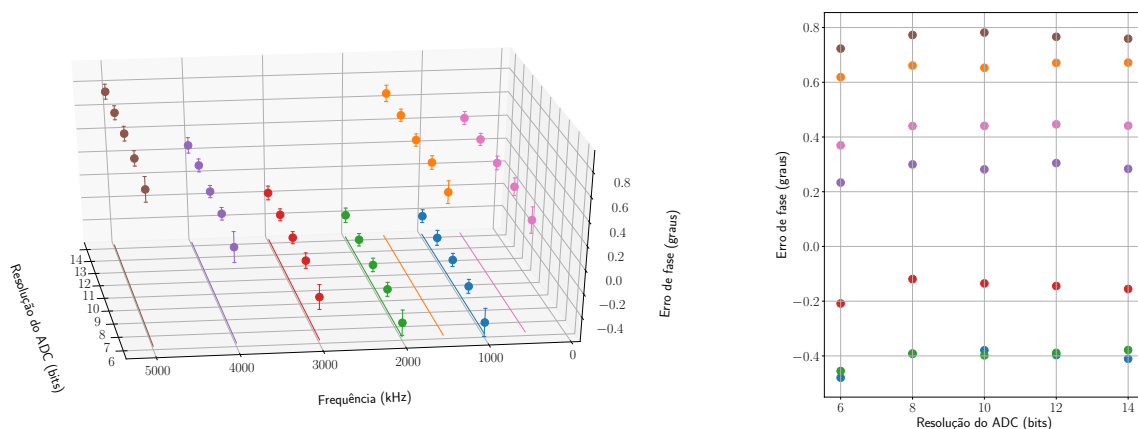
Figura 28 – Erro de magnitude em função da frequência do sinal de entrada e ADC, para (a) o sistema de subamostragem e (b) o demodulador aleatório.

Fonte: Autoria própria

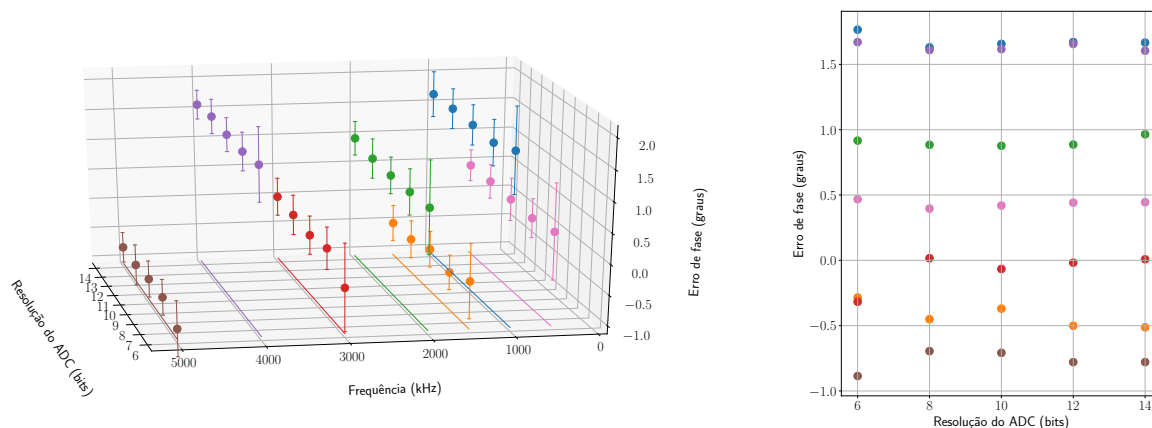
Para o sistema de subamostragem, nota-se que, em média, o erro de magnitude é o mesmo para as diferentes resoluções do ADC. No entanto, a dispersão é maior quando a resolução do ADC é menor. Além disso, para uma resolução maior do que 10 bits, a dispersão do erro não muda de maneira significativa.

Por outro lado, o demodulador aleatório possui um comportamento diferente. Para um ADC de 6 bits, o erro é maior para todas as frequências de entrada. A partir da resolução de 8 bits, o erro não muda de maneira significativa. O desvio padrão do erro também é elevado para uma resolução de ADC menor, enquanto que a partir dos 10 bits, seu valor não se altera de forma significativa.

No mesmo formato da Figura 28, a Figura 29 mostra o erro de fase para cada sistema, em função da resolução do ADC e da frequência do sinal de entrada.



(a) Sistema de subamostragem.



(b) Demodulador aleatório.

Figura 29 – Erro de fase em função da frequência do sinal de entrada e ADC, para (a) o sistema de subamostragem e (b) o demodulador aleatório.

Fonte: Autoria própria

Nota-se que para ambos os sistemas, o erro de fase varia pouco conforme a resolução do ADC muda, o que mostra a robustez dos sistemas para a detecção de fase. Entretanto, a dispersão do erro é maior para resolução menores.

No quesito de erro em função da resolução do ADC, pode-se dizer que o sistema de subamostragem leva vantagem, pois o mesmo poderia ser utilizado com um ADC em uma resolução mais baixa, apresentando resultados melhores do que o demodulador aleatório.

4.2.3 Tempo de processamento

Para o sistema de subamostragem, calcular o coeficiente de Fourier para apenas uma componente de frequência consiste apenas em executar as equações 11–14. Para 100 amostras, o cálculo das componentes consiste em realizar 200 multiplicações e 200 operações de soma. Após o cálculo do coeficiente, o módulo e a fase podem ser obtidas com algumas operações a mais. Por outro lado, para o demodulador aleatório, o processamento consiste em resolver um problema de otimização complexo, que é um processo iterativo. Enquanto o sistema de subamostragem opera apenas com números reais, o demodulador aleatório opera com números complexos.

Nas simulações realizadas, as operações para o processamento do sistema de subamostragem são executadas de forma instantânea, não sendo possível medir o tempo de execução. Para o demodulador aleatório, o tempo de processamento das amostras levou, em média, cerca de 4 segundos. Além disso, o tempo de processamento para o demodulador aleatório varia de forma não-linear com a quantidade de amostras. Para 25, 50, 100 e 200 amostras, o tempo de processamento é cerca de 0,3, 0,8, 4,3 e 24 segundos; respectivamente.

Em uma implementação, o processamento do sistema de subamostragem pode ser realizado em um microcontrolador, enquanto que o processamento das amostras do demodulador aleatório precisariam ser realizado em um computador com maior capacidade de processamento. Com isso, no quesito processamento, o sistema de subamostragem leva vantagem, uma vez que o tempo de processamento pode ser implementado com números reais e o tempo é determinístico.

4.3 VIABILIDADE DE HARDWARE

A implementação do sistema tem como ponto de partida a premissa do uso do ADC de um microcontrolador. No caso, o uso do microcontrolador STM32F407VGT6, junto a sua placa de desenvolvimento, indicada na Figura 30. Entre as principais características, o microcontrolador STM32F407VGT6 conta com um processador ARM Cortex M4 (32 bits) com *clock* até 168 MHz, 3 ADCs de 2,4 MS/s, 1 MB de memória *flash* e 192 kB de memória RAM (STMICROELECTRONICS, 2017). Um parâmetro importante do ADC é o tempo de aquisição, que é o tempo que o capacitor interno do conversor leva para ser carregado com a tensão de entrada. Conforme indicado em seu manual, o tempo de aquisição do ADC do microcontrolador STM32F407VGT6 é de, no mínimo, 100 ns. Um sinal de 5 MHz corresponde a um período de 200 ns, ou seja, até o capacitor se carre-

gar, a onda do sinal de entrada já percorreu metade de sua variação. Dessa forma, para garantir a correta amostragem, limita-se a aquisição de sinais de até 1 MHz, para ambos os sistemas.

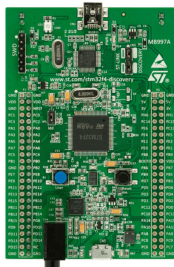


Figura 30 – *Kit* de desenvolvimento com o microcontrolador STM32F407VGT6.

Fonte: STMicroelectronics, 2017

Outra consideração importante é o *clock* do microcontrolador. Para o sistema do demodulador aleatório, será necessário gerar uma onda quadrada, a partir da sequência aleatória, com frequência elevada. Dessa forma, o microcontrolador deve possuir processamento suficiente para alternar a sua saída digital na taxa definida pela frequência de reconstrução. Para a implementação em frequências mais baixas, opta-se por definir uma frequência de reconstrução de 10 MHz, para garantir que haverá processamento o suficiente para que o microcontrolador consiga gerar a sequência aleatória e realizar o tratamento das amostras adquiridas.

4.3.1 *Hardware* para o sistema de subamostragem

O sistema de subamostragem possui como único parâmetro a frequência de amostragem. Para a implementação em hardware, opta-se por uma frequência de amostragem de 135 kHz.

Como o próprio ADC do microcontrolador escolhido possui um tempo de aquisição reduzido, o único componente necessário é um *buffer* na entrada do ADC. O componente selecionado é o BUF602, que possui uma largura de banda de 880 MHz e um *slew rate* de 8000 V/ μ s, a um custo unitário de U\$2,89¹. Embora a largura de banda do *buffer* selecionado é muito maior do que o necessário, essa folga possibilita uma futura expansão da frequência de amostragem.

¹O preço unitário dos componentes foram encontrados *online*, no site *Digikey*.

4.3.2 Hardware para o demodulador aleatório

Para o demodulador aleatório, propõe-se os seguintes parâmetros:

- Frequência de reconstrução: 10 MHz
- Frequência de amostragem: 500 kHz
- Frequência de corte do filtro: 250 kHz

O hardware proposto para o demodulador aleatório pode ser visto na Figura 31. O hardware conta com uma etapa de condicionamento para a sequência aleatória, multiplicação entre o sinal de entrada e a sequência de -1 e +1, filtro passa-baixas e amostragem pelo ADC.

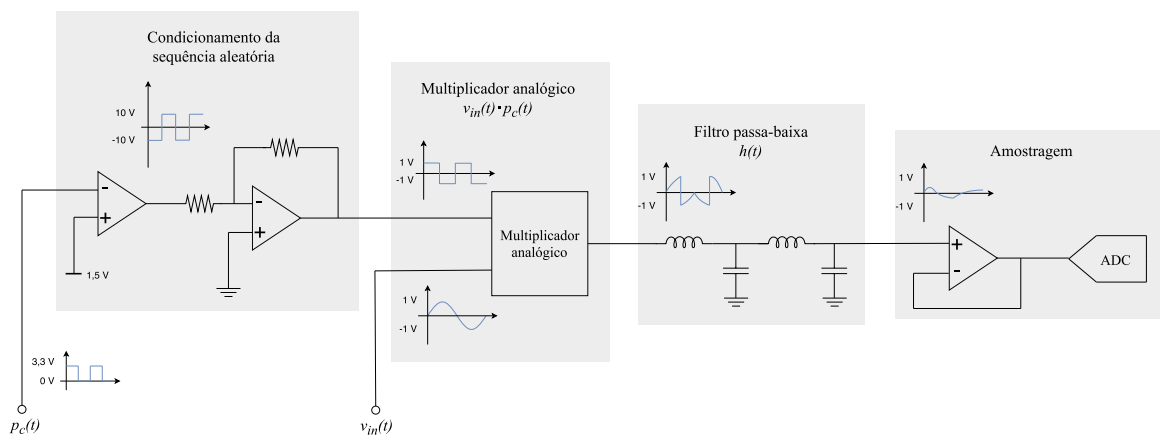


Figura 31 – Hardware proposto para o demodulador aleatório.

Fonte: Autoria própria

A sequência aleatória, $p_c(t)$, é gerada pelo microcontrolador como níveis lógicos variando entre 0 e 3,3 V. Esse sinal passa por um circuito com um comparador e um amplificador, conforme ilustrado na Figura 31. Esse circuito tem como propósito converter a sequência de 0 e 3,3 V para uma sequência de -1 V e +1 V. Teoricamente, a onda quadrada possui um número infinito de componentes de frequência, sendo a primeira componente a frequência da onda e as demais componentes as harmônicas ímpares (LATHI, 2000). Dessa forma, os amplificadores operacionais utilizados no processo de condicionamento da sequência aleatória precisam possuir uma largura de banda elevada. O circuito integrado (CI) OPA3695 apresenta-se como um forte candidato para a implementação, possuindo uma largura de banda de 600 MHz e um *slew rate* de 2600 V/ μ s, além de possuir três amplificadores operacionais por CI, com um custo unitário de U\$6,96.

O multiplicador realiza a etapa de multiplicação entre a sequência e o sinal de entrada. O componente do multiplicador precisa possuir largura de banda e *slew rate* elevados, para acompanhar a variação brusca de polaridade do sinal de entrada. Além disso, a largura de banda precisa ser elevada, uma vez que o processo de multiplicação espalha o conteúdo dos sinais no espectro da frequência. O componente AD835 apresenta uma largura de banda de 250 MHz com *slew rate* de 1000 V/ μ s. O CI possui apenas um multiplicador e possui um custo de U\$21,12.

Na Figura 31, o filtro passa-baixas é implementado com componentes passivos. O uso de componentes ativos também é uma possibilidade. Opta-se por componentes passivos devido a facilidade de implementação. Como a resposta do filtro é fundamental no modelo do demodulador aleatório, os componentes devem possuir uma tolerância reduzida.

A última etapa consiste na amostragem do sinal. O ADC do microcontrolador possui uma impedância relativamente baixa na entrada analógica, o que pode aumentar o tempo de aquisição do sinal. Para contornar esse problema, opta-se por utilizar um *buffer* na entrada do ADC. O *buffer* tem como função apresentar uma alta impedância para a aquisição do sinal do na saída do filtro, sem interferir com a impedância do sistema, além de apresentar uma baixa impedância para o ADC, acelerando o processo de carga do capacitor interno. O componente selecionado é o mesmo que para o sistema de subamostragem, o BUF602.

A Tabela 5 mostra os componentes propostos para o *hardware* do demodulador aleatório.

Tabela 5 – Componentes propostos para o demodulador aleatório.

Componente	Custo unitário
OPA3695	U\$6,96
AD835	U\$21,12
BUF602	U\$2,89

4.3.3 Comparação

Os componentes propostos para o desenvolvimento de *hardware* dos sistemas não estão disponíveis no Brasil. Para a implementação dos sistemas, será necessário a importação de componentes dos Estados Unidos ou da China. Portanto, no quesito disponibilidade de materiais, a implementação de ambos os sistemas apresentam a mesma dificuldade.

A quantidade de componentes previstas para o sistema de subamostragem é menor do que a quantidade proposta para a implementação do demodulador aleatório. O componente que mais agrega custo a implementação do demodulador aleatório é o multiplicador analógico, sendo responsável por cerca de 2/3 do preço dos três componentes principais do sistema. No quesito preço, o sistema de subamostragem apresenta um custo de implementação menor do que o demodulador aleatório e, portanto, leva vantagem no quesito de preço.

O *hardware* do sistema de subamostragem consiste no uso de apenas um componente externo, o *buffer* para o ADC. Com isso, a placa para o uso do componente requer apenas alguns componentes adicionais, como capacitores de desacoplamento, recomendados no manual do componente. Por outro lado, o demodulador aleatório conta com diversos componentes: amplificadores para o condicionamento da sequência aleatória, multiplicador analógico, filtro passa-baixas e *buffer* para o ADC. Além disso, o *design* da placa para o uso do sistema requer cuidados especiais, uma vez que o circuito opera com frequências superiores a 10 MHz. Portanto, no quesito complexidade da placa de circuito impresso, o sistema de subamostragem leva vantagem.

O demodulador aleatório apresenta outra dificuldade em sua implementação. A reconstrução das amostras simuladas, realizada em um computador pessoal, levaram um tempo considerável. Em um microcontrolador, executar o algoritmo pode ser inviável. Dessa forma, o *hardware* do demodulador aleatório, junto ao microcontrolador, realizam apenas o condicionamento e aquisição das amostras. O processamento, em si, deverá ser realizado externamente. Para o sistema de subamostragem, o processamento de amostras pode ser realizado localmente, o que lhe confere uma maior conveniência de uso.

Embora o sistema de subamostragem leve vantagem no quesito de custo e complexidade da placa de circuito impresso, o sistema não se mostra expansível. A frequência máxima do sinal de entrada é definida como sendo 1 MHz, que é limitada pelo ADC do microcontrolador utilizado. Por outro lado, o hardware proposto pelo demodulador aleatório pode ser expandido, uma vez que a frequência de corte do filtro foi ajustada em 250 kHz. A frequência de corte poderia ser aumentada, caso o microcontrolador se mostre capaz de gerar uma onda quadrada em frequências maiores. Por exemplo, se o microcontrolador conseguir gerar uma onda em 20 MHz, a frequência de corte do filtro pode ser ajustada em 500 kHz, e a frequência do sinal de entrada pode ser expandida para 2 MHz. Para aumentar a frequência do sinal de entrada do sistema de subamostragem, um ADC com um tempo menor de aquisição deverá ser utilizado. Portanto, visando uma expansão do sistema com o *hardware* adquirido, o demodulador aleatório leva vantagem.

5 CONCLUSÕES

O trabalho apresentou a formulação e modelagem de duas técnicas de aquisição e processamento de sinais, que buscam reduzir a taxa de amostragem necessária sem comprometer a qualidade do sinal digital.

No âmbito de modelagem do processo de aquisição de sinais, o processo de inserção de ruído no sinal de entrada e o processo de quantização do conversor analógico-digital foram investigados, implementados e validados.

Para o sistema de subamostragem, o processo de amostragem e processamento foram formulados e modelados. A partir da simulação do sistema de subamostragem, algumas de suas características, como o problema com a frequência de amostragem e o erro em função de amostras, foram apresentados.

No caso da técnica de *compressed sensing*, as suas principais características, junto a formulação do problema de otimização, foram discutidas. Como a técnica é formulada apenas para sinais discretos, sua implementação para sinais analógicos, incluindo o demodulador aleatório, também foi discutida e modelada. O modelo do demodulador aleatório contou com a formulação das matrizes de amostragem, multiplicação aleatória, filtro e transformação de base. As simulações realizadas com o demodulador aleatório exploraram os problemas relacionados com a diferença de fase obtida entre os sinais, com o algoritmo de reconstrução; a qualidade da reconstrução e a investigação do efeito do parâmetro de regularização nos resultados.

As simulações dos sistemas possibilitaram realizar algumas comparações de desempenho, baseadas em erros de amplitude e fase obtidos. Neste trabalho, foi realizada a análise do erro dos sistemas em função da frequência do sinal de entrada e da resolução do ADC. Em ambos os quesitos, o sistema de subamostragem levou vantagem, uma vez que os erros obtidos foram, em média, menores. Para frequências de até 1 MHz, o demodulador aleatório apresentou um erro, em média, de 1,5%, com um desvio padrão de cerca de 0,5%. Com isso, 99% das medidas estão dentro da faixa de até 4% de erro. Por outro lado, a maior parte dos pontos obtidos com o sistema de subamostragem estão entre +1% e -1% de erro, com desvio padrão de até 0,1%. O erro de fase obtido para os dois sistemas se mostraram em escalas comparáveis, com o sistema de subamostragem também apresentando menor desvio padrão nas medidas. Esse fato confere uma repetibilidade e

robustez maior para o sistema de subamostragem.

As simulações realizadas alterando a resolução do ADC mostraram comportamentos diferentes para cada sistema. No sistema de subamostragem, o erro não variou de maneira significativa. No entanto, o desvio padrão dos erros de fase e a amplitude variaram, sendo cerca de 0,18% para a resolução de 6 bits e cerca de 0,10% para as demais resoluções simuladas. Para o demodulador aleatório, a média e o desvio padrão do erro foram maior para a resolução de 6 bits. Em particular, o desvio padrão foi de cerca de 1,2% para a resolução de 6 bits, cerca de 0,55% para a resolução de 8 bits e cerca de 0,5% para as demais resoluções. Dessa forma, o sistema de subamostragem se mostra mais flexível a escolha da resolução do ADC.

A análise de viabilidade de *hardware* foi realizada baseada no uso do ADC do microcontrolador STM32F407VGT6. Definiu-se, como requisito, a aquisição de um sinal de até 1 MHz. Para essa restrição, o sistema de subamostragem mostrou ser mais vantajoso, pois necessita de menos componentes, além dos componentes apresentaram menor custo. Além da necessidade de menos componentes, uma possível placa de circuito impresso seria menos complexa, pois o sistema de subamostragem opera com frequência de até 1 MHz, enquanto o demodulador aleatório iria conter frequências de, pelo menos, 10 MHz. No entanto, a expansão do alcance de frequência do sistema de subamostragem se mostra difícil, uma vez que a aquisição de sinais com frequências mais altas requer a troca do ADC. Para o demodulador aleatório, a expansão do alcance de frequência poderia ser feito sem a troca do ADC, sendo limitado pela capacidade do microcontrolador de gerar a onda quadrada correspondente a sequência aleatória.

5.1 TRABALHOS FUTUROS

Durante a elaboração do trabalho, foram identificados pontos que fogem do escopo desse trabalho. No entanto, alguns dos pontos merecem maiores investigações.

Em relação ao ADC, a simulação considera apenas o processo de quantização. No entanto, cabe simular o tempo de aquisição (ou tempo de amostragem) do ADC. Uma simulação simplificada pode considerar o comportamento do circuito RC que representa a entrada do ADC (KESTER, 2008). Dessa forma, uma melhor avaliação dos parâmetros e limites de frequência de amostragem pode ser realizada.

Para o sistema de subamostragem, um estudo mais detalhado acerca do erro de magnitude e fase em função do número de amostras pode ser feito. Com esse estudo, seria possível escolher um número de amostras ótimo de forma a minimizar o erro. Conforme

visto, um número maior de amostras não necessariamente significa um erro menor, e a escolha de um número de amostras menor pode reduzir o tempo de processamento do sinal.

Para a simulação do demodulador aleatório, uma série de fatores merecem maiores investigações. Os principais pontos identificados são:

- Uso de outros algoritmos para a reconstrução do sinal: o tempo de processamento e os erros de amplitude e fase podem ser investigados com o uso de outros algoritmos. Por exemplo, o algoritmo do *Orthogonal Matching Pursuit* (OMP) poderia ser utilizado. O uso de outro algoritmo poderia diminuir o tempo de processamento e/ou melhorar o desempenho do sistema.
- *Basis pursuit denoising*: o algoritmo utilizado para resolver o problema de *basis pursuit denoising* pode ser comparado com uma implementação em outro *software*. Por exemplo, o Matlab possui um algoritmo para a resolução do problema.
- Parâmetro ϵ : com o algoritmo para o problema de *basis pursuit denoising*, o parâmetro de regularização poderia ser ajustado para reduzir os erros e melhorar a qualidade de reconstrução. O parâmetro poderia ser investigado para uma certa frequência de reconstrução ou o parâmetro poderia ser investigado para uma frequência “normalizada”, buscando otimizá-lo para diversas situações.
- Diferença de fase observada na reconstrução do sinal: a causa da diferença de fase poderia ser investigada de forma mais detalhada, buscando identificar a fonte do problema para tentar contorná-lo. Na situação atual, o erro é corrigido a partir de uma tabela.
- Simulação do filtro analógico: o filtro considerado na simulação é considerado como um filtro ideal. Para uma avaliação mais precisa do sistema, um filtro com características reais, como resistência da fonte de entrada e tolerância dos componentes, poderia ser incluído nas simulações.
- Implementação do filtro analógico: a implementação do filtro analógico considerou um filtro a partir de componentes passivos. No entanto, é possível implementar o filtro com componentes ativos. Cabe um estudo melhor sobre as vantagens de desvantagens de cada sistema, com o objetivo de identificar qual a melhor topologia para a implementação do demodulador aleatório.
- O *hardware* considerado para o demodulador aleatório possui amplificadores operacionais que operam com fontes simétricas, ou *dual supply*. A princípio, seria possível

realizar o processamento com uma fonte simples, ou *single supply*, o que torna o sistema mais simples. Dessa forma, a investigação de uma implementação de *hardware* poderia passar por um estudo mais aprofundado.

REFERÊNCIAS

- AMARAL, Carlos E. F. do. **Desenvolvimento de um medidor multifrequencial de bioimpedância complexa**. 2004. 127p. Dissertação de Mestrado - Centro Federal de Educação Tecnológica do Paraná. Curitiba, 2004.
- AYLLÓN, David; SEOANE Fernando; GIL-PITA, Roberto. Cole Equation and Parameter Estimation from Electrical Bioimpedance Spectroscopy Measurements - A Comparative Study. **31st Annual International Conference of the IEEE EMBS**, 2009.
- BENTSMAN, Joseph. **Introduction to Signal Processing, Instrumentation, and Control: An Integrative Approach**. 1^a ed. WSPC, 2016.
- BERTEMES-FILHO, P; NEGRI, Lucas H.; PATERNO, Aleksander S. Detection of Bovine Milk Adulterants Using Bioimpedance Measurement and Artificial Neural Network. **5th European Conference of the International Federation for Medical and Biological Engineering**, pp. 1275-1278, 2012.
- BERTOTTI, Fabio L. **Sistema de Biotelemetria passiva e técnica de medição para a aquisição da impedância elétrica de tecido animal POST MORTEM**. 2010. 176p. Tese de Doutorado - Universidade Tecnológica Federal do Paraná (Programa de PósGraduação em.). Curitiba, 2010.
- CANDÈS, Emmanuel J.; ROMBERG, Justin K. e TAO, Terence. Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information. **IEEE Transactions on Information Theory**, vol. 52, no. 2, pp. 489-509, Fev. 2006.
- CANDÈS, Emmanuel J. e WAKING, Michael B. An Introduction to Compressive Sampling. **IEEE Signal Processing Magazine**, vol. 25, no. 2, pp. 21-30, Março 2008.
- CASS, Stephen. The 2017 Top Programming Languages. **IEEE Spectrum**. Disponível em <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>. Acesso em 29, set. 2017.
- DUARTE, Marco. F. e ELDAR, Yonina C. Structured Compressed Sensing: From theory to applications. **IEEE Transactions on Signal Processing**, vol. 59, no. 9, pp. 4053-4085, Set. 2011.
- DUDYKEVYCH, Taras; GERSING, E.; Thiel, F. e HELLIGE, Gerhard. Impedance analyser module for EIT and spectroscopy using undersampling. **Physiological Measurement**, vol. 22, pp. 19-24, 2001.
- KESTER, Walt. **ADC Architectures II: Successive Approximation ADCs**. Analog Devices, 2008.

- LASKA, Jason N.; KIROLOS, Sam; DUARTE, Marco F.; RAGHEB, Tamer S.; BARANIUK, Richard G. e MASSOUD, Yehia. Theory and Implementation of an Analog-to-Information Converter using Random Demodulation. **2007 IEEE International Symposium on Circuits and Systems**, New Orleans, LA, 2007, pp. 1959-1962.
- LATHI, Bhagawandas P. **Signal processing and linear systems**. 1^a ed. Oxford University Press, 2000.
- LE, Bin; RONDEAU, Thomas W.; REED, Jeffrey H. e BOSTIAN, Charles W. Analog-to-digital converters. **IEEE Signal Process. Mag.** Vol. 22, no. 6, pp. 69–77, Nov. 2005.
- LING, Carolina H.Y.; de CRAEN, Anton J.M.; SLAGBOOM, Pieternella E.; GUNN, Dave A.; STOKKEL, Marcel P.M.; WESTENDORP, Rudi G.J.; MAIER, Andrea B. Accuracy of direct segmental multi-frequency bioimpedance analysis in the assessment of total body and segmental body composition in middle-aged adult population. **Clinical Nutrition**, vol. 30 , edição 5 , pp. 610 - 615, 2011.
- LUDWIG, Reinhold e BOGDANOV, Gene. **RF Circuit Design: Theory & Applications**. 2^a ed, Pearson, 2008.
- LYON, Douglas. The Discrete Fourier Transform, Part 4: Spectral Leakage. **Journal of object technology**, 2009.
- MARAVIC, Irena e VETTERLI, Martin. Sampling and reconstruction of signals with finite rate of innovation in the presence of noise. **IEEE Transactions on Signal Processing**, vol. 53, no. 8, pp. 2788-2805, Aug. 2005.
- MANCINI, Ron. **Op Amps For Everyone**. Texas Instruments, 2002.
- MÄRTENS, Olev e MIN, Mart. Multifrequency bio-impedance measurement: undersampling approach. **Proceedings of the 6th Nordic Signal Processing Symposium**, 2004.
- MARVASTI, Farokh; AMINI, Arash; HADDADI, Farzan; SOLTANOLKOTABI, Mahdi; KHALAJ, Babak H.; ALDROUBI, Akram; SANEI, Saeid e CHAMBERS, Janathon. A unified approach to sparse signal processing. **EURASIP Journal on Advances in Signal Processing**, Dez. 2012.
- MILLER, Steven J. The Method of Least Squares. **Brown University**, 2006.
- MISHALI, Moshe; ELDAR, Yonina C. e ELRON, Asaf J. Xampling: Signal Acquisition and Processing in Union of Subspaces. **IEEE Transactions on Signal Processing**, vol. 59, no. 10, pp. 4719-4734, Out. 2011.
- OLJACA, Miro e SANBORN, Keith. **Determining Minimum Acquisition Times for SAR ADCs When a Step Function is Applied to the Input**. Texas Instruments, 2010.
- PANKIEWICZ, Pawel J.; ARILDSEN, Thomas e LARSEN, Torben. Sensitivity of the random demodulation framework to filter tolerances. **2011 19th European Signal Processing Conference**, Barcelona, 2011, pp. 534-538.

- DAPONTE, Pasquale; DE VITO, Luca; RAPUANO, Sergio e TUDOSA, Ioan. Analog-to-information converters in the wideband rf measurement for aerospace applications: current situation and perspectives. **IEEE Instrumentation & Measurement Magazine**, vol. 20, no. 1, pp. 20-28, February 2017.
- PAULUZZI, David. R. e BEAULIEU, Norman C. **A comparison of SNR estimation techniques for the AWGN channel**. *IEEE Transactions on Communications*, vol. 48, no. 10, pp. 1681-1691, Out. 2000.
- PENG, Hao; PRODIC, Aleksandar; ALARCÓN, Eduard, MAKSIMOVIC, Dragan. Modeling of quantization effects in digitally controlled DC-DC converters. **2004 IEEE 35th Annual Power Electronics Specialists Conference**, pp. 4312-4318, vol.6, 2004.
- RAGHEB, Tamer; LASKA, Jason N.; NEJATI, Hamid; KIROLOS, Sami; BARANIUK, Richard G.; MASSOUD, Yehia. A prototype hardware for random demodulation based compressive analog-to-digital conversion. **51st Midwest Symposium on Circuits and Systems**, Knoxville, TN, 2008, pp. 37-40.
- ROSSUM, Guido V. e DRAKE, Fred L. Jr. **The Python Language Reference Manual**. Network Theory Ltd, 2003.
- STMicroelectronics. **STM32F4DISCOVERY**. Disponível em: <http://www.st.com/en/evaluation-tools/stm32f4discovery.html>. Acesso em: 08, out. 2017.
- SUNDSTRÖM, Timmy; MURMANN, Boris e SVENSSON, Christer. Power Dissipation Bounds for High-Speed Nyquist Analog-to-Digital Converters. **IEEE Transactions on Circuits and Systems I: Regular Papers**, vol. 56, no. 3, pp. 509-518, Março 2009.
- TOCCI, Ronald J.; WIDMER, Neal e MOSS, Greg. **Digital Systems: Principles and Applications**. 11^a ed. Pearson, 2010.
- VERHELST, Marian e BAHAI, Ahmad. Where Analog Meets Digital: Analog to Information Conversion and Beyond. **IEEE Solid-State Circuits Magazine**, vol. 7, no. 3, pp. 67-80, Verão 2015.
- VETTERLI, Martin; KOVACEVIC, Jelena e GOYAL, Vivek G. **Foundations of Signal Processing**. Cambridge University Press, 2014.
- VIDYASAGAR, Mathukumalli. A tutorial introduction to compressed sensing. *2016 IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, NV, 2016, pp. 5091-5104.
- WALDEN, Robert H. Analog-to-digital converter survey and analysis. **IEEE Journal on Selected Areas in Communications**, vol. 17, no. 4, pp. 539-550, Abril 1999.
- WE, Tzu-Fan; DEY, Sourya e CHEN, Mike S. W. A Nonuniform Sampling ADC Architecture With Reconfigurable Digital Anti-Aliasing Filter. **IEEE Transactions on Circuits and Systems I: Regular Papers**, vol. 63, no. 10, pp. 1639-1651, Out. 2016.
- YIN, Penghang; ESSER, Ernie e XIN, Jack. Ratio and Difference of l1 and l2 Norms and Sparse Representation with Coherent Dictionaries. **Communications in Information and Systems**. 2. 87-109. 2014.

YÜCEK, Tevfik e ARSLAN, Huseyin. A survey of spectrum sensing algorithms for cognitive radio applications. **IEEE Communications Surveys & Tutorials**, vol. 11, no. 1, pp. 116-130, Primeiro Quadrimestre 2009.

APÊNDICE A - VALIDAÇÃO DOS MODELOS

A.1 RUÍDO

Para validar o modelo do ruído, dois métodos foram utilizados. O primeiro método consiste em determinar o sinal de ruído e calcular seu SNR, em relação ao sinal de entrada. O segundo método utilizado consiste em analisar o histograma e espectro de frequência do sinal de ruído obtido, afim de verificar se a energia do ruído concentra-se em alguma frequência.

A Figura 32 mostra o sinal senoidal considerado para os testes, acrescido de um ruído de 40 dB. O sinal, uma onda senoidal de 60 Hz, foi gerado com uma frequência de discretização de 14,4 kHz. Após a geração do sinal de ruído, baseado no SNR informado e no sinal senoidal, o processo inverso foi realizado. A partir do sinal de entrada e do sinal de ruído obtido, a energia do sinal e do ruído foram calculadas com a Equação 2 e o SNR foi calculado com a Equação 1. Para o sinal apresentado na Figura 32, o resultado obtido foi de 39,82 dB, valor próximo ao desejado (40 dB).

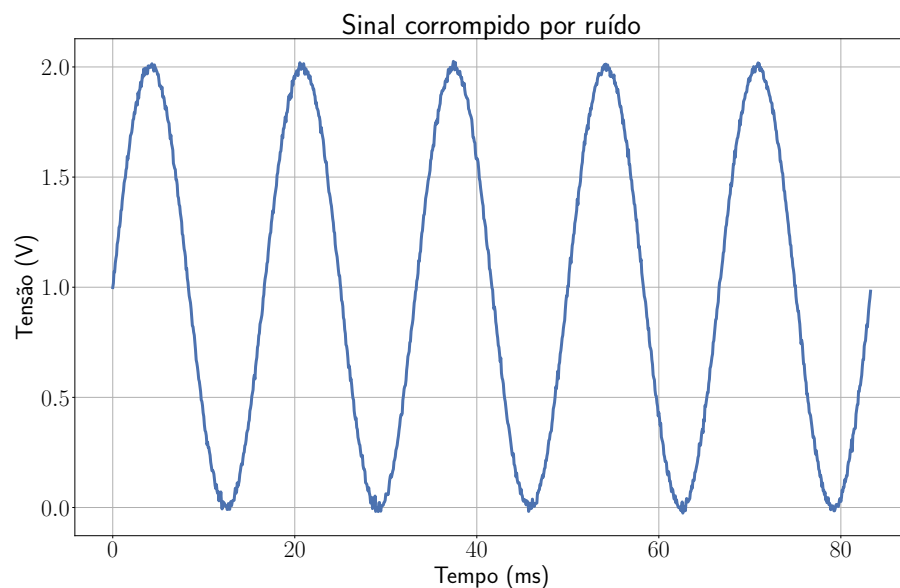


Figura 32 – Sinal com SNR de aproximadamente 40 dB.

Fonte: Autoria própria

Em seguida, o histograma e o espectro de frequência do sinal de ruído calculado foram obtidos, conforme indicado na Figura 33.

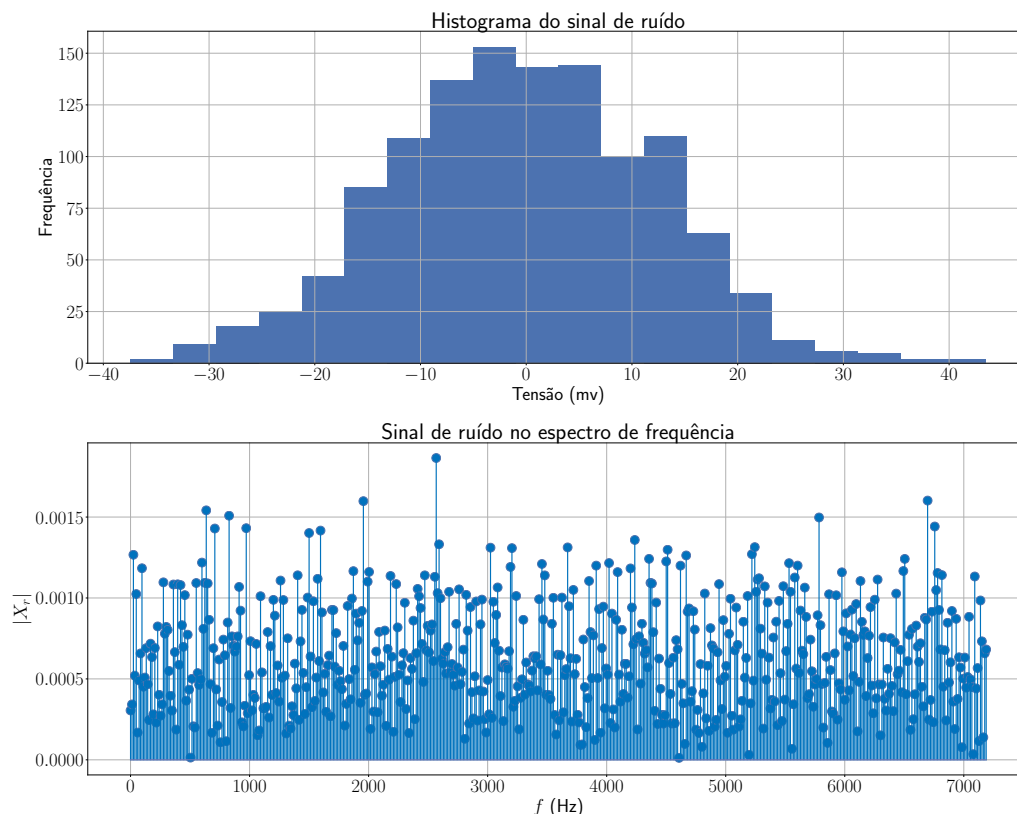


Figura 33 – Histograma (superior) e espectro de frequência (inferior) do sinal de ruído.

Fonte: Autoria própria

A Figura 33 mostra que a energia do ruído não concentra-se em nenhuma frequência específica, uma vez que o histograma mostra uma distribuição que se aproxima de uma distribuição normal com média zero. Além disso, verificou-se que quanto maior a frequência de discretização, o histograma do ruído se aproxima melhor de uma distribuição normal, com média zero. Dessa forma, obtém-se um erro menor entre o SNR desejado e o obtido posteriormente.

A.2 CONVERSOR ANALÓGICO-DIGITAL

Para validar o modelo do processo de quantização do ADC, o modelo implementado em Python foi comparado com o modelo disponível na ferramenta *Simulink*, do Matlab, que possui um bloco de quantização. Para a simulação do ADC, considerou-se um ADC com uma resolução de 4 bits ($N = 4$) e com uma referência de 5 V ($v_{ref} = 5$). Na entrada, foi aplicado um sinal crescente (rampa), começando em 1 V e terminando em 3,75 V.

A Figura 34 mostra a simulação do bloco do ADC, utilizando o modelo implementado em Python (conforme descrito na seção 2.1.2). O bloco do ADC foi configurado

Na Figura 34, o sinal em rampa é representado pela reta pontilhada em azul. As amostras que foram quantizadas são representadas pelos pontos, também na cor azul. O resultado da quantização é representado pela curva em degrau, na cor verde.

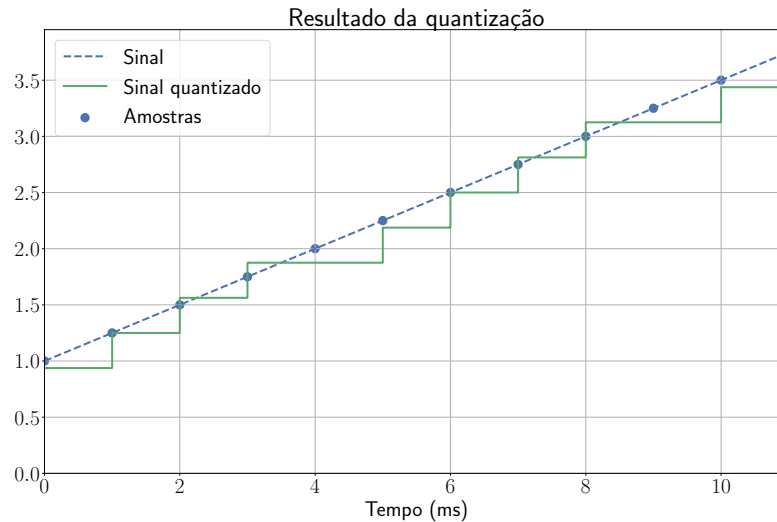


Figura 34 – Simulação do ADC.

Fonte: Autoria própria

A Figura 35 mostra a comparação entre os resultados obtidos com a simulação em Python e com o Matlab. Como a curva obtida com a simulação em Python corresponde a curva obtida com o software *Matlab*, conclui-se que o modelo do ADC está correto.

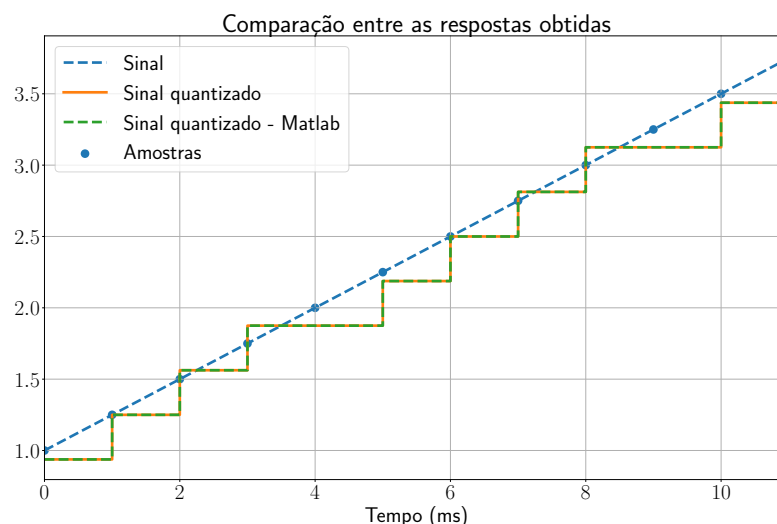


Figura 35 – Validação do modelo do ADC.

Fonte: Autoria própria

A.3 BLOCO DE PROCESSAMENTO DO SISTEMA DE SUBAMOSTRAGEM

Para validar o modelo do bloco de processamento, que calcula os coeficientes de Fourier nas frequências específicas, os resultados obtidos com a simulação implementada em Python foram comparados com o resultado obtido por um algoritmo que obtém a FFT de um sinal. No caso, foi utilizado um algoritmo fornecido pelo pacote Numpy. No entanto, a simulação do bloco de processamento do sistema de subamostragem é feita com uma frequência maior do que a taxa de Nyquist, uma vez que o modelo do bloco é genérico (ver seção 2.2.2.2).

A Figura 36 mostra o sinal utilizado (esquerda) e a comparação entre a magnitude dos coeficientes de Fourier obtida com os dois algoritmos (direita).

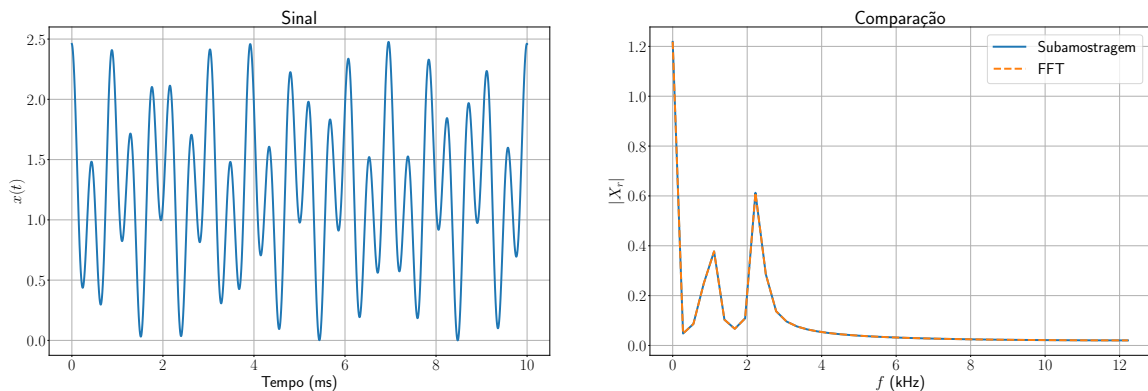


Figura 36 – Sinal (esquerda) e coeficientes de Fourier (direita).

Fonte: Autoria própria

É possível notar que os coeficientes de Fourier obtido com os dois métodos são iguais. Portanto, conclui-se que a implementação das Equações 11–14 está correta.

A.4 RESPOSTA DO FILTRO

A validação da resposta ao impulso do filtro foi realizada obtendo a função analítica e comparando com a resposta gerada pelo *software* Matlab. O filtro utilizado na simulação do demodulador aleatório é um filtro passa-baixas, de quarta ordem, com coeficientes de Butterworth. A função de transferência do filtro é dada por:

$$\frac{Y(s)}{X(s)} = \frac{1}{(s^2 + 0.765s + 1)(s^2 + 1.848s + 1)}. \quad (33)$$

A Equação 33 apresenta um filtro normalizado, ou seja, com frequência de corte ω_0 igual a 1 rad/s. De forma mais genérica, a Equação 33 pode ser escrita em sua

forma não normalizada:

$$\frac{Y(s)}{X(s)} = \frac{1}{((s/\omega_0)^2 + \alpha(s/\omega_0) + 1)((s/\omega_0)^2 + \beta(s/\omega_0) + 1)}. \quad (34)$$

Aplicando a transformada inversa de Laplace na Equação 34, com $X(s) = 1$, resulta na resposta do filtro no domínio do tempo. Separando a Equação 33 em frações parciais, a resposta obtida para o filtro pode ser vista na Equação 35.

$$h(t) = \frac{2}{\beta - \alpha} \left(-\frac{1}{k_2} e^{-\alpha k_1 t} \cos(k_2 k_1 t + \theta_1) + \frac{1}{k_3} e^{-\beta k_1 t} \cos(k_3 k_1 t + \theta_2) \right), \quad (35)$$

em que:

$$k_1 = \frac{\omega_0}{2}, \quad k_2 = \sqrt{4 - \alpha^2}, \quad k_3 = \sqrt{4 - \beta^2}, \quad (36)$$

$$\tan \theta_1 = -\frac{\alpha}{k_2}, \quad \tan \theta_2 = -\frac{\beta}{k_3}. \quad (37)$$

A Figura 37 mostra a comparação entre os resultados obtidos com a resposta analítica (Equação 35) e a resposta obtida com a `impz` do Matlab. Como é possível notar, as curvas obtidas com a Equação 35 e com o Matlab se sobrepõem, indicando que a resposta analítica do filtro está correta.

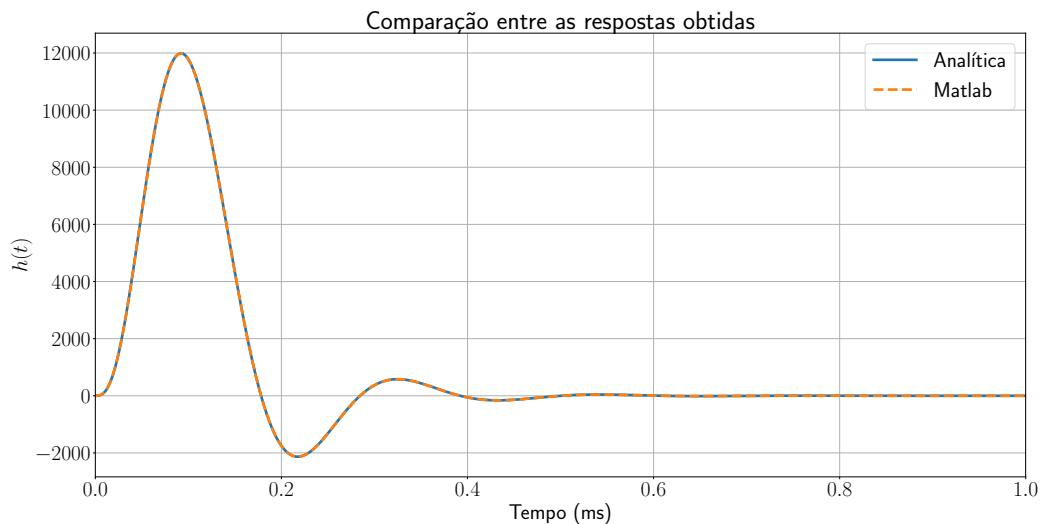


Figura 37 – Resposta do filtro ao impulso.

Fonte: Autoria própria