



Universidade Federal da Paraíba
Centro de Energias Alternativas e Renováveis
Programa de Pós-Graduação em Engenharia Elétrica



Hugo Bruno Santos Araujo

Análise de Performance de Algoritmos de Reconstrução para um Conversor Analógico para Informação

João Pessoa

2022

Hugo Bruno Santos Araujo

Análise de Performance de Algoritmos de Reconstrução para um Conversor Analógico para Informação

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Cleonilson Protásio de Souza

João Pessoa

2022

Catálogo na publicação
Seção de Catalogação e Classificação

A663a Araujo, Hugo Bruno Santos.

Análise de performance de algoritmos de reconstrução para um conversor analógico para informação / Hugo Bruno Santos Araujo. - João Pessoa, 2022.

58 f. : il.

Orientação: Cleonilson Protásio de Souza.
Dissertação (Mestrado) - UFPB/CEAR.

1. Engenharia elétrica. 2. Amostragem compressiva.
3. Conversor analógico para informação. 4. Sinais esparsos. 5. Algoritmos de reconstrução. I. Souza, Cleonilson Protásio de. II. Título.

UFPB/BC

CDU 621.3(043)

UNIVERSIDADE FEDERAL DA PARAÍBA – UFPB
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS – CEAR
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA – PPGE

A Comissão Examinadora, abaixo assinada, aprova a Dissertação

**ANÁLISE DE PERFORMANCE DE ALGORITMOS DE RECONSTRUÇÃO PARA
UM CONVERSOR ANALÓGICO PARA INFORMAÇÃO**

Elaborada por

HUGO BRUNO SANTOS ARAÚJO

como requisito parcial para obtenção do grau de
Mestre em Engenharia Elétrica.

COMISSÃO EXAMINADORA



PROF. DR. CLEONILSON PROTÁSIO DE SOUZA
Orientador – UFPB

FBSC

PROF. DR. FABRÍCIO BRAGA SORES DE CARVALHO
Examinador Interno – UFPB

Documento assinado digitalmente

gov.br

VERONICA MARIA LIMA SILVA

Data: 29/11/2022 15:21:54-0300

Verifique em <https://verificador.iti.br>

PROF. DR. VERONICA MARIA LIMA SILVA
Examinador Externo – UFPB

Dedico este trabalho à minha mãe Izaete Cordeiro dos Santos, e a minha tia Rutileide de Araujo Silva, que sempre me apoiaram em todas as escolhas que fiz em minha vida.

Agradecimentos

Primeiramente queria agradecer a Deus por me permitir desenvolver este trabalho durante uma época tão conturbada no mundo e sempre mantendo o foco.

Gostaria de agradecer ao meu orientador Cleonilson Protásio de Souza por todo apoio e empenho dado até o momento no mestrado, sem os quais este trabalho não seria possível.

Aos professores que tive contato nas disciplinas que fiz durante o mestrado. Todo o conhecimento o adquirido com vocês foi essencial para chegar aqui.

Agradeço também aos meus familiares, em especial minha mãe Izaete e a minha tia Rutilaide, que são a minha base em todos os momentos.

De antemão também queria agradecer aos professores convidados para minha banca pela disponibilidade, pois sei que terão muito a contribuir.

Por fim, gostaria de agradecer a todos que contribuíram de forma direta e indireta para realização deste trabalho.

Resumo

A abordagem convencional para amostragem de sinais analógicos para digital segue o Teorema de Nyquist, em que a taxa de amostragem, chamada de Taxa de Nyquist, deve ser pelo menos duas vezes o valor da frequência máxima do sinal analógico. Em termos práticos, no geral, as taxas de amostragem de conversores analógicos digitais (ADC) são bem superiores à taxa de Nyquist a fim de otimizar a recuperação do sinal, porém aumenta-se também a necessidade de memória e de poder de processamento, os custos dos sistemas como um todo e um maior consumo de energia. A Amostragem Compressiva (AC) é uma técnica que explora a esparsidade de um sinal em um determinado domínio, i.e., a informação do sinal se concentra em poucos coeficientes, e a maior parte de seus coeficientes é igual ou próxima de zero. O Conversor Analógico para Informação (AIC) é o dispositivo que implementa o conceito de amostragem compressiva, em que, ao passo que realiza a amostragem do sinal analógico de entrada, um processo de compressão é realizado e, assim, obtém-se como saída uma versão digitalizada e comprimida do sinal de entrada que será transmitido e reconstruído no receptor. Diante do exposto, este trabalho tem como objetivo a análise de performance dos algoritmos de reconstrução Busca de Base (BP), Busca por Correspondência Ortogonal (OMP) e Amostragem Compressiva com Busca por Correspondência (CoSaMP) para uma configuração adaptada de AIC baseada no Pré-Integrador de Modulação Aleatória (RMPI). Simulações foram feitas no Proteus e no Simulink para validação da configuração do AIC, e sinais de um tom e de dois tons foram reconstruídos com componentes de frequência aproximados dos sinais originais. O desempenho dos três métodos de reconstrução foi avaliado com a métrica do erro quadrático médio (MSE).

Palavras-chave: Amostragem Compressiva, Conversor Analógico para Informação, Sinais Esparsos, Algoritmos de Reconstrução.

Abstract

The conventional approach to sampling analog signals to digital follows the Nyquist Theorem, in which the sampling rate, called Nyquist rate, must be at least two times bigger than the maximum frequency of the analog signal. In practical terms, generally, analog to digital converters' (ADC) sampling rates are much higher to optimize signal reconstruction, however, they increase the need for memory and processing power, systems costs, and also increase energy consumption. Compressive Sensing (CS) is a technique that takes advantage of signal sparsity in a given domain, that is, the number of non-zero values of a signal, and captures only the regions that concentrate information, with sampling rates lower than what the Nyquist Theorem says. A device that implements the CS model is the Analog-to-Information Converter (AIC), which obtains an output that is a digitized and compressed version of the input. The main objective is to analyze the performance of reconstruction algorithms - Basis Pursuit (BP), Orthogonal Matching Pursuit (OMP) and Compressive Sampling Matching Pursuit (CoSaMP) - for an adapted AIC configuration based on Random Modulator Pre-Integration (RMPI). Simulations were done on Proteus and Simulink to validate the AIC configuration used, and single-tone and duo-tones signals were reconstructed with frequency components similar to the original signals. Mean squared error (MSE) was calculated for the three reconstruction methods to determine which one had better performance.

Keywords: Compressed Sensing, Analog-to-Information Converter, Sparse Signals, Reconstruction Algorithms.

Lista de siglas

AC *Amostragem Compressiva*

ADC *Conversor Analógico-digital*

AIC *Conversor Analógico para Informação*

CI *Circuito Integrado*

DFT *Transformada Discreta de Fourier*

ECG *Eletrocardiograma*

GF *Campos de Galois*

IoT *Internet das Coisas*

LFSR *Registrador de Deslocamento com Realimentação Linear*

MSE *Erro Quadrático Médio*

NUS *Amostrador Não Uniforme*

PC *Computador Pessoal*

PCI *Placa de Circuito Impresso*

RD *Demodulador Aleatório*

RF *Rádio-frequência*

RMPI *Pré-Integrador de Modulação Aleatória*

RS *Amostragem Aleatória*

RSSF *Redes de Sensores Sem Fio*

S/H *Sample-and-Hold*

XNOR *Ou-exclusivo*

XOR *Não Ou-exclusivo*

ZOH *Zero-Order Hold*

Sumário

Lista de Figuras	xi
1 Introdução	13
1.1 Motivação	14
1.2 Trabalhos Relacionados	15
1.3 Objetivos	15
1.3.1 Objetivo Geral	15
1.3.2 Objetivos Específicos	15
1.4 Organização do Texto	16
2 Revisão Bibliográfica	17
2.1 Amostragem Compressiva	17
2.1.1 Esparsidade	18
2.1.2 Medidas e Matrizes na Amostragem Compressiva	20
2.1.3 Incoerência	22
2.1.4 Reconstrução	23
2.1.4.1 Abordagem Convexa	25
2.1.4.2 Abordagem Gulosa/Míope	27
2.2 Arquiteturas de AIC	28
2.2.1 Amostragem Aleatória	29
2.2.2 Demodulador Aleatório	30
2.2.3 Pré-Integrador de Modulação Aleatória	32
3 Metodologia da Pesquisa	35
3.1 Configuração Geral do Sistema	35
3.1.1 AIC no Proteus	36
3.1.1.1 Gerador de Sequências Pseudo-aleatórias	36
3.1.1.2 Circuito condicionador de sinal	38
3.1.1.3 Multiplicador	39
3.1.1.4 Filtro Passa-baixas (Integrador)	39
3.1.2 AIC no Simulink	40

4 Resultados e Discussões	43
4.1 Simulações no Simulink e Proteus	43
4.2 Reconstrução de um sinal de um tom	47
4.3 Reconstrução de um sinal de dois tons	49
5 Conclusões	54
5.1 Trabalhos Futuros	55
5.2 Publicações	55
Referências	56

Lista de Figuras

1	Exemplo de Compressão de Dados	18
2	Aplicação da amostragem compressiva	18
3	Sinal senoidal com dois coeficientes no domínio da frequência	19
4	Medidas no domínio não esparsa	21
5	Medidas no domínio esparsa	21
6	Matrizes de medidas e de transformação	21
7	Representação geométrica de várias normas l_p	24
8	Representação geométrica 2-D de várias normas l_p	26
9	Etapas de um algoritmo guloso	27
10	Framework da amostragem compressiva	28
11	Representação de um sinal cossenoidal no tempo.	29
12	Diagrama de blocos do Amostrador Não Uniforme	30
13	Diagrama de blocos do Demodulador Aleatório com filtro passa-baixas	31
14	Representação dos sinais em cada etapa do RD	32
15	Diagrama de blocos do RMPI	33
16	Diagrama de blocos do sistema simulado	35
17	Arquitetura de Galois para o polinômio $1 + x + x^4 + X^5 + X^6$	37
18	Arquitetura de Fibonacci para o polinômio $1 + x + x^4 + X^5 + X^6$	38
19	Design do LFSR no Proteus	38
20	Circuito condicionador de sinal no Proteus	39
21	Multiplicador no Proteus	40
22	Filtro passa-baixas com chave analógica	40
23	Diagrama de blocos do AIC no Simulink	41
24	Bloco Condicionador de sinal no Simulink	41
25	Bloco Integrador no Simulink	42
26	Bloco ADC no Simulink	43
27	Diagrama de blocos do AIC no Simulink	44
28	Sinal senoidal com <i>delay</i> em (a); Sinal do LFSR em (b)	44
29	Sinais do LFSR do Proteus (em azul) e do Simulink (em laranja)	45
30	Sinais do multiplicador do Proteus (em azul) e do Simulink (em laranja)	45

31	Sinais do integrador do Proteus (em azul) e do Simulink (em laranja) para um 1 segundo	46
32	Sinais do integrador do Proteus (em azul) e do Simulink (em laranja) para 5 segundos	46
33	Fluxograma da reconstrução para medidas feitas no domínio não esparsos	48
34	Sinal de um tom original (a) e sinais reconstruídos com o CVX (b), OMP (c), e CoSaMP (d) no domínio da frequência	49
35	Sinal de um tom original (a) e sinais reconstruídos com o CVX (b), OMP (c), e CoSaMP (d) no domínio do tempo	50
36	Sinal de dois tons original (a) e sinais reconstruídos com o CVX (b), OMP (c), e CoSaMP (d) no domínio da frequência	52
37	Sinal de dois tons original (a) e sinais reconstruídos com o CVX (b), OMP (c), e CoSaMP (d) no domínio do tempo	53
38	Percentual de testes com menor MSE por método de reconstrução	53

1 Introdução

Nos últimos anos, a Internet das Coisas (IoT, do inglês *Internet of Things*) se tornou uma das tecnologias mais importantes do século 21. Seja com a utilização de dispositivos domésticos inteligentes, automóveis, ou dispositivos móveis, a sociedade está cada vez mais conectada.

Segundo a projeção do IoT Analytics (SINHA, 2021), apesar da crise de *chips* semicondutores, que começou no início da pandemia de COVID-19, e do impacto extenso da pandemia na logística de fornecimento, o número de dispositivos conectados até 2025 será, provavelmente, de mais de 27 bilhões.

No contexto de IoT, que tem como premissa que objetos do dia-a-dia podem gerar, processar e enviar dados via Internet sem nenhuma intervenção humana (SOUZA; BAIOCCHI, 2018), tem-se como implicação principal a produção de uma quantidade extraordinária de dados gerados por objetos, e não por pessoas. A indústria IoT é prevista ser fundamental para a sociedade, pois pode ser empregada em diversos campos tecnológicos, tais como: cidades inteligentes (*Smart Cities*), agricultura inteligente, monitoramento ambiental, indústria inteligente (*Smart Factory* ou *Industry 4.0*), casas inteligentes (*Smart Homes*), logística, medidores inteligentes (*Smart Metering*), veículos autônomos e seus sistemas de suporte, redes de sensores sem fio (RSSF), entre outros.

Conversores analógico-digitais (ADC) são componentes essenciais em sistemas que precisam adquirir e processar sinais físicos (contínuos tanto em amplitude quanto no tempo) usando circuitos digitais. Nesses casos, os conversores de dados promovem um *link* entre o mundo analógico e o domínio digital, e vice-versa (SILVA; SOUZA; CATUNDA, 2014), e podem ser usados como interface entre sensores e microcontroladores ou processadores digitais de sinais.

Os ADCs seguem o teorema de Nyquist no processo de digitalização de dados, que diz que a taxa de amostragem utilizada deve ser pelo menos duas vezes o valor da frequência máxima do sinal analógico (SHANNON, 1948). No entanto, em termos práticos, as taxas de amostragem de ADC são bem superiores à taxa de Nyquist a fim de otimizar a recuperação do sinal gerando altas quantidade de dados, e com altas taxas de amostragem, aumenta-se também a necessidade de memória e de poder de processamento (RAPUANO, 2016).

1.1 Motivação

Diante do contexto apresentado, nos últimos anos uma nova técnica de amostragem foi proposta: a Amostragem Compressiva (AC, do inglês *Compressed Sensing* (CS)), que explora a esparsidade de certas classes de sinais (CANDÈS; WAKIN, 2008; DONOHO, 2006), e que possui diversas aplicações, tais como: aquisição e compressão de dados (CANDÈS; WAKIN, 2008), reconstrução de imagens (LI; ZHANG, 2016; SALAN; MURALIDHARAN, 2017; S.; K.V., 2019), redes de rádio cognitivas (SALAHADINE; KAABOUCHE; GHAZI, 2018), reconhecimento de alvos por radar, (BLACKNELL, 2016), monitoramento de eletrocardiograma (ECG) (DJELOUAT *et al.*, 2017), entre outras.

A teoria da amostragem compressiva afirma que é possível recuperar sinais com uma taxa de amostragem menor do que a taxa de Nyquist dado que o sinal a ser amostrado seja esparso em um dado domínio (BARANIUK, 2007; DONOHO, 2006). Um sinal pode ser considerado esparso quando é possível caracterizá-lo em poucos coeficientes não-nulos após uma transformação entre domínios (STANKOVIĆ *et al.*, 2019). Um exemplo clássico de sinal esparso no tempo é o sinal de ECG, em que no domínio do tempo a maior parte do sinal é zero (ou próximo de zero) e em uma pequena fração existem pulsos cardíacos. O sinal senoidal é outro exemplo clássico, que é esparso no domínio da frequência pois apresenta picos concentrados de energia.

Um dispositivo que implementa o conceito de amostragem compressiva é o Conversor Analógico para Informação (AIC, do inglês *Analog-to-Information Converter*) (RAPU-ANO, 2016), em que, ao passo que realiza a amostragem do sinal analógico de entrada, um processo de compressão (AC) é realizado e, assim, é obtida como saída uma versão digitalizada e comprimida do sinal de entrada (SILVA, 2019; SILVA *et al.*, 2020). Cada saída digitalizada e comprimida é denominada de “medição”. Para a reconstrução do sinal analógico de entrada, as medições devem ser aplicadas em um algoritmo de reconstrução.

Um fato importante da aplicabilidade da AC/AIC é que a etapa de reconstrução do sinal, que possui alta complexidade computacional, pode ser realizada em um dispositivo diferente do que gerou as medidas comprimidas, por exemplo, uma máquina computacional de maior porte interconectada na internet ou uma nuvem computacional.

O objetivo desta Dissertação é analisar a performance de algoritmos de reconstrução no que se refere ao Erro Quadrático Médio (MSE) para um conversor analógico para

informação baseado na configuração RMPI.

1.2 Trabalhos Relacionados

Em [Djelouat et al. \(2017\)](#), foi proposta a incorporação da Amostragem compressiva em uma plataforma de monitoramento para ECG baseada em IoT, que também é capaz de detectar anormalidades em batimentos cardíacos usando diferentes algoritmos de reconhecimento de padrões.

[\(DONG; WU, 2021\)](#) fez um estudo e comparação de dois algoritmos de reconstrução para a Amostragem Compressiva: a Busca por Correspondência com Representação Esparsa e a Busca por Correspondência Ortogonal.

[\(DONG; WU, 2021\)](#) discute o problema de otimização da matriz de medidas na Amostragem Compressiva e propõe um método de otimização de anelamento.

Em [Nouasria e Et-Tolba \(2018\)](#), foi investigada uma nova abordagem de sensoriamento que leva em consideração a tomada de medidas no domínio esparso.

[\(S.; K.V., 2019\)](#) investigou a aplicação da Amostragem Compressiva em Imagens por ressonância magnética (MRI) e observou a redução de tempos de varredura.

Neste trabalho de Dissertação é feito a análise da performance de três algoritmos de reconstrução com medidas obtidas de um conversor analógico para informação baseado em um RMPI.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral consiste na análise de performance de algoritmos de reconstrução para um conversor analógico simulado, utilizando o erro médio quadrático como métrica de avaliação. A configuração do AIC utilizada foi baseada no Pré-Integrador de Modulação Aleatória (RMPI) e foi tanto no Simulink quanto no Proteus, o último visando aproximar do comportamento em *hardware*.

1.3.2 Objetivos Específicos

Dentre os objetivos específicos desta proposta, pode-se destacar:

- Estudar os fundamentos matemáticos da amostragem compressiva e as arquiteturas de conversores analógico para informação;
- Elaborar uma configuração de AIC baseada no Pré-Integrador de Modulação Aleatória (RMPI);
- Validar o AIC elaborado no Simulink e no Proteus, visando aproximar do comportamento em *hardware*;
- Reconstruir sinais de um tom e dois tons a partir de medidas geradas pelo AIC proposto;
- Avaliar a reconstrução de sinais de um tom e dois tons ajustando parâmetros do AIC proposto;
- Avaliar o desempenho de algoritmos de reconstrução utilizando o erro médio quadrático como métrica de avaliação.

1.4 Organização do Texto

O presente trabalho está organizado em cinco capítulos, de acordo com a seguinte estrutura: no Capítulo 2 são apresentados os fundamentos da Amostragem Compressiva, os conceitos de Esparsidade, Incoerência, Reconstrução, Arquiteturas do AIC. No Capítulo 3 é abordada a metodologia utilizada para a validação do AIC proposto. No Capítulo 4 são apresentados e discutidos os resultados de simulação para os cenários considerados. Por fim, as conclusões deste trabalho e os trabalhos futuros são descritos no Capítulo 5.

2 Revisão Bibliográfica

Neste capítulo são descritos os fundamentos da Amostragem Compressiva, os conceitos de Esparsidade, Medidas e Matrizes na Amostragem Compressiva, Incoerência, Reconstrução, e os tipos de Arquiteturas do AIC, com suas características e diferenças.

2.1 Amostragem Compressiva

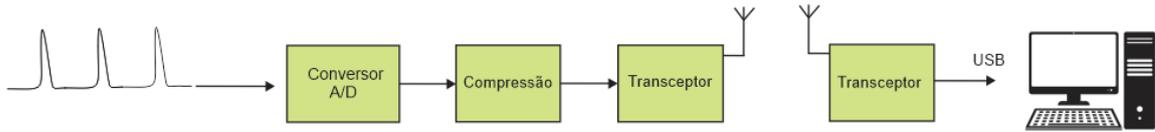
A abordagem convencional de amostragem de sinais analógicos para digital segue o Teorema de Nyquist, em que a taxa de amostragem, normalmente conhecida como taxa de Nyquist, deve ser pelo menos duas vezes o valor da frequência máxima do sinal analógico (SHANNON, 1948). Caso o Teorema de Nyquist não seja obedecido, o sinal amostrado terá em sua composição frequências indesejadas e não será reconstruído com precisão. Para garantir que apenas a faixa de frequências de interesse do sinal de entrada seja obtida no sinal resultante amostrado, convencionalmente utilizam-se filtros anti-recobrimento (*anti-aliasing*) na entrada do sinal.

Geralmente, as taxas de amostragem de conversores analógicos-digitais (ADC) são bem superiores à taxa de Nyquist a fim de otimizar a recuperação do sinal. Entretanto, com altas taxas de amostragem, a grande quantidade de amostras obtidas ocasiona diretamente a necessidade de mais memória e de maior poder de processamento, aumentando-se assim o custo do sistema e causando maiores consumos de energia (RAPUANO, 2016).

A Figura 1 demonstra a aquisição de um sinal de ECG, passa por um ADC tradicional seguido de compressão, e envia os dados por um rádio transceptor para um computador. Com a compressão dos dados, há uma redução no consumo de energia gasto com a transmissão pelo transceptor, mas a adição de um bloco de compressão pode implicar em um aumento do consumo de energia do dispositivo.

A Amostragem Compressiva é uma técnica que propõe a aquisição de sinais sem a limitação de Nyquist. A principal ideia da amostragem compressiva é que é possível recuperar sinais de certas classes utilizando menos amostras do que a taxa de amostragem indicada pelo teorema de Nyquist. Particularmente, a AC permite a reconstrução de sinais que são considerados esparsos (conceito detalhado na Seção 2.1.1) ou que podem ser comprimidos quando expressos em uma determinada base (CANDES; WAKIN, 2008).

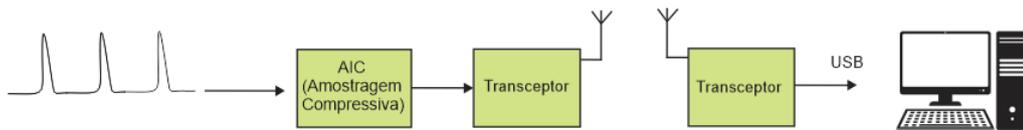
Figura 1: Exemplo de Compressão de Dados



Fonte: Elaborado pelo autor.

Com a aplicação da amostragem compressiva no exemplo demonstrado na Figura 1, não há necessidade para um bloco de ADC e um bloco de compressão. Tem-se apenas um bloco único que amostra e comprime concomitantemente o sinal, como pode ser visto na Figura 2.

Figura 2: Aplicação da amostragem compressiva



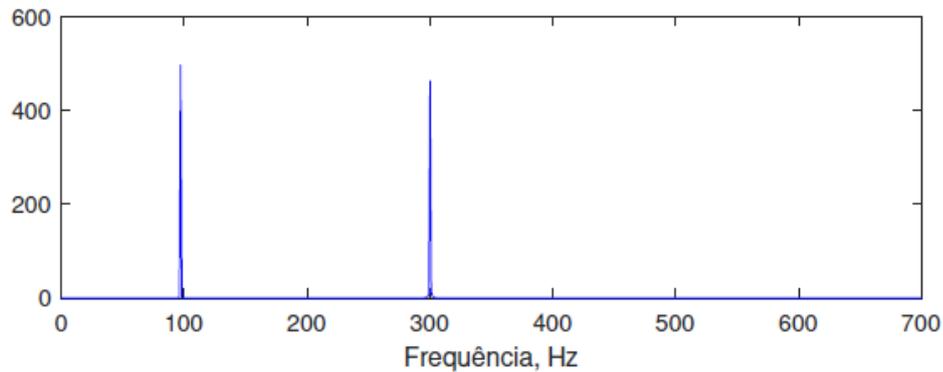
Fonte: Elaborado pelo autor.

2.1.1 Esparsidade

Sinais podem ser representados em diferentes domínios. Um sinal senoidal, por exemplo, pode ser representado por um único coeficiente no domínio da frequência, e também por várias amostras no domínio do tempo. A Figura 3 mostra um sinal senoidal no domínio da frequência com dois coeficientes.

Considere um sinal finito, unidimensional e discreto no tempo x , que pode ser visto como um vetor $N \times 1$ em R^N com elementos $x[n] = 1, 2, \dots, N$ e que é expansível em uma base ortonormal de $N \times 1$ vetores $\{\psi_i\}_{i=1}^N$ (BARANIUK, 2007; CANDÈS; WAKIN, 2008). Usando a matriz $\psi = [\psi_1 | \psi_2 | \psi_3 | \dots | \psi_N]$ de tamanho $N \times N$ com os vetores ψ_i sendo as colunas dessa matriz, o sinal x pode ser expresso como mostra a Equação 1, em que ψ em $R^{N \times N}$ representa a base de transformação linear, e s em R^N é a representação esparsa do sinal x .

Figura 3: Sinal senoidal com dois coeficientes no domínio da frequência



Fonte: Elaborado pelo autor.

$$x = \sum_{i=1}^N s_i \psi_i \text{ ou } x = \psi s \quad (1)$$

Os vetores x e s representam o mesmo sinal, sendo x a sua representação no domínio do tempo e s na base ψ . Quando um sinal é dito esparsos, a maioria dos seus coeficientes é igual a zero ou são pequenos o suficiente para serem ignorados sem que haja perdas significativas. O sinal x é K -esparsos se somente K coeficientes de s na Equação 1 são diferentes de zero e $(N - K)$ são iguais a zero (BARANIUK, 2007; SILVA, 2019). Se a base de transformação for genérica, tais como a de Fourier ou Wavelet, então apenas alguns elementos ativos em s são necessários para a reconstrução do sinal x , reduzindo os dados necessários para armazenar ou transmitir o sinal (BRUNTON; KUTZ, 2017). No caso da Figura 3, o sinal possui esparsidade $K = 2$.

O caso de interesse é quando $K \ll N$. O sinal x é comprimível se a representação na Equação 1 possui poucos coeficientes maiores que zero e muitos coeficientes iguais ou próximos de zero. (BARANIUK, 2007).

Segundo Mallat (1999), o fundamento da codificação com transformadas é o fato de que sinais comprimíveis podem ser bem aproximados por representações com esparsidade K . Em sistemas de aquisição de dados, a codificação com transformadas tem um papel central que consiste em: o sinal x completo com tamanho N é adquirido, uma representação s de N amostras é gerada, e então os coeficientes próximos ou iguais a zero são descartados. Se as amostras descartadas são $(N - K)$, então o sinal codificado precisará

de somente K elementos, com $K \ll N$ (BARANIUK, 2007; RAPUANO, 2016).

O fenômeno da compressão levanta várias questões naturais, tais como: Por que é necessário adquirir todas as amostras de um sinal se a maior parte será descartada? Será que é possível medir somente a parte que não será descartada? A amostragem compressiva tenta lidar com essa questão obtendo a representação de um sinal comprimido sem precisar adquirir N amostras (DONOHO, 2006).

Informação é um conjunto de símbolos físicos – palavras, pontos e traços, entre outros – que carregam significado para as partes envolvidas (HARTLEY, 1928). No caso de amostragem compressiva, os símbolos que carregam significado estão concentrados em poucos coeficientes.

De acordo com Donoho (2006), a principal diferença para a compressão digital é que a AC tem como objetivo amostrar diretamente o sinal com uma taxa de amostragem muito menor do que indica o Teorema de Nyquist, adquirindo somente as amostras mínimas necessárias para representar a informação de interesse. Em outras palavras, a AC comprime o sinal enquanto amostra, não depois da amostragem.

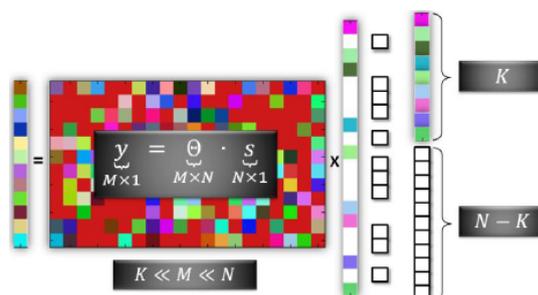
2.1.2 Medidas e Matrizes na Amostragem Compressiva

Considere um processo de medição linear que determina $M < N$ produtos internos entre x a coleção de vetores $\{\phi_j\}_{j=1}^M$, ou seja, $y_j = \langle x, \phi_j \rangle$. Organize as medidas y_j em vetores y de tamanho $M \times 1$ e os vetores de medição ϕ_j^T como colunas de uma matriz ϕ de tamanho $M \times N$ (BARANIUK, 2007). Assim, o vetor y pode ser escrito como mostra a primeira igualdade da Equação 2. Substituindo x da Equação 1, o vetor y pode ser representado pela segunda igualdade da Equação 2, onde $\theta = \phi\psi$.

$$y = \phi x = \phi \psi s = \theta s \quad (2)$$

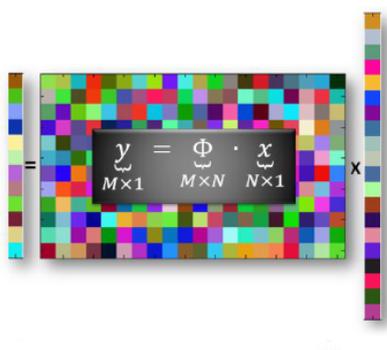
A matriz ϕ é comumente chamada de matriz de medição (ou matriz de medidas) e a matriz ψ é chamada de matriz de transformação (ou matriz de esparsidade). A Figura 4 mostra o processo da tomada de medidas no domínio não esparsa em forma de matriz, a Figura 5 mostra o processo da retirada de medidas no domínio esparsa também em forma de matriz, e na Figura 6 é possível ver a representação da igualdade central da Equação 2.

Figura 4: Medidas no domínio não esparsas



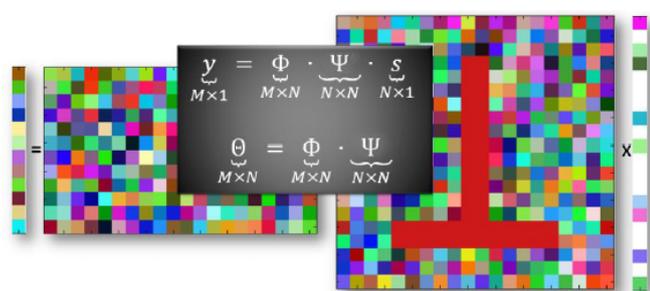
Fonte: (REIS, 2017).

Figura 5: Medidas no domínio esparsas



Fonte: (REIS, 2017).

Figura 6: Matrizes de medidas e de transformação



Fonte: Adaptado de (REIS, 2017).

2.1.3 Incoerência

A escolha da matriz de medidas é extremamente importante para garantir que a recuperação do sinal x seja feita corretamente. A matriz de medidas ϕ deve permitir a reconstrução do sinal x de tamanho N a partir do vetor de medidas y com $M < N$ (FOUCART; RAUHUT, 2013). A matriz de medidas também deve obedecer à algumas condições para que seja possível reconstruir o sinal após a amostragem compressiva, e o principal critério a ser obedecido é a incoerência entre as matrizes ϕ e ψ .

Coerência (ou baixa incoerência) pode ser definida como a máxima correlação entre quaisquer dois elementos de ϕ e ψ , ou seja, o maior valor dos produtos internos de dois elementos das matrizes ϕ e ψ , como mostra a Equação 3.

$$\mu(\phi, \psi) = \sqrt{N} \max\{\phi^T \psi\} \quad (3)$$

Se ϕ e ψ contêm elementos correlacionados, diz-se que a coerência é alta, caso o contrário, a coerência é baixa, variando entre 1 e \sqrt{N} . Em outras palavras, quando $\mu(\phi, \psi) = 1$ a coerência entre as matrizes é mínima (ou a incoerência é máxima), e quando $\mu(\phi, \psi) = \sqrt{N}$ a coerência é máxima, ou seja, as bases são minimamente incoerentes (CANDES; WAKIN, 2008).

A amostragem compressiva baseou-se no princípio de que se ψ for uma matriz de transformação e possuir baixa coerência com ϕ , então a representação do sinal em ϕ não será esparsa. Ou seja, a informação contida em x será uniformemente espalhada por todos os coeficientes do vetor de medidas y , assim cada coeficiente terá parte da informação global do sinal (MEDEIROS *et al.*, 2010).

Pode-se concluir então que é possível reconstruir a informação completa do sinal x amostrando-se somente $M < N$ coeficientes, pois existe redundância no sinal x (ele precisa ser esparsa em alguma base), e a representação do sinal em ϕ resulta em coeficientes que contém a mesma quantidade de informação. Assim, a incoerência entre as matrizes de medidas e de transformação é uma condição necessária para o sucesso da amostragem compressiva (MEDEIROS *et al.*, 2010; SILVA, 2019).

2.1.4 Reconstrução

O sinal original pode ser reconstruído a partir das medidas obtidas pela amostragem compressiva através da solução de um problema de otimização. O algoritmo de reconstrução do sinal precisa das medidas M do vetor y , da matriz de medidas ϕ , e da matriz de transformação ψ para reconstruir o sinal x de tamanho N , ou a sua representação esparsa s . Entretanto, para sinais K -esparsos, existem infinitas soluções que satisfazem $\theta s = y$ pois $M < N$ na Equação 2. Isso acontece pois se $\theta s = y$ então $\theta(s + r) = y$ para qualquer vetor r no espaço nulo de θ , $\ker(\theta) = \{r \in R^N : r = 0\}$ (BARANIUK, 2007; SILVA, 2019).

A reconstrução do sinal x pode ser feita através da minimização das normas l_0 , l_2 e l_1 , mas para tal é importante primeiro saber o conceito de norma.

Se u e v são vetores no espaço vetorial S , u será uma norma somente se satisfizer as seguintes propriedades (MOON; STIRLING, 2000):

- $\|u\| \geq 0$ para qualquer $u \in S$.
- $\|u\| = 0$ se e somente se $u = 0$.
- $\|\lambda u\| = |\lambda| \times \|u\|$, sendo λ um escalar arbitrário.
- $\|u + v\| \leq \|u\| + \|v\|$, chamada de desigualdade triangular.

Deste modo, pode-se definir uma variedade de normas, tais como (REIS, 2017):

- **Norma l_1 :** representa a soma do módulo dos valores de u .

$$\|u\|_1 = \sum_{i=1}^n |u_i| \quad (4)$$

- **Norma l_2 :** é o tamanho do vetor, chamada de norma Euclidiana.

$$\|u\|_2 = \left(\sum_{i=1}^n |u_i|^2 \right)^{\frac{1}{2}} \quad (5)$$

- **Norma l_p :** é a forma genérica de uma norma de ordem p , sendo $0 \leq p \leq \infty$. As

normas l_2 e l_1 são casos específicos para $p = 2$ e $p = 1$, respectivamente.

$$\|u\|_p = \left(\sum_{i=1}^n |u_i|^p \right)^{\frac{1}{p}} \quad (6)$$

- **Norma l_∞ :** é o valor máximo em módulo dos elementos de u .

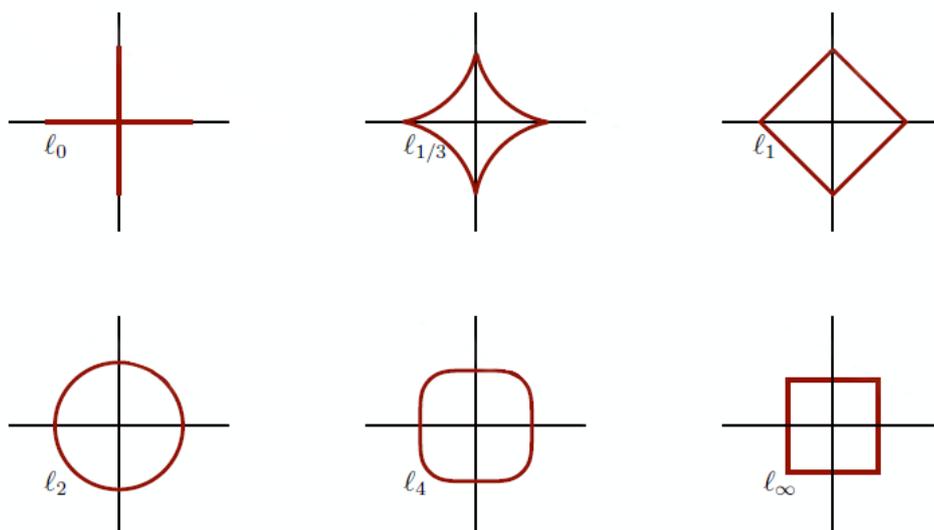
$$\|u\|_0 = \max_{i=1,2,\dots,N} |u_i| \quad (7)$$

- **Norma l_0 :** representa a quantidade de valores não nulos de u , também dita ser a sua cardinalidade. Por não satisfazer a propriedade desigualdade triangular, a norma l_0 é chamada de quase-norma.

$$\|u\|_1 = \#\{i : u_i \neq 0\} \quad (8)$$

Cada uma das normas descritas possui a sua geometria. A Figura 7 apresenta as formas geométricas de algumas normas.

Figura 7: Representação geométrica de várias normas l_p



Fonte: Adaptado de (BRUNTON; KUTZ, 2017).

Para problemas inversos deste tipo, a utilização da norma l_2 é uma escolha adequada visando encontrar a solução de menor energia para $\theta s = y$, como mostra a Equação 9.

Esta solução também é chamada de método dos mínimos quadrados. Entretanto, a norma l_2 quase nunca retorna uma solução K -esparsa, retornando ao invés disso uma solução não esparsa \hat{s} com muitos elementos não nulos (BARANIUK, 2007).

$$\hat{s} = \min_{s \in R^n} \|s\|_{l_2} \text{ sujeito a } \theta s = y \quad (9)$$

A solução mais esparsa que satisfaz $\theta s = y$ é dada pela Equação 10.

$$\hat{s} = \min_{s \in R^n} \|s\|_{l_0} \text{ sujeito a } \theta s = y \quad (10)$$

Embora a solução encontrada pela norma l_0 seja a mais esparsa, a otimização feita na Equação 10 é do tipo NP-completo, e no geral, a solução só pode ser encontrada por meio de uma busca combinatória $\binom{N}{K}$. Todos os vetores possíveis K -esparsos em R^N devem ser verificados, e se o valor da esparsidade K não for conhecido a busca é ainda mais extensiva, o que a invalida como a melhor solução para este problema tendo em vista o alto custo computacional necessário para encontrar a solução desejada, mesmo para valores pequenos de N (BRUNTON; KUTZ, 2017; TROPP; GILBERT, 2007).

Entre os diversos métodos de reconstrução desenvolvidos, duas abordagens são bastante utilizadas, que são: a abordagem convexa, com destaque para a Busca de Base (BP, do inglês *Basis Pursuit*) (HE; QU, 2013), e a abordagem gulosa (ou míope), com destaque para a Busca por Correspondência Ortogonal (OMP, do inglês *Matching Pursuit*) (SATHYABAMA; SANKARI; NAYAGARA, 2013; DONG; WU, 2021) e a Amostragem Compressiva com Busca por Correspondência (CoSaMP, do inglês *Compressive Sampling Matching Pursuit*) (NEEDELL; TROPP, 2008).

2.1.4.1 Abordagem Convexa

Essa abordagem trata a reconstrução como um problema de otimização convexa. Tendo em vista que a minimização na pseudo-norma l_0 não é computacionalmente viável, a Busca por Base propõe uma alteração no método de busca: o algoritmo deve buscar a solução pela minimização da norma l_1 , como mostra a Equação 11, que apresenta uma solução esparsa o suficiente sem a dificuldade computacional de processamento (DONOHO, 2006; HE; QU, 2013).

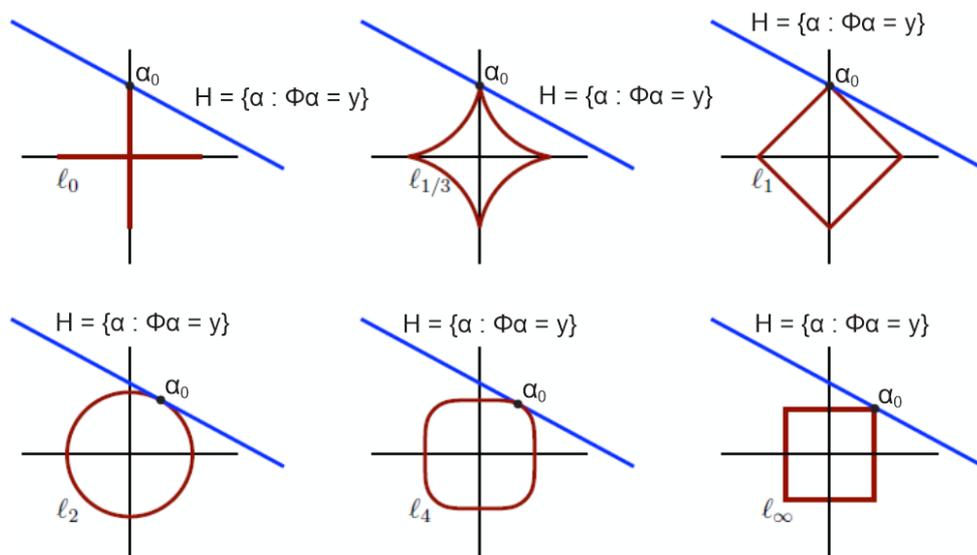
$$\hat{s} = \min_{s \in \mathbb{R}^n} \|s\|_{l_1} \text{ sujeito a } \theta s = y \quad (11)$$

A geometria da norma l_1 ajuda a entender porque esta é uma solução adequada para a amostragem compressiva, como mostra a Figura 8. A forma geométrica da norma l_1 é um diamante que tem suas pontas nos eixos de coordenadas, enquanto que a forma geométrica da norma l_2 é esférica. O ponto α_0 que toca no diamante é uma solução esparsa com apenas um componente não-nulo. A linha azul H na Figura 8 representa um conjunto com todas as soluções α que compartilham a mesma medida (ROMBERG, 2008).

A tarefa da minimização pela norma l_1 é encontrar o ponto exato em H que toca o diamante, e esse ponto pode ser calculado expandindo o diamante até que ele toque em H . Assim, basta verificar em qual eixo de coordenada a linha H tocará primeiro para determinar precisamente onde a solução esparsa s está localizada (ROMBERG, 2008; SILVA, 2019).

Comparando com o que acontece com a norma l_2 , nota-se que ao expandir a forma esférica, o primeiro ponto de interseção com H não ocorre nos eixos de coordenada, e, assim, não é esparsa.

Figura 8: Representação geométrica 2-D de várias normas l_p

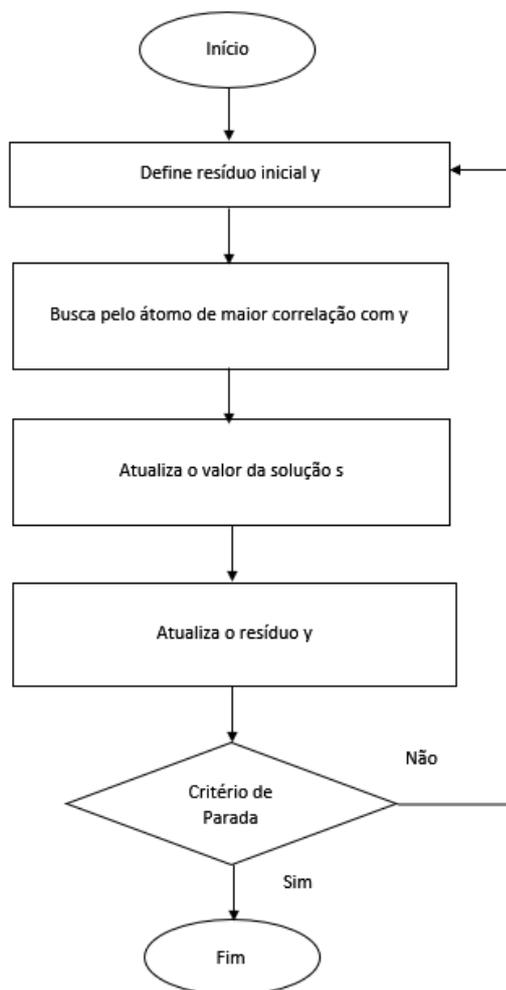


Fonte: Adaptado de (BRUNTON; KUTZ, 2017).

2.1.4.2 Abordagem Gulosa/Míope

Algoritmos gulosos (ou míopes) são algoritmos que buscam a solução global ótima a partir de valores ótimos locais baseada na ideia de dicionários atômicos correspondentes. Esses algoritmos são usados para obter a solução mais próxima do sinal original iterativamente, um etapa de cada vez (DONG; WU, 2021; BAYAR; BOUAYNAYA; SH-TERENBERG, 2014). A Figura 9 mostra as etapas de um algoritmo guloso.

Figura 9: Etapas de um algoritmo guloso



Fonte: Elaborado pelo Autor.

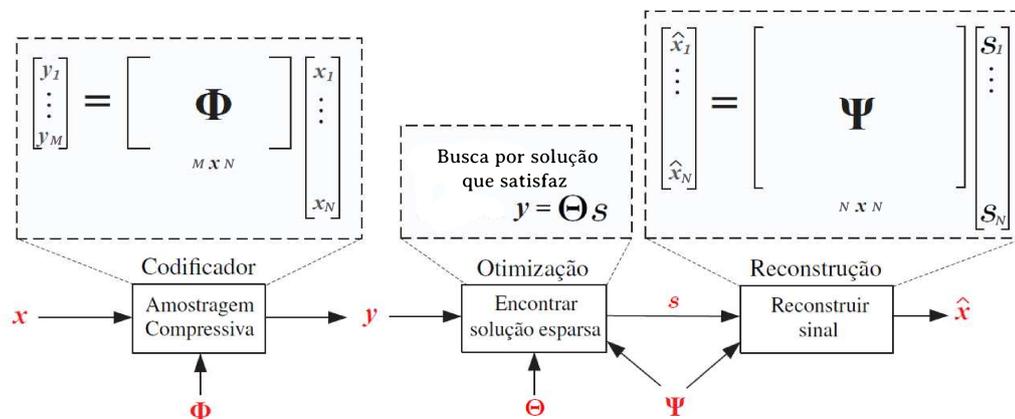
Primeiramente, define-se um conjunto vazio inicial que será atualizado por meio da seleção das colunas da matriz Θ que são altamente correlacionadas com as medidas de y . A cada iteração, o algoritmo seleciona a coluna de Θ , que é chamada de átomo, que melhora a solução de forma mais eficiente. A solução é, então, atualizada, e um novo resíduo é calculado. O procedimento é repetido até que um critério de parada seja

atingido (RANI; DHOK; DESHMUKH, 2018).

Dois algoritmos gulosos de destaque são o OMP e o CoSaMP. No OMP, a cada iteração um átomo é selecionado, e a solução é atualizada através do método dos mínimos quadrados, isto é, pela minimização da norma l_2 mostrada em 9 (RANI; DHOK; DESHMUKH, 2018). No caso do CoSaMP, a cada iteração o algoritmo seleciona múltiplas colunas de Θ que são maximamente correlacionadas com o vetor residual, por isso é chamado de algoritmo guloso paralelo (NEEDELL; TROPP, 2008).

A Figura 10 mostra o *framework* completo da amostragem compressiva. Em resumo, o sinal x com N amostras passa por um codificador onde, por meio da matriz de medidas ϕ de tamanho $M \times N$, são retiradas M medidas que concentram a informação do sinal (vetor y). A etapa seguinte é a de otimização, na qual procura-se a solução esparsa s que satisfaz $y = \Theta s$ - na figura isso é mostrado através da minimização da norma l_1 . Por fim, estima-se um sinal \hat{x} que é próximo de x .

Figura 10: Framework da amostragem compressiva



Fonte: Adaptado de (ABARI *et al.*, 2012).

2.2 Arquiteturas de AIC

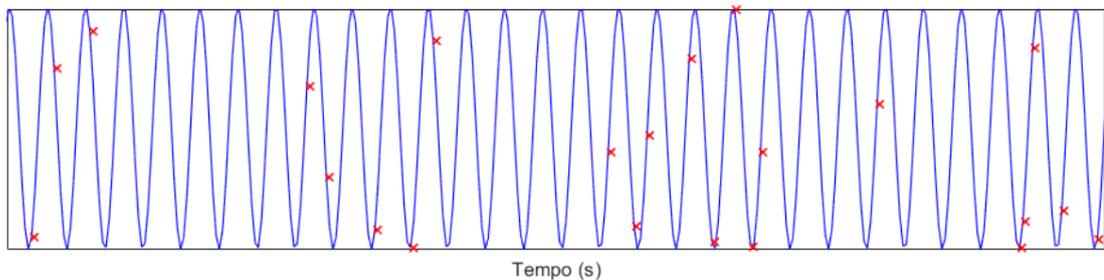
O AIC (conversor analógico para informação) é o dispositivo que implementa a Amostragem Compressiva. Neste tópico são apresentadas algumas arquiteturas de AIC.

2.2.1 Amostragem Aleatória

A amostragem aleatória (RS, do inglês *Random Sampling* consiste em retirar um número de medidas M aleatoriamente espaçadas de um sinal x que, em média, é menor que a quantidade gerada pela amostragem de Nyquist, e que são suficientes para reconstruir todo o sinal de interesse, devido a esparsidade do sinal (MANGIA *et al.*, 2018).

A Figura 11 mostra um sinal cossenoidal x com tamanho N , no tempo. Seguindo o Teorema de Nyquist, o sinal precisa ser amostrado com uma taxa de amostragem de pelo menos $f_s = 2 \times f_m = 2 \times 1/T_s$, sendo que f_m é a frequência máxima do sinal e T_s é o período de amostragem. São selecionadas M medidas aleatoriamente, representados na Figura 11 pelas marcações em vermelho. Neste caso, as amostras são adquiridas a uma frequência de amostragem média igual à $f_s \times (M/N)$ (SILVA, 2019).

Figura 11: Representação de um sinal cossenoidal no tempo.



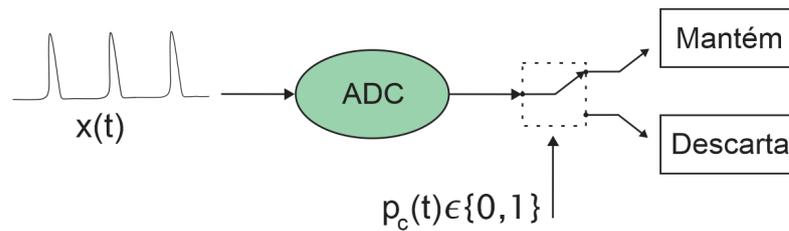
Fonte: Elaborado pelo autor.

Do ponto de vista de *hardware*, o RS é a arquitetura mais simples para se usar com a amostragem compressiva, sendo necessário apenas modular o *clock* de um ADC convencional para implementar. A matriz de medidas pode ser construída simplesmente considerando uma representação esparsa que contém um elemento igual a 1 em cada a linha, em todas as colunas correspondentes a uma posição na qual a amostragem ocorre, como mostra a Equação 12 (SILVA, 2019; MANGIA *et al.*, 2018).

$$\begin{cases} A_{i,j} \in \{0, 1\}, & \forall i, \forall j \\ \|A_{i,\cdot}\|_0 = 1 & \forall i \\ \|A_{\cdot,j}\|_0 \leq 1 & \forall j \end{cases} \quad (12)$$

Um tipo de AIC que adapta a amostragem aleatória é o Amostrador Não Uniforme (NUS, do inglês *Non-Uniform Sampler*) (SILVA *et al.*, 2017), que para a sua implementação precisa de um ADC, um gerador de sequências aleatórias e uma chave seletora, como mostra a Figura 12. O ADC retira as amostras do sinal, e uma chave seletora separa apenas algumas amostras, e descarta outras.

Figura 12: Diagrama de blocos do Amostrador Não Uniforme



Fonte: Elaborado pelo Autor.

No Amostrador Não Uniforme, o descarte dos dados representaria uma multiplicação por uma matriz semelhante a da Equação 14.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \quad (13)$$

2.2.2 Demodulador Aleatório

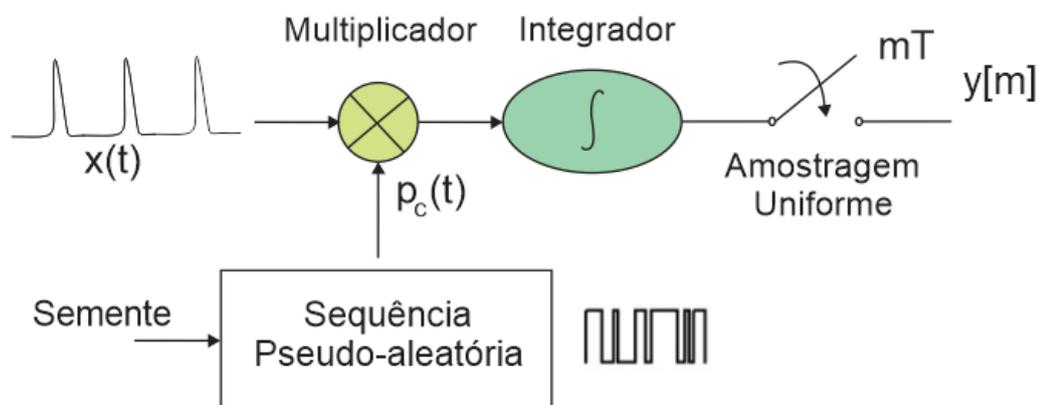
O Demodulador Aleatório (RD, do inglês *Random Demodulator*) foi proposto em (LASKA *et al.*, 2007) e consiste de um multiplicador, um integrador e um ADC.

O sinal de entrada $x(t)$ contínuo no tempo é multiplicado por uma sequência pseudo-aleatória de $+1$ e -1 , chamada de sequência de *chipping*, que pode ser gerada por um

registor de deslocamento com realimentação linear (LFSR, do inglês *Linear Feedback Shift Register*). Esse processo é equivalente à convolução no domínio da frequência e o resultado é o espalhamento da frequência do sinal que serve como uma assinatura única do sinal. O sinal resultante passa por um integrador (ou um filtro passa-baixas no lugar do integrador), e depois é amostrado pelo ADC. Após a retirada de cada amostra, o integrador é reiniciado (LASKA *et al.*, 2007; MASSOUD; SMAILI; SINGAL, 2011; KIROLOS *et al.*, 2006; RAGHEB *et al.*, 2008; TROPP *et al.*, 2009).

Embora o RD apresente baixa dificuldade de implementação, a falta de estruturas em paralelo diminui o seu desempenho, que é bem similar ao do RS (MANGIA *et al.*, 2018). A Figura 13 mostra o diagrama de blocos de um RD com um filtro passa-baixas e a Figura 14 mostra o comportamento dos sinais em cada etapa do RD.

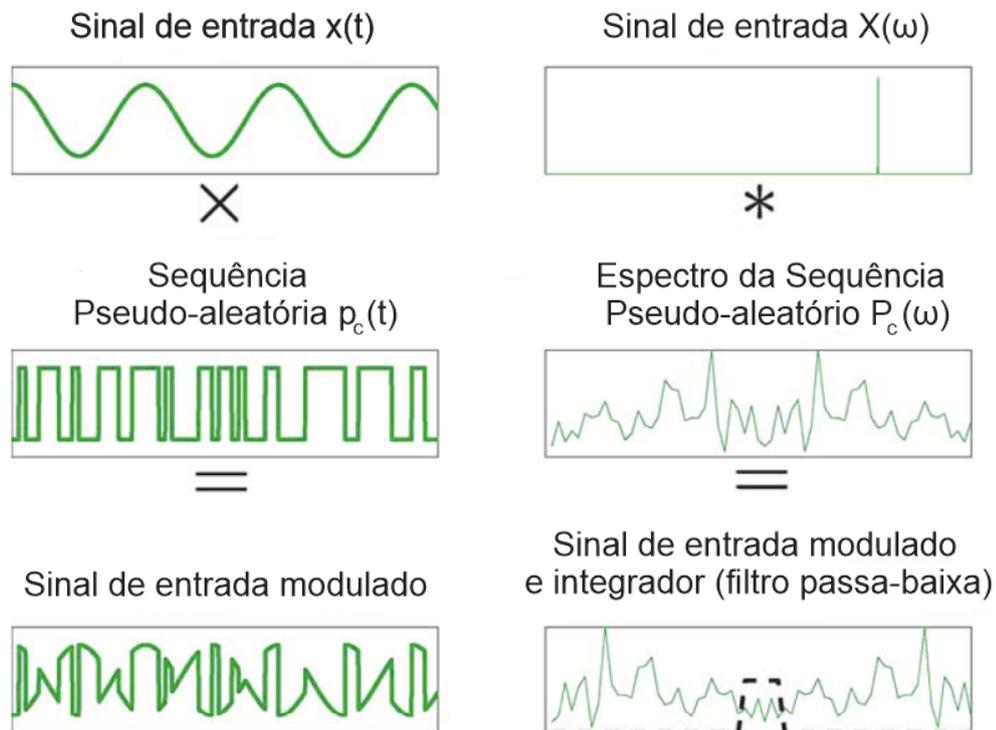
Figura 13: Diagrama de blocos do Demodulador Aleatório com filtro passa-baixas



Fonte: Elaborado pelo autor.

Para M e N , considerando R como a taxa N/M , cada medida y_j será o resultado do somatório de R produtos da matriz de medidas e do sinal de entrada. A multiplicação

Figura 14: Representação dos sinais em cada etapa do RD



Fonte: Adaptado de (TROPP; GILBERT, 2007).

pode ser vista como mostra Equação 14

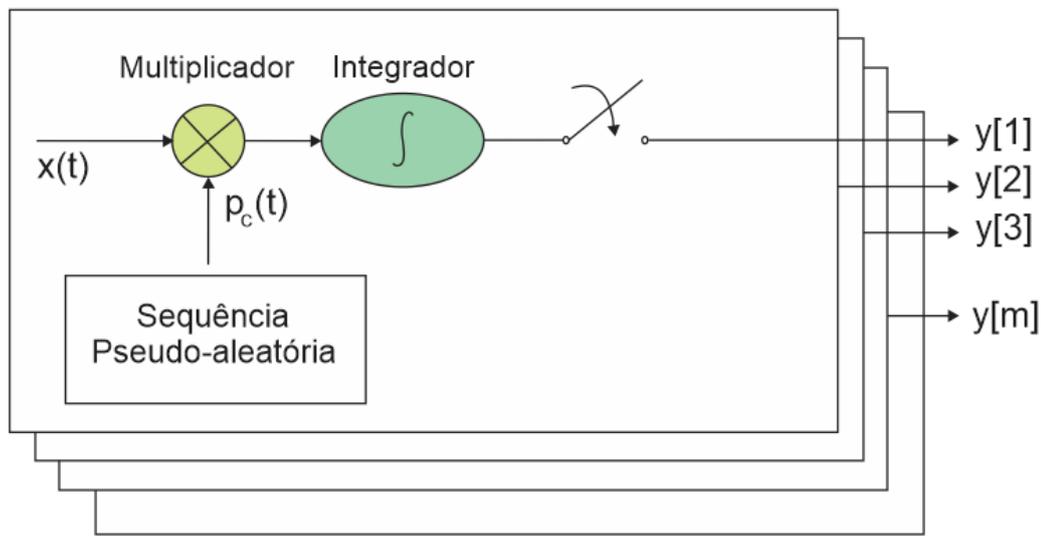
$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_M \end{bmatrix} = \begin{bmatrix} p_1 & \dots & p_R & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & p'_1 & \dots & p'_R & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & p''_1 & \dots & p''_R \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \quad (14)$$

2.2.3 Pré-Integrador de Modulação Aleatória

O Pré-Integrador de Modulação Aleatória (RMPI, do inglês *Random Modulator Pre-Integration*) foi proposto em (YOO *et al.*, 2012) e consiste em uma versão do RD que usa canais paralelos, como mostra a Figura 15. Cada canal contém um multiplicador e um integrador, e ao final de um período T_w , a saída y_i será a medida deste canal, sendo i o indicador do canal.

No RMPI, o sinal $x(t)$ é processado independentemente por M canais para calcular ao mesmo tempo os M produtos internos. No j -ésimo canal, $x(t)$ é multiplicado primeiro

Figura 15: Diagrama de blocos do RMPI



Fonte: Elaborado pelo Autor.

pela j -ésima função de sensoriamento, nesse caso, as sequências pseudo-aleatórias, e, em seguida, é integrado em T_w . Os y_j resultantes podem ser digitalizados por um ADC sub-Nyquist compartilhado, cuja taxa de amostragem é M/T_w conversões por unidade de tempo, o que é muito menor do que a taxa de Nyquist $1/T_w = N/T_w$, em que N são as amostras requeridas obedecendo ao critério de Nyquist e $M < N$ (PARESCHI *et al.*, 2015; SILVA, 2019).

Um sinal de entrada discreto no tempo em uma janela de tempo de tamanho $T_w = NT$ pode ser expresso por um vetor $x = \{x_1, x_2, \dots, x_N\} \in R^N$, onde x_i para $i = 1, 2, \dots, N$ representam as amostras da taxa de Nyquist equivalentes. As M medidas ainda são obtidas pelo cálculo do produto interno de x e as sequências pseudo-aleatórias $\phi_j \in R^{\{M \times N\}}$, como mostra a Equação 15, na qual $\phi_{j,i}x_i$ é o i -ésimo elemento do j -ésimo vetor (PARESCHI *et al.*, 2015; SILVA, 2019).

$$y_j = \langle \phi_j, x \rangle_N = \sum_{i=1}^N \phi_{j,i}x_i \quad (15)$$

Definindo a matriz de medidas $\phi \in R^{M \times N}$, cujas M linhas são os vetores ϕ_j , pode-se representar o vetor de medidas y por $y = \phi x$, como descrito na primeira igualdade da Equação 2, com j variando de 1 a M (PARESCHI *et al.*, 2015; SILVA, 2019). Na

implementação de [Reis et al. \(2016\)](#), cada linha da matriz de medidas ϕ é uma sequência pseudoaleatória gerada por um polinômio primitivo usando um LFSR.

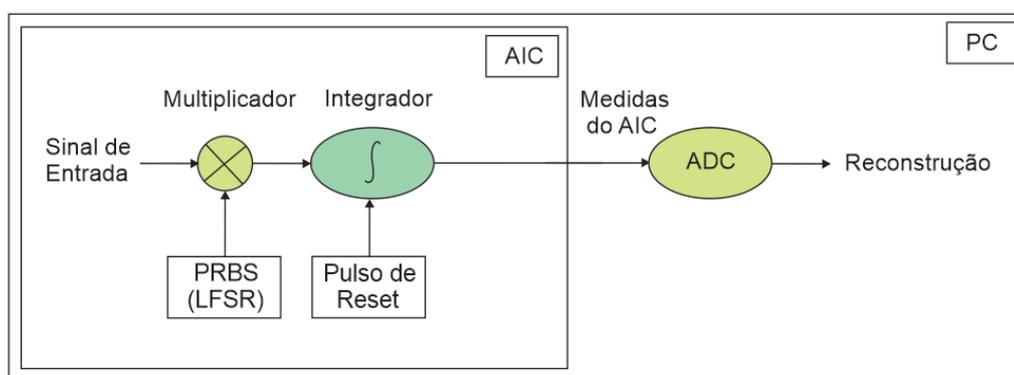
3 Metodologia da Pesquisa

Neste capítulo será apresentada a estrutura do AIC utilizado e os modelos de simulação.

3.1 Configuração Geral do Sistema

O AIC simulado é composto por um multiplicador, um LFSR, um filtro passa-baixas, e, ao final, é conectado a um ADC Sub-Nyquist, como é descrito na Figura 16.

Figura 16: Diagrama de blocos do sistema simulado



Fonte: Elaborado pelo Autor.

O PC foi utilizado para desenvolver as simulações do AIC e também para a reconstrução do sinal após a amostragem compressiva. Os softwares utilizados para essas finalidades são descritos a seguir.

- **Proteus:** É um software voltado para o design e simulação de circuitos eletrônicos. Foi criado pela *Labcenter Electronics*, e possui um ambiente bastante intuitivo para design de PCs. Foi utilizado para simulações do AIC até a etapa antes do ADC.
- **MATLAB:** É uma plataforma voltada para a análise de dados, desenvolvimento de algoritmos e para a criação de modelos e aplicações. Pertence à empresa *MathWorks*, e possui a sua própria linguagem de programação. Foi utilizada para simulações e, principalmente, para a reconstrução do sinal.
- **Simulink:** É uma ferramenta de diagramas de blocos voltada para a modelagem, análise e desenvolvimento de sistemas dinâmicos, e pode ser acessada pelo MATLAB.

Foi utilizado para as simulações do AIC completo, tendo como resultado as medidas comprimidas.

3.1.1 AIC no Proteus

A partir do que mostra a Figura 16, o AIC é constituído por: Gerador de sequências pseudo-aleatórias (LFSR), Circuito condicionador de sinal, Multiplicador, Filtro passa-baixas (integrador). A escolha do Proteus foi feita para aproximar o AIC simulado do comportamento em *hardware*.

O design feito no Proteus simula até o filtro passa-baixas, e o sinal gerado foi comparado com o sinal obtido no Simulink. As comparações serão comentadas na Seção de Simulações e Resultados.

3.1.1.1 Gerador de Sequências Pseudo-aleatórias

O Gerador de sequências pseudo-aleatórias é um *hardware* ou *software*, que gera uma sequência de números binários pseudo-aleatórios. Quando feito em *hardware*, pode-se utilizar um registrador de deslocamento de realimentação linear (LFSR) na implementação (REIS, 2017).

O circuito de um LFSR consiste em um conjunto de *flip-flops* tipo D (ou T), cuja as saídas de algum *flip-flops* realimentam o primeiro *flip-flop* com uma soma módulo 2 por meio de portas Ou-exclusivo (XOR, do inglês *Exclusive-OR*) ou Não Ou-exclusivo (XNOR, do inglês *Not Exclusive-OR*). Os *flip-flops* "ativos" são chamados de *taps*. Os *taps* determinam o polinômio característico de módulo 2 que representa o LFSR, e que é expresso na forma descrita pela Equação 16, sendo $a_0 = 1$ e $a_1 = 1$ para os *taps*.

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-2}x^{n-2} + a_{n-1}x^{n-1} + a_nx^n \quad (16)$$

Para um LFSR de grau 6 com *taps* (2, 3), por exemplo, o seu polinômio característico é $p(x) = 1 + 0x + 0x^2 + 1x^3 + 1x^4$, ou $p(x) = 1 + x^3 + x^4$. Se o primeiro *flip-flop* for numerado como 1 e não como 0, os *taps* são numerados de acordo com os graus do polinômio primitivo, ou seja, o polinômio (2, 3) passa a ser $p(x) = 1 + 0x + 1x^2 + 2x^3$, ou $p(x) = 1 + x^2 + x^3$.

Os *taps* se baseiam na teoria dos Campos de Galois (GF, do inglês *Galois Field*, que são

campos finitos com $GF(2^m)$ elementos, sendo m o número de bits. Um campo binário com m bits pode assumir 2^m bits valores únicos, mas um LFSR de tamanho máximo (polinômio primitivo) com m bits tem somente $(2^m - 1)$ valores. Isso ocorre porque um LFSR com portas XOR não vai percorrer o valor em que todos os bits são 0, enquanto que um LFSR com portas XNOR não vai percorrer o valor com todos os bits iguais a 1 (MAXFIELD, 2018). A Tabela 1 apresenta os *taps* para os polinômios primitivos de 2 a 10 *flip-flops* e o tamanho máximo de cada sequência.

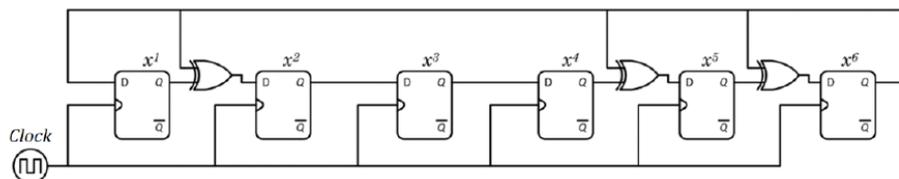
Tabela 1: *Taps* para LFSRs com 2 a 10 *flip-flops*

# de Flip-Flops	Tamanho da Sequência	Taps
2	3	[0,1]
3	7	[0,2]
4	15	[0,3]
5	31	[1,4]
6	63	[0,5]
7	127	[0,6]
8	255	[1,2,3,7]
9	511	[3,8]
10	1023	[2,39]

Fonte: Adaptado de (MAXFIELD, 2018).

O LFSR pode ser classificado pelo modo que a realimentação é feita, podendo ser de Galois (*One-to-many*) ou de Fibonacci (*Many-to-one*), como mostram as Figuras 17 e 18, respectivamente.

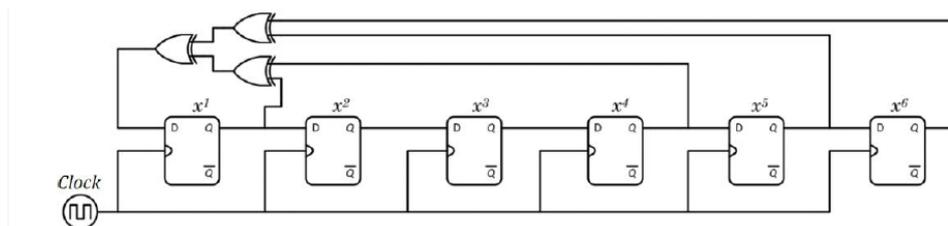
Figura 17: Arquitetura de Galois para o polinômio $1 + x + x^4 + X^5 + X^6$



Fonte: (REIS, 2017).

Neste trabalho, o polinômio primitivo adotado para as simulações no Proteus foi $1 + x^2 + x^3 + x^4 + X^8$, pois a sequência gerada resulta em uma matriz de medição incoerente. O design do bloco LFSR no Proteus pode ser visto na Figura 19, que é composto pelo

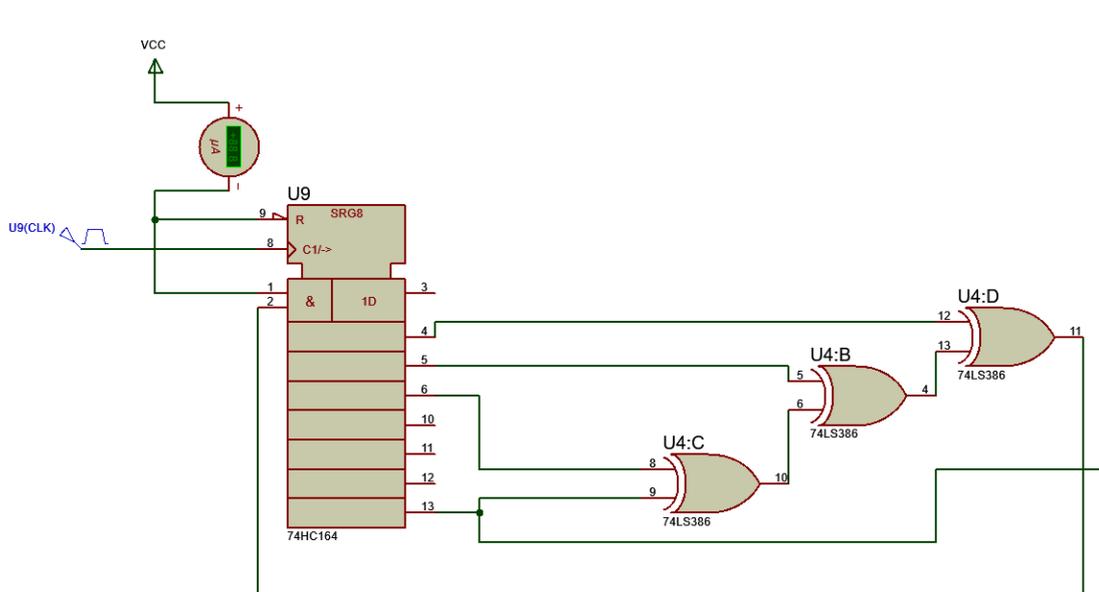
Figura 18: Arquitetura de Fibonacci para o polinômio $1 + x + x^4 + X^5 + X^6$



Fonte: (REIS, 2017).

registrador de deslocamento 74HC164 e três portas XOR, e que o sinal na entrada 8 do registrador é o sinal do *clock*.

Figura 19: Design do LFSR no Proteus



Fonte: Elaborado pelo autor.

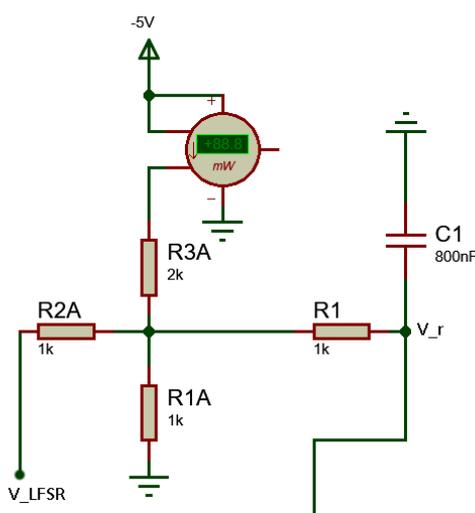
3.1.1.2 Circuito condicionador de sinal

O Circuito condicionador de sinal é um divisor resistivo que recebe o sinal gerado pelo LFSR, que possui os valores de tensão +5V no nível alto e 0V no nível baixo, e faz uma atenuação e um deslocamento para +1V e -1V, respectivamente, como propôs REIS (2017). Isso é necessário para que, quando o sinal do LFSR for multiplicado pelo sinal de entrada, parte deste sinal não seja perdido na multiplicação pelas parcelas iguais a 0V.

A Equação 17 representa a operação que o circuito resistivo faz e a Figura 20 mostra o circuito feito no Proteus.

$$V_r = \frac{V_{LFSR}}{2,5} - 1 \quad (17)$$

Figura 20: Circuito condicionador de sinal no Proteus



Fonte: Elaborado pelo autor.

3.1.1.3 Multiplicador

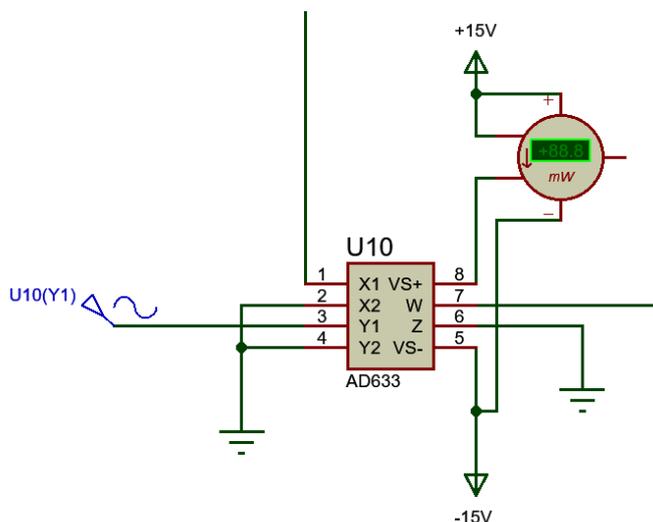
O multiplicador é responsável pela multiplicação do sinal de entrada com uma sequência LFSR. Nas simulações foi utilizado o CI AD633 como propôs REIS (2017), que é um multiplicador de 4 quadrantes amplamente utilizado em projetos eletrônicos, e que opera na faixa de tensão de $\pm 8V$ a $\pm 18V$. A saída do LFSR entra no pino X_1 do multiplicador, o sinal de entrada que será amostrado entra no pino Y_1 , e o resultado da multiplicação sai no pino W . A função de transferência do AD633 é mostrada na Equação 18, sendo X_1, X_2, Y_1 e Y_2 os pinos de entrada e Z um pino somador (DEVICES, 2015).

$$W = \frac{(X_1 - X_2)(Y_1 - Y_2)}{10V} + Z \quad (18)$$

3.1.1.4 Filtro Passa-baixas (Integrador)

O circuito do filtro passa-baixas utilizado no Proteus é mostrado na Figura 22, como

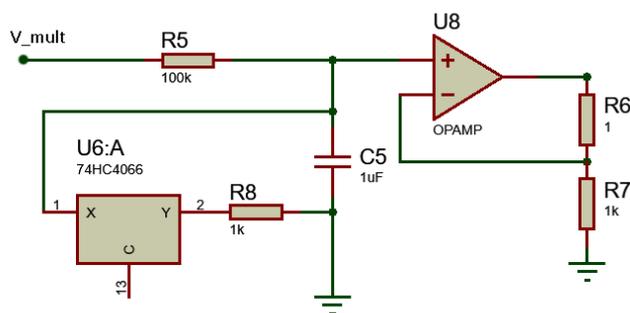
Figura 21: Multiplicador no Proteus



Fonte: Elaborado pelo autor.

proposto em (REIS, 2017). O reset do capacitor do filtro é feito pela chave analógica 74HC4066, que opera na faixa de tensão de +2V a +6V (INSTRUMENTS, 2016).

Figura 22: Filtro passa-baixas com chave analógica

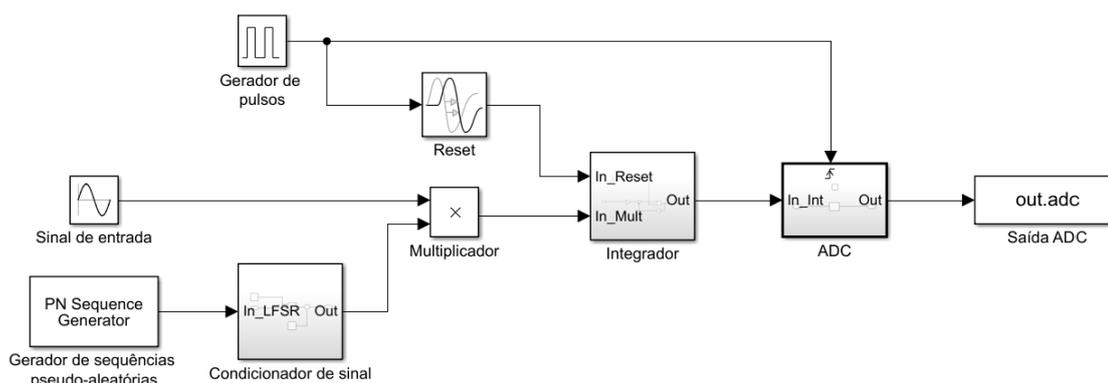


Fonte: Elaborado pelo autor.

3.1.2 AIC no Simulink

As simulações no Simulink foram feitas em um AIC que é uma versão adaptada de um RMPI, como mostra a Figura 23 e é constituído de: bloco gerador de sequências pseudo-aleatórias (LFSR), bloco condicionador de sinal, multiplicador, bloco integrador com reset, e ADC acionado por gatilho.

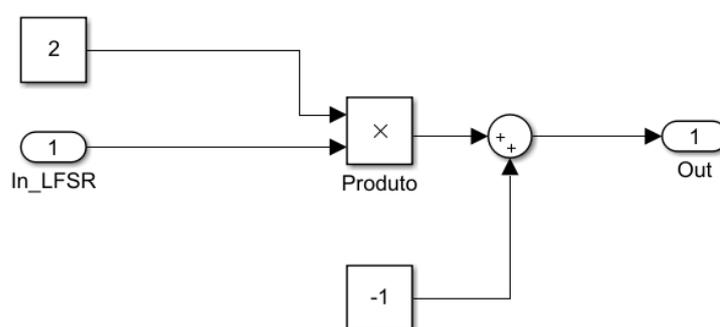
Figura 23: Diagrama de blocos do AIC no Simulink



Fonte: Elaborado pelo autor.

O bloco Gerador de seqüências pseudo-aleatórias simula um LFSR de Fibonacci, e tem como parâmetros o polinômio primitivo, o estado inicial, e a frequência do *clock*. No bloco Condicionador de sinal, apresentado na Figura 24, o sinal do LFSR, que é composto por valores lógicos 0 e 1, entra em um bloco Produto com duas entradas e é multiplicado por um sinal constante 2. O sinal resultante passa por um bloco de soma onde -1 é somado ao sinal, gerando valores escalares de -1 e $+1$, respectivamente.

Figura 24: Bloco Condicionador de sinal no Simulink



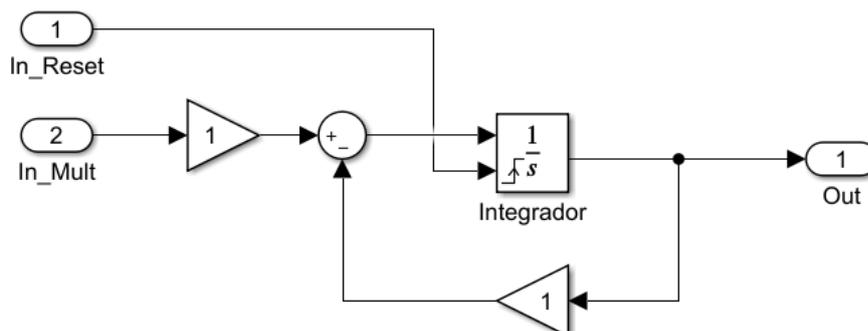
Fonte: Elaborado pelo autor.

Após passar pelo bloco condicionador de sinal, o sinal do LFSR e o sinal de entrada são multiplicados por um bloco Produto, e o resultado é enviado ao bloco integrador.

O bloco integrador tem duas entradas: a saída do multiplicador e o sinal de *reset*. A

configuração mostrada na Figura 25 possui um bloco somador e dois blocos de ganho. Nos trabalhos de (REIS, 2017; SILVA, 2019), os blocos de ganho correspondem aos valores de ganho e frequência de corte do filtro, mas para os testes deste trabalho eles foram adicionados para aproximar a integração do Simulink com a integração obtida no Proteus.

Figura 25: Bloco Integrador no Simulink



Fonte: Elaborado pelo autor.

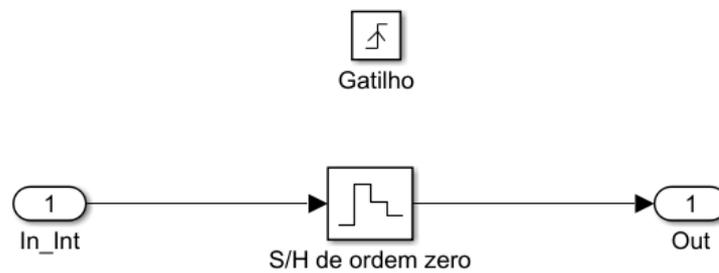
O bloco Gerador de pulsos gera um sinal que serve de gatilho para o ADC retirar uma medida. O mesmo pulso passa por um bloco *delay* que atrasa o sinal 0.05 segundos para reiniciar o integrador. Em outras palavras, após o ADC retirar uma medida, logo em seguida o integrador é zerado.

O bloco ADC mostrado na Figura 23 é um ADC de amostragem e retenção (S/H, do inglês *Sample-and-Hold*) de ordem zero (ZOH, do inglês *Zero-Order Hold*), ativado por um gatilho do tipo borda de subida (*rising*).

Como mencionado anteriormente, o AIC descrito neste trabalho é um AIC que possui estrutura similar à de um RD, mas as medidas são tomadas de forma similar ao que ocorre em um RMPI.

O resultado é então passado para o *Workspace* por meio do bloco Saída ADC (out.adc), e com esse resultado é possível fazer a reconstrução do sinal medido.

Figura 26: Bloco ADC no Simulink



Fonte: Elaborado pelo autor.

4 Resultados e Discussões

Nesta seção são apresentados resultados gerados após a implementação do AIC no Simulink, que se baseia no modo de funcionamento de um RMPI. Primeiramente é feito um comparativo dos resultados dos sinais do Proteus e Simulink para validar a configuração do AIC. Em seguida é feita a reconstrução de dois sinais medidos no domínio não esparso (tempo) no Simulink/Matlab utilizando três algoritmos de reconstrução: BP com CVX, OMP e CoSaMP. Por fim, o erro quadrático médio para os três algoritmos é calculado e avaliado.

4.1 Simulações no Simulink e Proteus

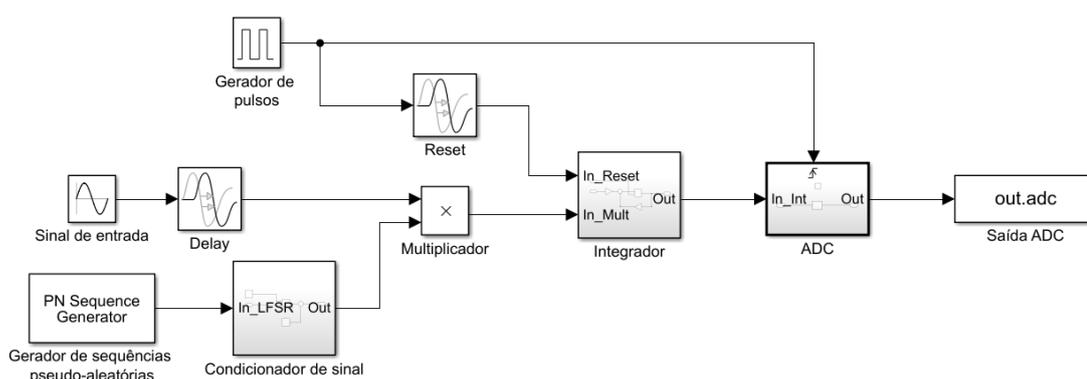
Como mencionado anteriormente, o polinômio primitivo adotado para as simulações desta seção foi $1+x^2+x^3+x^4+X^8$. A frequência do *clock* adotada foi 64Hz e foi escolhida de modo aleatório, o que influencia diretamente no tamanho da matriz de medidas. Nesse caso, $N = 64$. O valor de M para as comparações com o Simulink é $M = 1$.

Para as comparações das simulações, foi feita uma pequena adaptação no AIC do Simulink, como mostra a Figura 27. Um bloco com *delay* de 0.023 segundos foi adicionado logo após o sinal senoidal, tendo em vista que os sinais do LFSR do Simulink e do Proteus geram a mesma sequência, com exceção dos primeiros valores iguais 1 que o Proteus não gera. Isso se dá provavelmente em função da *seed* utilizada, que foi [00000001] no Simulink e no Proteus foi 0 V como valor inicial. O sinal de entrada com *delay* e o sinal

do LFSR são mostrados na Figura 28

A intenção por trás do bloco de *delay* é zerar os valores iniciais do sinal de entrada, para que no multiplicador os valores iniciais do LFSR sejam multiplicados por zero.

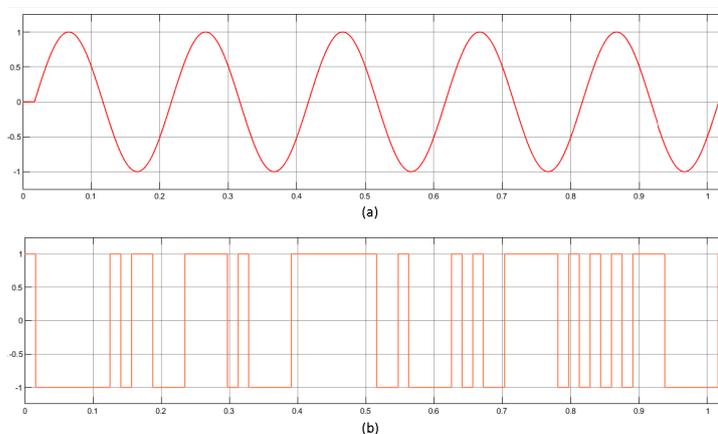
Figura 27: Diagrama de blocos do AIC no Simulink



Fonte: Elaborado pelo autor.

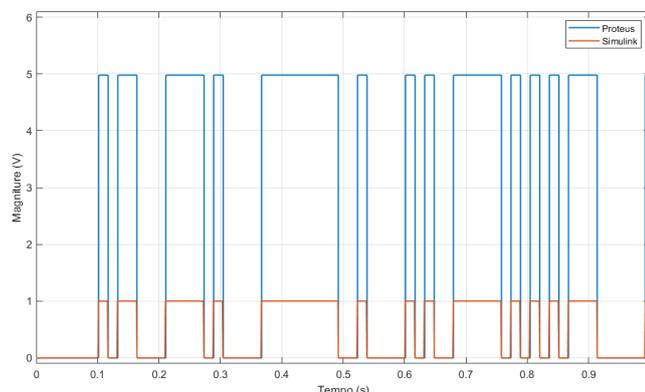
É importante frisar que esse *delay* foi adicionado apenas para fins comparativos, e não foi considerado no tópico de reconstrução de sinais. A Figura 29 mostra os gráficos do LFSR gerados pelo Proteus pelo Simulink, para uma janela de 1 segundo. Observa-se que as sequências têm as mesmas subidas e as mesmas descidas, nos mesmo pontos exatos, com as magnitudes originais mantidas para a melhor visualização dos gráficos.

Figura 28: Sinal senoidal com *delay* em (a); Sinal do LFSR em (b)



Fonte: Elaborado pelo autor.

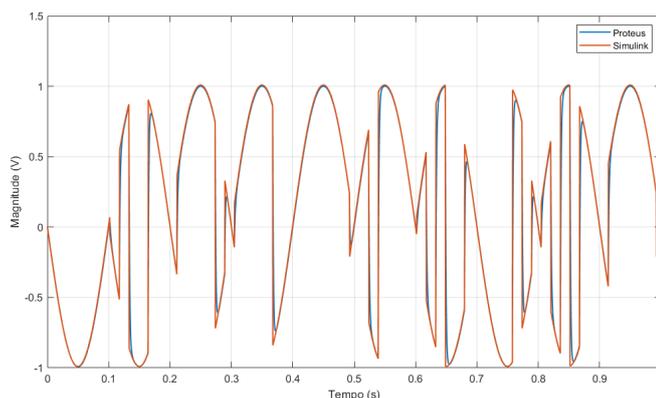
Figura 29: Sinais do LFSR do Proteus (em azul) e do Simulink (em laranja)



Fonte: Elaborado pelo autor.

Após o condicionamento de ambos os sinais do Proteus e Simulink, os valores de magnitudes são -1 V e 1 V . A Figura 30 mostra os sinais de saída do multiplicador de ambos os softwares em uma janela de 1 segundo. Observa-se que eles estão sobrepostos, mas o sinal em laranja apresenta picos mais rígidos, enquanto que o sinal em azul tem picos mais arredondados em função do comportamento analógico do multiplicador.

Figura 30: Sinais do multiplicador do Proteus (em azul) e do Simulink (em laranja)

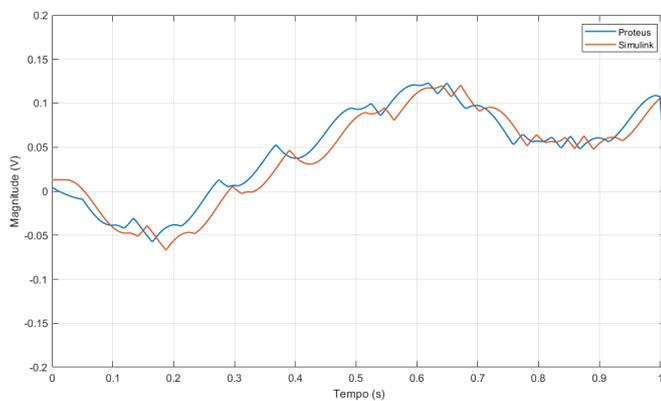


Fonte: Elaborado pelo autor.

Os sinais do integrador são mostrados na Figura 31 para um 1 segundo, e na Figura 32 para 5 segundos. Observa-se que as curvas não estão bem alinhadas como as etapas anteriores. Na 32 tem-se uma visão mais completa dos gráficos, e nota-se que as curvas são semelhantes, têm picos e vales parecidos, mas têm trechos divergentes, tal qual no

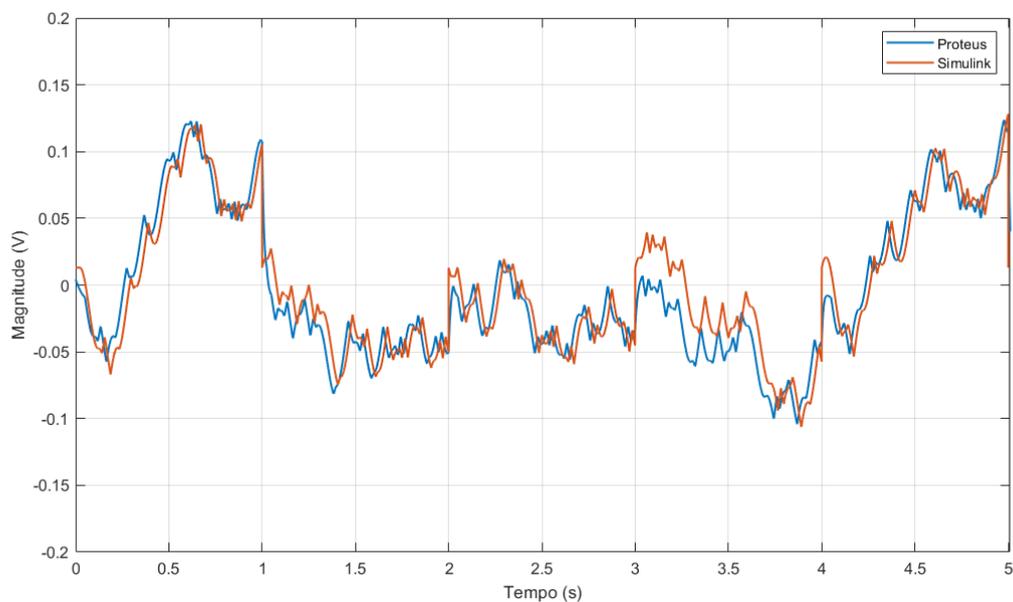
intervalo de 3,0 a 4,2 segundos.

Figura 31: Sinais do integrador do Proteus (em azul) e do Simulink (em laranja) para um 1 segundo



Fonte: Elaborado pelo autor.

Figura 32: Sinais do integrador do Proteus (em azul) e do Simulink (em laranja) para 5 segundos



Fonte: Elaborado pelo autor.

4.2 Reconstrução de um sinal de um tom

No primeiro ensaio apresentado foi utilizado um sinal de um tom, isto é, um sinal com uma senoide, com amplitude 1, frequência 20 Hz e fase 0. Três métodos de reconstrução foram utilizados: Busca de Base com CVX, Busca por Correspondência Ortogonal (OMP) e Amostragem Compressiva com Busca por Correspondência (CoSaMP).

A simulação possui 32 ciclos, cada ciclo com duração de 1 segundo, e ao final de cada ciclo uma medida é obtida. O tamanho do sinal de entrada é $N = 256$, então, a matriz de medidas ϕ gerada possui dimensões $M(32) \times N(256)$, onde cada linha corresponde à parte da sequência gerada pelo LFSR, representando o comportamento dos canais paralelos de um RMPI.

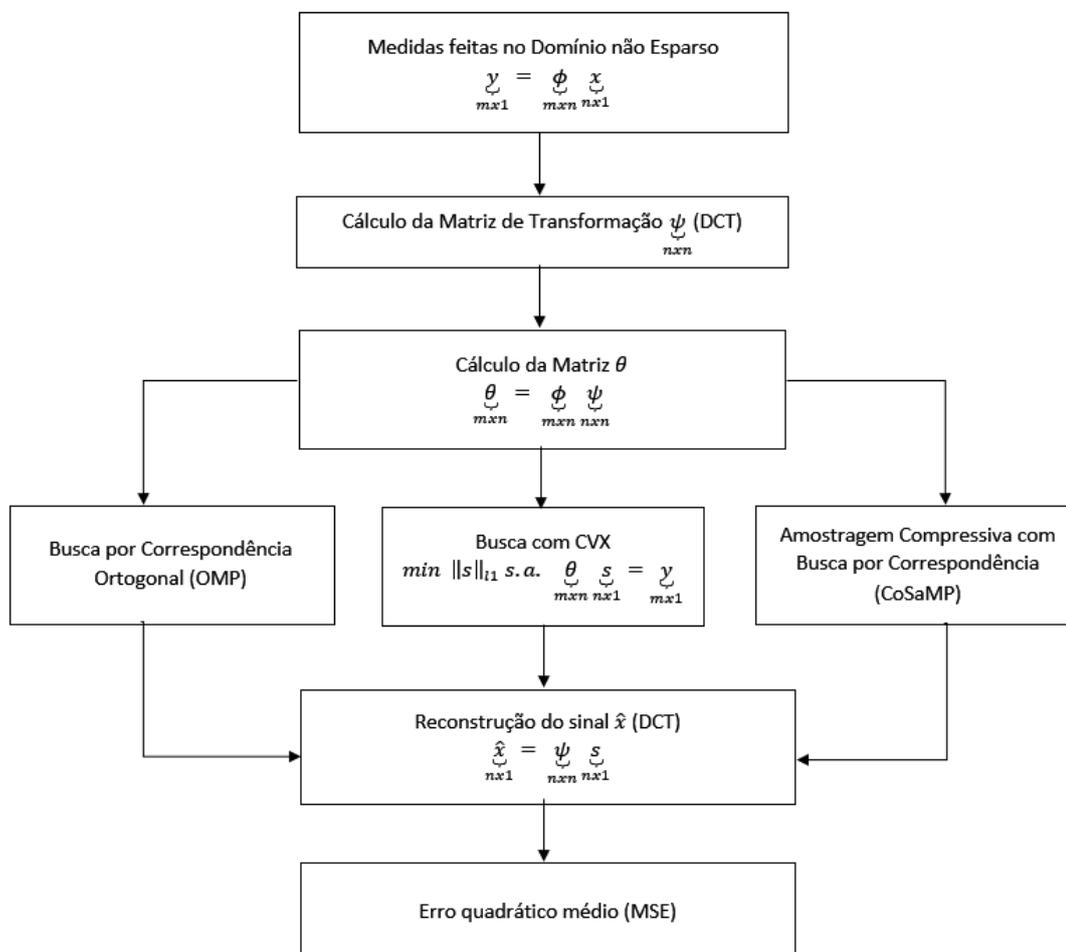
Um fluxograma das medidas feitas no domínio não esparso é mostrado na Figura 33. As medidas obtidas no Simulink ao final de cada ciclo são repassadas para o Matlab para compor o vetor de medidas y . Como o sinal utilizado é uma senoide, cujo domínio esparso é a frequência, a matriz de transformação foi construída a partir da transformada discreta de Fourier (DFT, do inglês *Discrete Fourier Transform*) através da ferramenta *dftmtx* do próprio Matlab, e possui tamanho $N(256) \times N(256)$. A matriz Θ é calculada por $\Theta = \Phi\Psi$, como mostrado na Equação 2, e possui dimensões $M(32) \times N(256)$.

O sinal reconstruído está no domínio esparso, então a transformada inversa de Fourier foi utilizada para retornar o sinal para o domínio não esparso. Nenhum ajuste ou refinamento foi feito nos sinais reconstruídos. A Figura 34 mostra, no domínio da frequência, os sinais: original e as reconstruções pelo CVX, OMP e CoSaMP, respectivamente.

Pode-se observar que os sinais reconstruídos não são exatamente iguais ao sinal original, mas apresentam picos de energia bem próximos de 19.9219Hz. Como mencionado na seção 2.1.4, o CVX não precisa informação de esparsidade do sinal para ser executado, e isso é representado pelo resíduos distribuídos em todo o espectro. Tanto o OMP quanto o CoSaMP necessitam da informação da esparsidade antes de serem executados, por isso os resultados encontrados possuem exatamente o mesmo valor de esparsidade do sinal original.

Observa-se também que as amplitudes dos sinais reconstruídos são muito menores que a do sinal original. Na Figura 35, nota-se a diferença do sinal original e dos sinais reconstruídos no domínio do tempo, com os CVX, OMP e CoSaMP, respectivamente. O

Figura 33: Fluxograma da reconstrução para medidas feitas no domínio não esparsos



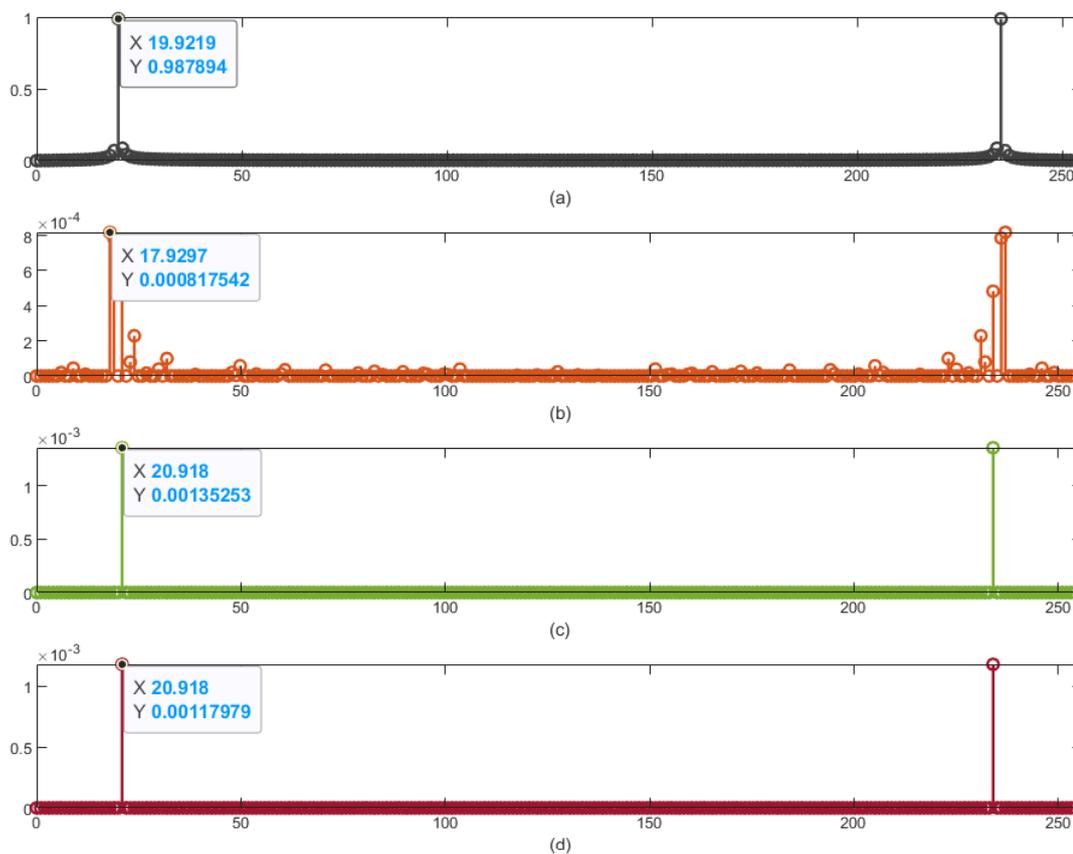
Fonte: Elaborado pelo autor.

o sinal reconstruído pelo CVX é o que mais diverge visualmente do sinal original, enquanto que os reconstruídos pelo OMP e pelo CoSaMP apresentam formas de onda bem similares a do sinal original.

Os sinais reconstruídos não são iguais ao sinal original, pois os picos de energia não se concentram na mesma frequência do sinal original. Os valores de erro quadrático médio (MSE) obtidos entre o sinal original e os reconstruídos são: 0,4981 para o CVX, 0,4983 para o OMP, e 0,4982 para o CoSaMP.

Ou seja, apesar de o sinal reconstruído pelo CVX apresentar resíduos espalhados pelo espectro, os três métodos de reconstrução apresentam valores de MSE bem similares.

Figura 34: Sinal de um tom original (a) e sinais reconstruídos com o CVX (b), OMP (c), e CoSaMP (d) no domínio da frequência



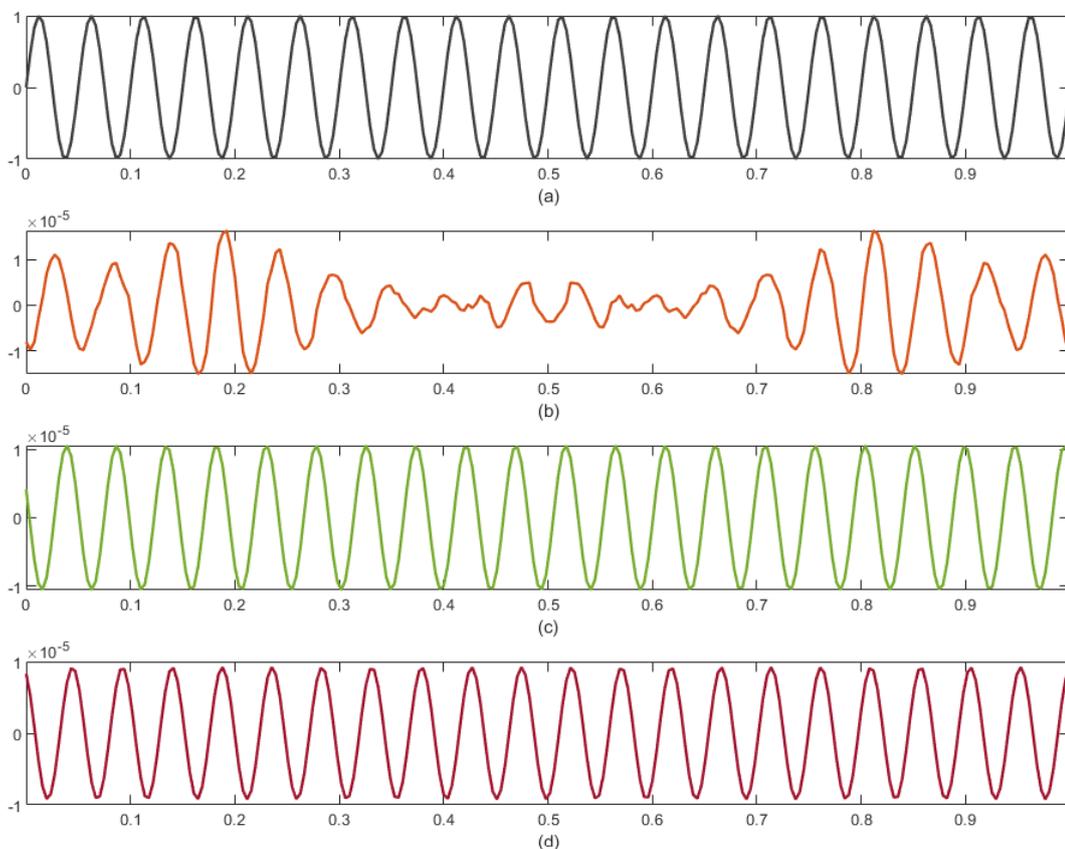
Fonte: Elaborado pelo autor.

4.3 Reconstrução de um sinal de dois tons

No segundo ensaio, foi utilizado um sinal de dois tons, isto é, um sinal com duas senoides, com frequências 15Hz e 50Hz, respectivamente, e ambas com amplitude 1 e fase 0. Também foram utilizados três métodos de reconstrução: Busca de Base com CVX, OMP e CoSaMP. Assim como no primeiro ensaio, os valores de frequência escolhidos foram aleatórios.

De forma similar a Subseção 4.2, a simulação é executada por M ciclos, cada ciclo com duração de 1, e ao final de cada ciclo uma medida é obtida. A matriz de medidas Φ gerada possui dimensões M (um determinado número de ciclos) \times N (tamanho do sinal de entrada), onde cada linha corresponde a parte da sequência gerada pelo LFSR, representando o comportamento dos canais paralelos de um RMPI. As medidas obtidas no Simulink ao final de cada ciclo são repassadas para o Matlab para compor o vetor de

Figura 35: Sinal de um tom original (a) e sinais reconstruídos com o CVX (b), OMP (c), e CoSaMP (d) no domínio do tempo



Fonte: Elaborado pelo autor.

medidas y , e a matriz de esparsidade utilizada foi a DFT, com dimensões $N \times N$.

Para a reconstrução do sinal de dois tons, 214 testes foram feitos, variando os valores de M , N , e o polinômio primitivo do LFSR. Os valores de M testados foram: 32, 48, 64, 128 (exclusivamente para polinômios de grau 9 e 10) e 256 (exclusivamente para os polinômios de grau 10). Os valores de N testados foram: 128, 192, 256, 512 (exclusivamente para polinômios de grau 9 e 10) e 1024 (exclusivamente para os polinômios de grau 10). E foram testados 22 polinômios primitivos, variando de graus 4 a 10. O Algoritmo [I](#) mostra a rotina utilizada para executar os testes.

O grau do polinômio indica a sequência máxima que ele é capaz de gerar, e dependendo do tamanho N definido, a sequência pseudo-aleatória gerada pelo LFSR pode se repetir após uma determinada quantidade de amostras. Por exemplo, um polinômio primitivo de grau 7 gera uma sequência de tamanho 128, e para um $N = 256$, a sequência se repetiria

Algoritmo 1: Rotina de testes com AIC adaptado

```

Entrada: Sinal de entrada  $x$ 
Saída: Sinais reconstruídos  $\hat{s}$ 
1 Definir tempo de simulação  $T$ 
2 Definir polinômio primitivo do LFSR
3 Definir  $N = [128, 192, 256, 512, 1024]$ 
4 Definir  $N_{max}$ 
5 Definir  $M = [32, 48, 64, 128, 256]$ 
6  $M_{max} \leftarrow T$ 
7 Gerar vetor  $y$  e tamanho máximo  $M_{max}$ 
8 para  $i$  de 1 a 5 faça
9   se  $N(i) \leq N_{max}$  então
10    para  $i$  de 1 a 5 faça
11     se  $M(i) \leq M_{max}$  então
12      Gerar  $\hat{s}$  com CVX
13      Gerar  $\hat{s}$  com OMP
14      Gerar  $\hat{s}$  com CoSaMP
15     fim
16    fim
17   fim
18 fim

```

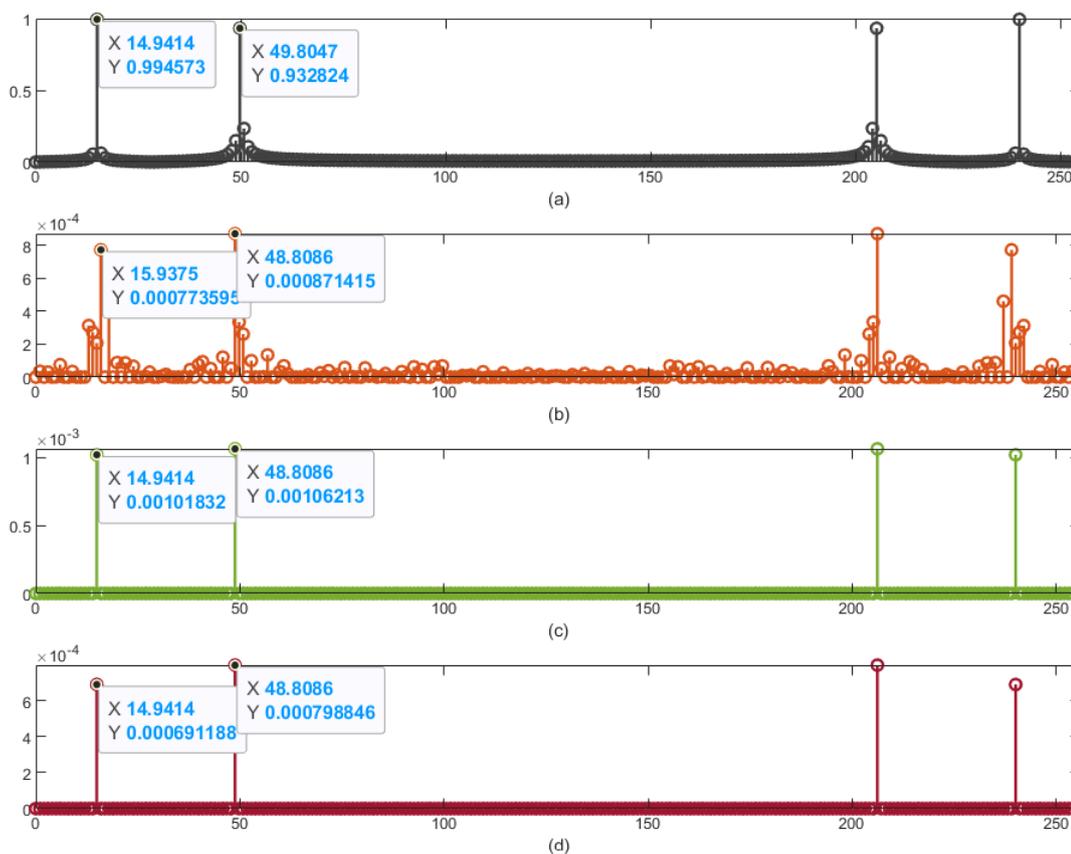
a partir do seu elemento 129.

Os sinais reconstruídos estão no domínio esparsos, então a DFT foi utilizada para retornar o sinal para o domínio não esparsos. A Figura 36 mostra, no domínio da frequência, o cenário de um dos testes feitos, com os sinais: original e as reconstruções pelo CVX, OMP e CoSaMP, respectivamente.

De forma similar ao sinal de um tom, os sinais de dois tons reconstruídos não são iguais ao sinal original, mas apresentam picos de energia bem próximos das frequências originais. O sinal reconstruído pelo CVX apresenta resíduos de frequência pelo espectro além dos picos centrais, pois como foi mencionado anteriormente, ele não necessita da informação de esparsidade para ser executado, enquanto que o OMP e o CoSaMP encontram sinais com a mesma quantidade de picos de energia do sinal original. As amplitudes dos picos dos sinais reconstruídos também são menores que as do sinal original, assim como no caso anterior.

Na Figura 37, nota-se a diferença do sinal original e dos sinais reconstruídos no domínio do tempo, com os CVX, OMP e CoSaMP, respectivamente. O sinal reconstruído pelo CVX é o que mais diverge visualmente do sinal original, enquanto que os reconstruídos

Figura 36: Sinal de dois tons original (a) e sinais reconstruídos com o CVX (b), OMP (c), e CoSaMP (d) no domínio da frequência



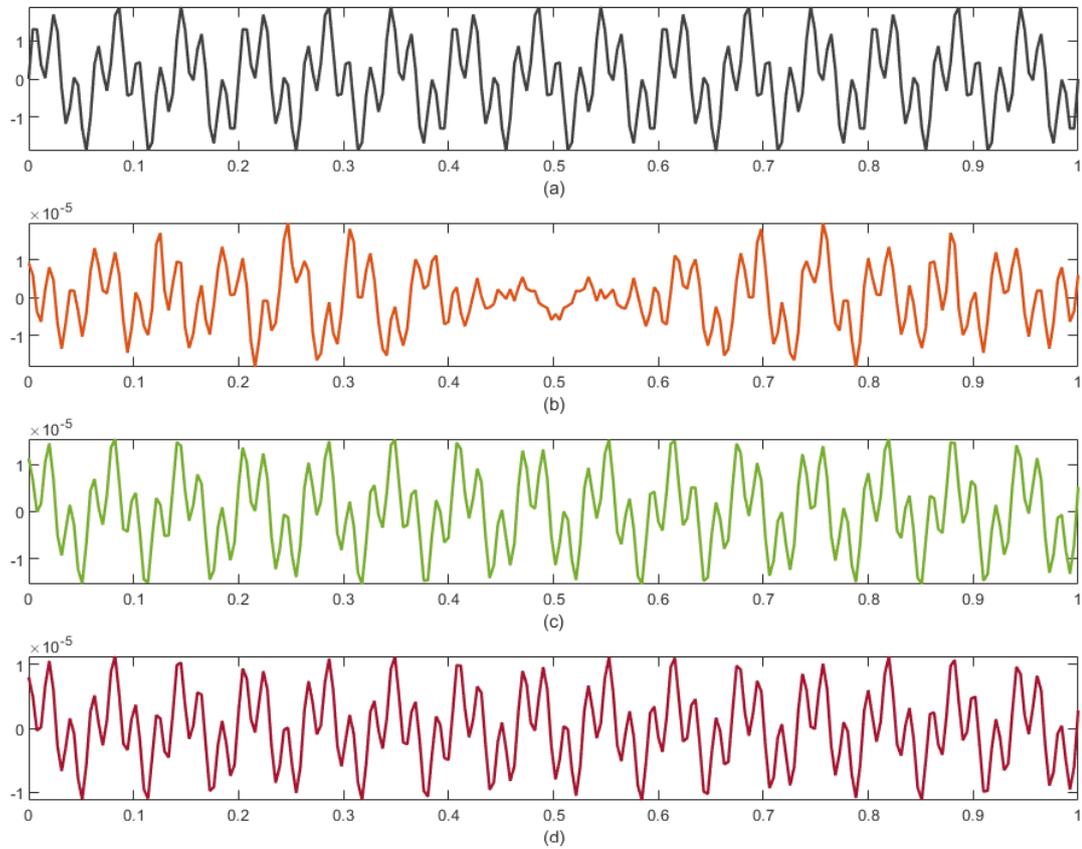
Fonte: Elaborado pelo autor.

pelo OMP e pelo CoSaMP apresentam formas de onda bem similares a do sinal original. As amplitudes, assim como no domínio da frequência, também são menores que a do sinal original.

Foi calculado o valor erro quadrático médio nos 214 testes feitos, e nenhum ajuste ou refinamento foi feito nos sinais reconstruídos. A Figura 38 mostra o percentual de testes com menor MSE para cada método de reconstrução utilizado. Os valores de MSE variaram de 0,9807 to 1,5516, com o CVX encontrando o menor valor de MSE em 55,1% dos testes. Vale destacar que em 23% dos testes, o CVX não conseguiu encontrar uma solução. Isso foi observado nos testes que o polinômio primitivo utilizado era de ordens 4, 5 e 6.

Em outras palavras, o CVX apresentou o melhor desempenho no que se refere a MSE calculada em 214 testes, mesmo sem utilizar a informação da esparsidade na reconstrução

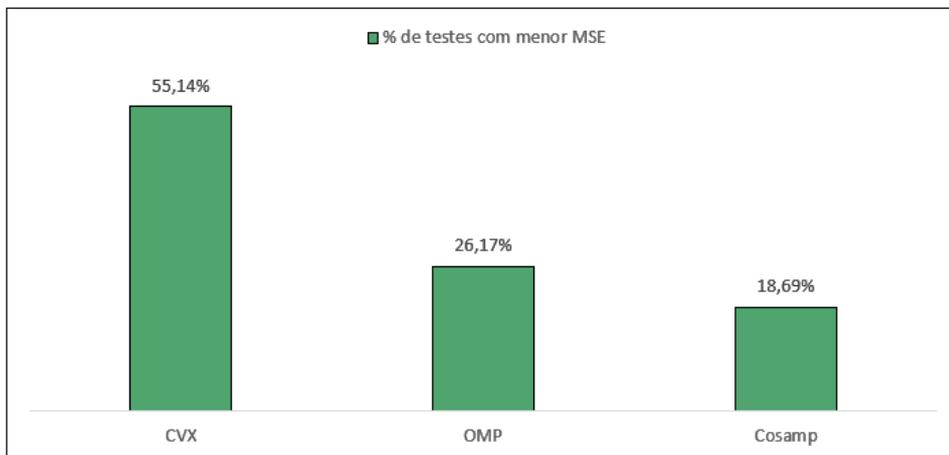
Figura 37: Sinal de dois tons original (a) e sinais reconstruídos com o CVX (b), OMP (c), e CoSaMP (d) no domínio do tempo



Fonte: Elaborado pelo autor.

e apresentar resíduos no sinal reconstruído.

Figura 38: Percentual de testes com menor MSE por método de reconstrução



Fonte: Elaborado pelo autor.

5 Conclusões

No contexto da Internet das Coisas (IoT), em que dispositivos geram dados sem nenhuma intervenção humana, e que implica em grandes quantidades de dados gerados, a técnica da Amostragem Compressiva foi proposta visando a aquisição de sinais sem a limitação de Nyquist. A Amostragem Compressiva explora a esparsidade que certas classes de sinais apresentam, sendo que um sinal pode ser considerado esparso quando é possível caracterizá-lo em poucos coeficientes não-nulos após uma transformação entre domínio.

O Conversor Analógico para Informação (AIC) implementa a Amostragem Compressiva e, ao passo que realiza a amostragem do sinal analógico de entrada, o sinal também é comprimido, e, assim, a saída obtida é uma versão digitalizada e comprimida do sinal de entrada. As saídas digitalizadas e comprimidas são utilizadas para a reconstrução do sinal por meio de algoritmos de reconstrução.

Neste trabalho, uma configuração de AIC com base no Pré-Integrador de Modulação Aleatória (RMPI) foi elaborada, de modo que embora possua estrutura similar a do Demodulador Aleatório (RD), o modo de operação é similar ao do RMPI. Para a configuração do AIC proposta, observou-se que os sinais gerados em cada etapa do AIC - LFSR, Multiplicador, Integrador/Filtro, e sinal de saída para o ADC - são similares nos softwares simulados, Proteus e Simulink.

As medidas geradas pelo AIC permitiram que os sinais senoidais testados fossem reconstruídos com componentes de frequências aproximados dos sinais originais, ainda que que alguns ajustes sejam necessários para que sejam mais próximos.

Foram simulados 214 testes foram simulados, e a métrica de avaliação de desempenho dos algoritmos de reconstrução utilizada foi o erro médio quadrático calculado. Os parâmetros do AIC variados nos testes foram os polinômios primitivos utilizados no LFSR e os tamanhos de M e N .

Os testes realizados indicam que o a Busca de Base com o CVX apresentou o menor valor do erro quadrático médio em 55.1% dos testes, ainda que em 23% dos testes não tenha conseguido encontrar uma solução adequada. É importante ressaltar que o CVX não necessita da informação da esparsidade para reconstruir um sinal, portanto o sinal reconstruído apresenta resíduos espalhados pelo espectro de frequência.

Ainda que os sinais reconstruídos pelo CVX apresentem resíduos, para a configuração

de AIC estudada neste trabalho o CVX foi o algoritmo que apresentou melhor desempenho no que se refere ao erro quadrático médio.

5.1 Trabalhos Futuros

Diante dos resultados obtidos, foram elencadas algumas atividades futuras para continuação das atividades deste trabalho de Mestrado visando futuras pesquisas:

1. Avaliar outros algoritmos de reconstrução e ajustar parâmetros relevantes do AIC;
2. Otimização dos algoritmos de reconstrução;
3. Inserção de ruídos nos sinais;
4. Avaliar o comportamento dos sinais com a variação das frequências de corte do filtro do AIC;
5. Implementar fisicamente o AIC;
6. Integrar o AIC com redes de sensores sem fio (RSSF);
7. Realização de testes experimentais.

5.2 Publicações

Ao longo deste trabalho, os resultados obtidos durante esta Dissertação de Mestrado resultaram nas seguintes publicações:

1. ARAUJO, Hugo Bruno Santos et al. A Performance Analysis of Reconstruction Algorithms for an Analog-to-Information Converter Architecture. INSCIT 2022 - 6th International Symposium on Instrumentation Systems, Circuits and Transducers, 2022. IEEE
2. ARAUJO, Hugo Bruno Santos et al. Amostragem Compressiva: Princípios Fundamentais e Aplicação com RMPI Adaptado. XXIV Congresso Brasileiro de Automática. 2022.

Referências

- ABARI, O. *et al.* Performance trade-offs and design limitations of analog-to-information converter front-ends. In: IEEE. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2012. p. 5309–5312.
- BARANIUK, R. G. Compressive sensing [lecture notes]. *IEEE signal processing magazine*, IEEE, v. 24, n. 4, p. 118–121, 2007.
- BAYAR, B.; BOUAYNAYA, N.; SHTERENBERG, R. Kernel reconstruction: An exact greedy algorithm for compressive sensing. In: *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. [S.l.: s.n.], 2014. p. 1390–1393.
- BLACKNELL, D. A comparison of compressive sensing and fourier reconstructions for radar target recognition. In: *2016 4th International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*. [S.l.: s.n.], 2016. p. 217–221.
- BRUNTON, S. L.; KUTZ, J. N. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. [S.l.]: Cambridge University Press, 2017.
- CANDÈS, E. J.; WAKIN, M. B. An introduction to compressive sampling. *IEEE signal processing magazine*, IEEE, v. 25, n. 2, p. 21–30, 2008.
- DEVICES, A. Ad633: Low cost analog multiplier. *Analog Devices*, 2015.
- DJELOUAT, H. *et al.* lot based compressive sensing for ecg monitoring. In: *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. [S.l.: s.n.], 2017. p. 183–189.
- DONG, J.; WU, L. Comparison and simulation study of the sparse representation matching pursuit algorithm and the orthogonal matching pursuit algorithm. In: *2021 International Conference on Wireless Communications and Smart Grid (ICWCSSG)*. [S.l.: s.n.], 2021. p. 317–320.
- DONOHUO, D. L. Compressed sensing. *IEEE Transactions on information theory*, IEEE, v. 52, n. 4, p. 1289–1306, 2006.
- FOUCART, S.; RAUHUT, H. *A mathematical introduction to compressive sensing*. [S.l.]: Springer, 2013.
- HARTLEY, R. V. Transmission of information 1. *Bell System technical journal*, Wiley Online Library, v. 7, n. 3, p. 535–563, 1928.
- HE, W.; QU, T. Audio lossless coding/decoding method using basis pursuit algorithm. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.: s.n.], 2013. p. 552–555.
- INSTRUMENTS, T. *SN74HC4066, Quadruple Bilateral Analog Switch (datasheet)*. 2016.

-
- KIROLOS, S. *et al.* Analog-to-information conversion via random demodulation. In: IEEE. *2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*. [S.l.], 2006. p. 71–74.
- LASKA, J. N. *et al.* Theory and implementation of an analog-to-information converter using random demodulation. In: IEEE. *2007 IEEE International Symposium on Circuits and Systems*. [S.l.], 2007. p. 1959–1962.
- LI, T.; ZHANG, Q. Compressive sensing sar image reconstruction based on a pseudorandom 2-d subsampling measurement matrix. In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. [S.l.: s.n.], 2016. p. 1094–1097.
- MALLAT, S. *A wavelet tour of signal processing*. [S.l.]: Elsevier, 1999.
- MANGIA, M. *et al.* *Adapted compressed sensing for effective hardware implementations: A design flow for signal-level optimization of compressed sensing stages*. [S.l.]: Springer, 2018.
- MASSOUD, Y.; SMAILI, S.; SINGAL, V. Efficient realization of random demodulator-based analog to information converters. In: IEEE. *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. [S.l.], 2011. p. 133–136.
- MAXFIELD, M. Tutorial: Linear feedback shift registers (lfsrs)—part 1. *online] EE Times*, 2018.
- MEDEIROS, R. J. V. d. *et al.* *Investigação sobre aplicação de amostragem compassiva a sinais de áudio*. Universidade Federal de Campina Grande, 2010.
- MOON, T. K.; STIRLING, W. C. *Mathematical methods and algorithms for signal processing*. [S.l.: s.n.], 2000.
- NEEDELL, D.; TROPP, J. A. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. arXiv, 2008. Disponível em: <https://arxiv.org/abs/0803.2392>.
- NOUASRIA, H.; ET-TOLBA, M. New sensing approach for compressive sensing using sparsity domain. In: *2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON)*. [S.l.: s.n.], 2018. p. 20–24.
- PARESCHI, F. *et al.* Hardware-algorithms co-design and implementation of an analog-to-information converter for biosignals based on compressed sensing. *IEEE transactions on biomedical circuits and systems*, IEEE, v. 10, n. 1, p. 149–162, 2015.
- RAGHEB, T. *et al.* A prototype hardware for random demodulation based compressive analog-to-digital conversion. In: IEEE. *2008 51st Midwest Symposium on Circuits and Systems*. [S.l.], 2008. p. 37–40.
- RANI, M.; DHOK, S. B.; DESHMUKH, R. B. A systematic review of compressive sensing: Concepts, implementations and applications. *IEEE Access*, v. 6, p. 4875–4894, 2018.
- RAPUANO, S. Analog-to-information converters: research trends and open problems. In: IEEE. *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*. [S.l.], 2016. p. 10–17.

REIS, V. d. L. *Conversor configurável analógico para informação*. João Pessoa: Universidade Federal de Campina Grande, 2017.

REIS, V. d. L. *et al.* Influence of integrators in the performance of analog-to-information converters. In: IEEE. *2016 1st International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)*. [S.l.], 2016. p. 118–121.

ROMBERG, J. Imaging via compressive sampling. *IEEE Signal Processing Magazine*, IEEE, v. 25, n. 2, p. 14–20, 2008.

S., S. K.; K.V., S. Reconstruction of mri images based on compressive sensing. In: *2019 International Conference on Communication and Signal Processing (ICCSP)*. [S.l.: s.n.], 2019. p. 0787–0791.

SALAHDINE, F.; KAABOUC, N.; GHAZI, H. E. One-bit compressive sensing vs. multi-bit compressive sensing for cognitive radio networks. In: *2018 IEEE International Conference on Industrial Technology (ICIT)*. [S.l.: s.n.], 2018. p. 1610–1615.

SALAN, S.; MURALIDHARAN, K. B. Image reconstruction based on compressive sensing using optimized sensing matrix. In: *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*. [S.l.: s.n.], 2017. p. 252–256.

SATHYABAMA, B.; SANKARI, S. S.; NAYAGARA, S. Fusion of satellite images using compressive sampling matching pursuit (cosamp) method. In: *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCV-PRIPG)*. [S.l.: s.n.], 2013. p. 1–4.

SHANNON, C. E. A mathematical theory of communication. *The Bell system technical journal*, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948.

SILVA, V. M. L. *Métodos de testes de conversores analógico para informação baseados no padrão IEEE 1241*. João Pessoa: Universidade Federal de Campina Grande, 2019.

SILVA, V. M. L.; SOUZA, A. A. L. de; CATUNDA, S. Y. C. Flexible a/d converter architecture targeting sparse signals. In: IEEE. *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. [S.l.], 2014. p. 1496–1500.

SILVA, V. M. L. *et al.* Non-uniform sampling based adc architecture using an adaptive level-crossing technique. In: *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. [S.l.: s.n.], 2017. p. 1–6.

SILVA, V. M. L. *et al.* Novel ieee-std-1241-based test methods for analog-to-information converter. *IEEE Transactions on Instrumentation and Measurement*, v. 69, n. 4, p. 1609–1619, 2020.

SINHA, S. *State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion*. 2021. Disponível em: <https://iot-analytics.com/number-connected-iot-devices/>.

SOUZA, C. P. de; BAIOCCHI, O. Energy resources in agriculture and forestry: How to be prepared for the internet of things (iot) revolution. In: *Energy Systems and Environment*. [S.l.]: InTechOpen, 2018.

STANKOVIĆ, L. *et al.* A tutorial on sparse signal reconstruction and its applications in signal processing. *Circuits, Systems, and Signal Processing*, Springer, v. 38, n. 3, p. 1206–1263, 2019.

TROPP, J. A.; GILBERT, A. C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, IEEE, v. 53, n. 12, p. 4655–4666, 2007.

TROPP, J. A. *et al.* Beyond nyquist: Efficient sampling of sparse bandlimited signals. *IEEE transactions on information theory*, IEEE, v. 56, n. 1, p. 520–544, 2009.

YOO, J. *et al.* Design and implementation of a fully integrated compressed-sensing signal acquisition system. In: IEEE. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2012. p. 5325–5328.