

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA
ELÉTRICA

Kátia Táila Silva de Melo

Compressão de sinais por Códigos Chirp

Juiz de Fora

2022

Kátia Táila Silva de Melo

Compressão de sinais por Códigos Chirp

Dissertação apresentada ao Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas Eletrônicos

Orientador: Carlos Augusto Duque, Prof. Dr.

Coorientador: Mahdi Khosravy, Prof. Dr

Juiz de Fora

2022

Kátia Táila Silva de Melo

Compressão de Sinais por Código Chirp

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas Eletrônicos

Aprovada em 15 de março de 2022.

BANCA EXAMINADORA

Prof. Dr. Carlos Augusto Duque - Orientador

Universidade Federal de Juiz de Fora

Prof. Dr. Leandro Rodrigues Manso Silva

Universidade Federal de Juiz de Fora

Prof. Dr. Danton Diego Ferreira

Universidade Federal de Lavras

Juiz de Fora, 28/02/2022.



Documento assinado eletronicamente por **Danton Diego Ferreira, Usuário Externo**, em 15/03/2022, às 16:10, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

Documento assinado eletronicamente por **Leandro Rodrigues Manso Silva, Professor(a)**, em



15/03/2022, às 16:10, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Carlos Augusto Duque, Professor(a)**, em 15/03/2022, às 16:11, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **0695598** e o código CRC **DC46AF80**.

Dedico este trabalho aos meus grandes amores que sempre estiveram ao meu lado. À todas pessoas que me apoiaram e não me deixaram desistir mesmo com todas as adversidades encontradas no meio do caminho. Ao meu Orientador Carlos Augusto Duquepor sempre confiar em mim e no meu trabalho. Ao meu Coorientador Mahdi Khosravypor contribuir com o trabalho. À minha família, meus pais e minha irmã que me ajudaram de todas as formas possíveis e impossíveis nunca saíram do meu lado e sempre entenderam tudo que precisei passar para chegar até aqui. Aos meus amigos Iago, Pedro e Lais por toda ajuda, por toda contribuição, e apoio inestimáveis.

AGRADECIMENTOS

Agradeço à Deus, minha família, meus amores, minhas filhas Luz e Lili que nunca saíram do meu lado. A todos colegas que me auxiliariam nesse período. Ao meu orientador Dr. Carlos Augusto Duque, por todo apoio e compreensão. Ao meu coorientador Dr. Mahdi Khosravy por incentivar esse tema e contribuir com o trabalho. Aos meus colegas de trabalho que me apoiaram e estiveram ao meu lado para que me fosse permitido cumprir com minhas obrigações do mestrado. Aos meus amigos que sempre me deram forças para continuar.

RESUMO

Compressive sensing é uma técnica que utiliza da esparsidade dos sinais em alguma dimensão, para processá-los enquanto são comprimidos utilizando menos amostras que técnicas como Shannon-Nyquist, possibilitando que menos memória seja utilizada, e conseqüentemente proporcionando o barateamento de *hardwares*. O método utilizado neste trabalho é focado no uso de matrizes determinísticas feitas com códigos de *chirp* para processar sinais sintéticos com harmônicos e reconstruí-los. É demonstrada a técnica neste trabalho, suas limitações e os resultados de sua aplicação em sinais sintéticos harmônicos. Uma análise dos casos foi realizada identificando os casos em que ela trás bons resultados e situações que não é recomendado seu uso. É visto que a técnica tem suas particularidades de implementação, mas dentro dos requisitos ela cumpre o objetivo proposto, recuperando o sinal com poucas amostras com um erro relativamente baixo possibilitando assim a redução de custos de armazenamento dos dados.

Palavras-chave: *Compressive Sensing*. Códigos de *Chirp*. Sinais harmônicos.

ABSTRACT

Compressive sensing is a technique that uses the sparseness of signals in some dimension, to process them while they are compressed using fewer samples than techniques such as Shannon-Nyquist, allowing less memory to be used, making cheaper hardware. The method used in this work is focused on the use of deterministic matrices made with chirp codes to process signals with harmonics and reconstruct them, the technique is demonstrated in this work, its limitations and the results of its application in harmonic signals, an analysis was made of the cases in which it brings good results and situations in which its use is not recommended. It is seen that the technique has its implementation particularities, but within the requirements it fulfills the proposed objective, recovering the signal with few samples with a relatively low error, thus enabling the reduction of data storage costs.

Keywords: *Compressive Sensing*. *Chirp* code. Harmonics.

LISTA DE ILUSTRAÇÕES

Figura 1 – Código de <i>Chirp</i> como uma sequência de número complexo	51
Figura 2 – Parte real e imaginária do código de <i>Chirp</i> da imagem 2.	52
Figura 3 – Componentes multiplicativas do código de <i>chirp</i> parte real (A e B) e imaginária (C e D) 3.	53
Figura 4 – DFT das componentes do código de <i>chirp</i> sendo (A)DFT da componente da frequência de base (B) DFT da componente da frequência de <i>chirp</i> e (C) DFT de ambas as componentes4.	54
Figura 5 – Fluxograma do algoritmo de recuperação do sinal.	57
Figura 6 – Fluxograma do ciclo de recuperação do sinal.	58
Figura 7 – Módulo da DFT do código de <i>Chirp</i> no topo, Módulo da DFT de $f(l)$ no meio e Módulo daDFT de uma senóide com frequência $r_5 2T \text{ mod } K$ em baixo.	59
Figura 8 – Módulo da DFT dos códigos de <i>chirp</i> separadamente no topo, Módulo da DFT da mistura dos códigos de <i>chirp</i> no meio e Módulo da DFT de \vec{f} em baixo.	60
Figura 9 – Ilustração da bijeção entre dois conjuntos $\{r_i\}$ e $\{2r_i T \text{ mod } K\}$ para os valores de K de 6, 7, 8, 9, 10, 11.	61
Figura 10 – Ilustração da bijeção entre dois conjuntos $\{r_i\}$ e $\{2r_i T \text{ mod } K\}$ para os valores de K de 12, 13, 14, 15, 16, 17.	62
Figura 11 – Mistura decifrada de r_1	63
Figura 12 – Espectro do sinal original acima e reconstruído abaixo.	65
Figura 13 – Sinal original e reconstruído sobrepostos.	66
Figura 14 – Resultados do Caso 1	71
Figura 15 – Resultados do Caso 2	72
Figura 16 – Resultados do Caso 3	73
Figura 17 – Resultados do Caso 4	76
Figura 18 – Resultados do Caso 5	77
Figura 19 – Resultados do Caso 6	78
Figura 20 – Caso 7: MSE para cada número de componentes de frequência para $K = 41$	80

Figura 21 – Caso 8: MSE para cada número de componentes de frequência para $K = 97$	80
Figura 22 – Caso 9: Espectro do sinal original e do sinal estimado por CS	82
Figura 23 – Caso 9: Sinal real e estimado no domínio do tempo	83
Figura 24 – MSE para sinal real com $K=41$ com uma, duas e três iterações	85
Figura 25 – MSE para sinal real com $K=41$ com duas iterações e ruídos de 20dB,40dB e 60dB	87
Figura 26 – Resultados do Caso 10	89
Figura 27 – Resultados do Caso 11	91
Figura 28 – Resultados do Caso 12	93
Figura 29 – Resultados do Caso 13	95
Figura 30 – Casos 1,2 e 3: Sinal original e sinal reconstruído no tempo	97
Figura 31 – Casos 1,2 e 3: Sinal original e sinal reconstruído no espectro da frequência	98
Figura 32 – Casos 4, 5 e 6: Sinal original e sinal reconstruído no tempo	99
Figura 33 – Casos 4, 5 e 6: Sinal original e sinal reconstruído no espectro da frequência	100
Figura 34 – Caso 10: Sinal original e sinal reconstruído no tempo	101
Figura 35 – Caso 10: Sinal original e sinal reconstruído no espectro da frequência	102
Figura 36 – Caso 11: Sinal original e sinal reconstruído no tempo	103
Figura 37 – Caso 11: Sinal original e sinal reconstruído no espectro da frequência	104
Figura 38 – Caso 12: Sinal original e sinal reconstruído no tempo	105
Figura 39 – Caso 12: Sinal original e sinal reconstruído no espectro da frequência	106
Figura 40 – Caso 13: Sinal original e sinal reconstruído no tempo	107
Figura 41 – Caso 13: Sinal original e sinal reconstruído no espectro da frequência	108
Figura 42 – Casos 1,2 e 3: Sinal original e sinal original com ruído	109
Figura 43 – Casos 1,2 e 3: Sinal original e sinal com ruído no espectro da frequência	110
Figura 44 – Casos 4, 5 e 6: Sinal original e sinal original com ruído	111

Figura 45 – Casos 4, 5 e 6: Sinal original e sinal com ruído no espectro da frequência	112
Figura 46 – Caso 10: Sinal original e sinal original com ruído	113
Figura 47 – Caso 10: Sinal original e sinal com ruído no espectro da frequência	114
Figura 48 – Caso 11: Sinal original e sinal original com ruído	115
Figura 49 – Caso 11: Sinal original e sinal com ruído no espectro da frequência	116
Figura 50 – Caso 12: Sinal original e sinal original com ruído	117
Figura 51 – Caso 12: Sinal original e sinal com ruído no espectro da frequência	118
Figura 52 – Caso 13: Sinal original e sinal original com ruído	119
Figura 53 – Caso 13: Sinal original e sinal com ruído no espectro da frequência	120

LISTA DE TABELAS

Tabela 1 – Casos estudados neste capítulo	69
Tabela 2 – Erros dos casos 1, 2 e 3	70
Tabela 3 – Erros dos casos estudados neste capítulo	75
Tabela 4 – Erros dos casos estudados neste capítulo	96

LISTA DE ABREVIATURAS E SIGLAS

CS	Compressive sensing
QEE	Qualidade de Energia Elétrica
ECG	Eletrocardiograma
EEG	Eletroencefalograma
MIMO	Multiple Input Multiple Output
RIP	Restricted Isometry Property
FFT	Transformada Rápida de Fourier
DFT	Transformada Discreta de Fourier
NSP	Espaço Nulo
BCH	Bose, Chaurdhuri e Hocquenghen
OOC	Códigos óticos ortogonais
RMSE	Erro quadrático médio absoluto

LISTA DE SÍMBOLOS

\forall	Para todo
\in	Pertence
Σ	Somatório
ϵ	Epsilon
$ x $	Norma de x
∞	Infinito

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVOS	17
1.2	DIVISÃO DO TRABALHO	19
2	REVISÃO TEÓRICA DE <i>COMPRESSIVE SENSING</i> .	20
2.1	ESPAÇO VETORIAL	20
2.2	NORMA	20
2.3	BASES E FRAMES	21
2.4	CONDIÇÕES DE ESPAÇO NULO (NSP)	23
2.5	PROPRIEDADE DE ISOMETRIA RESTRITA (RIP)	25
2.5.1	Relação entre RIP e NSP	25
2.6	MÉTODOS DE OTIMIZAÇÃO	26
2.7	COMPRESSIVE SENSING	27
2.7.1	Matrizes de códigos de <i>Chirp</i>	30
2.7.2	Matrizes de segunda ordem de Reed-Muller	32
2.7.2.1	Algoritmo de reconstrução rápida	34
2.7.3	Aproximação de Amini para matrizes que satisfazem RIP	36
2.7.4	Matrizes binárias Amini-Marvasti	37
2.7.4.1	Código BCH	38
2.7.4.2	Teoria dos conjuntos extremos	42
2.7.4.3	Expansão de gráficos desbalanceados	44
2.7.5	Matrizes com RIP estatístico	46
2.7.6	Matrizes determinísticas de curvas algébricas através de campos finitos	48
2.7.7	Matrizes construídas a partir de curvas algébricas sobre campos finitos	48
3	PROPOSTA DA METODOLOGIA	50
3.1	Apresentação da metodologia	50
3.2	Códigos de Chirp	50
3.3	Aproximação para extração dos parâmetros	54
3.3.1	Um único chirp	58
3.3.2	Mistura de 2 chirps	59

3.3.3	Bijecção entre dois conjuntos $\{r_i\}$ e $\{2r_i T \bmod K\}$	60
3.3.4	Decifrando (Dechirping)	62
3.4	Detecção e recuperação do sinal	63
3.5	Análise de K e taxa de compressão	65
3.6	Energia do sinal	67
3.7	Limitações	67
4	RESULTADOS	68
4.1	Método utilizado para gerar resultados	68
4.2	Análise dos Casos 1, 2 e 3	69
4.3	Análise dos Caso 4, 5 e 6	74
4.4	Análise dos Casos 7 e 8	79
4.5	Análise do caso 9	81
4.5.1	Sinal real com K=41 e frequências aleatórias	84
4.5.2	Sinal real com K=41 e com adição de ruído e frequências aleatórias	86
4.6	Análise dos casos 10, 11, 12 e 13	88
4.7	Análise dos casos com 100 interações	96
4.8	Análises dos casos com 100 interações e o sinal original com ruído	109
5	CONCLUSÕES	121
6	ANEXOS	123
6.1	Código MATLAB	123
	REFERÊNCIAS	132

1 INTRODUÇÃO

Um dos maiores problemas em processamento de sinais, tem sido o armazenamento de dados, isso requer *hardwares* mais robustos e conseqüentemente mais caros. O método tradicional utiliza a técnica de Shannon/Nyquist, em que se necessita de duas vezes o valor da frequência de amostragem para realizar o processamento do sinal [Qaisar et al. 2013][Baraniuk 2007][Eldar e Kutyniok 2012].

Em 2007 o desenvolvimento do método denominado de *Compressive sensing* deu um novo impulso na área de aquisição, compressão e processamento de dados [Baraniuk 2007]. Este método realiza a compressão do sinal simultaneamente ao seu processamento, com uma taxa inferior a de Nyquist, diminuindo assim o volume de dados a serem armazenados, e barateando os *hardwares* utilizados [Eldar e Kutyniok 2012].

Esse método tem inúmeras aplicações, por exemplo, compressão de imagens, espectral e hiper-espectral, interferometria de rádio, imagem microscópicas, radar e processamento de imagens [Rani e Dhok 2018]. Já na área biomédica temos o processamento de sinais de EEG (Eletroencefalograma) e ECG (Eletrocardiograma), aquisição de sinal neural, reconstrução da composição de bactérias [Zhang et al. 2010]. Temos também o setor de comunicação, com destaque para redes *wireless*, comunicação MIMO, redes de rádio cognitivos e conjuntos de antenas [Sharma et al. 2016]. Além dessas utilizações temos também área de reconhecimento de padrões onde são alguns exemplos reconhecimento de face, de gestos, da maneira de andar e de íris [Zhang e Zhao 2012].

Compressive sensing é uma técnica de processamento de sinais que utiliza da esparsidade do sinal em algum domínio de transformada para processá-lo utilizando menos dados. Um sinal esparso inclui mais elementos nulos ou considerados nulos do que elementos que contém alguma informação, por exemplo, um sinal de tensão em sistemas de potência, no domínio da frequência, possui apenas componentes em algumas frequências com energia significativa, sendo que na maior parte do espectro as energias são nulas ou com energia tão pequenas que poderiam ser descartadas de maneira adequada sem que se perca informação significativa do sinal. Assim, armazenando-se um vetor contendo poucos parâmetros no domínio

da transformada é possível reconstruir o sinal no domínio original com grande fidelidade [Eldar e Kutyniok 2012].

Compressive sensing comprime e processa o sinal esparsos através de uma matriz que pode ser aleatória ou determinística, e o recupera utilizando métodos de otimização. Neste trabalho temos a descrição do método *Compressive Sensing* sendo apresentadas as suas formas principais utilizando matrizes aleatórias e outras formas utilizando matrizes determinísticas, e como é feita a recuperação dos sinais [Khosravy et al. 2020] .

Além disso, será aprofundado um dos métodos que utilizam matrizes determinísticas e mostrada sua aplicação em sinais senoidais sintéticos, comprovando a aplicabilidade do método. Os dados serão analisados e comparados e por fim as devidas conclusões.

1.1 OBJETIVOS

Em Engenharia Elétrica temos uma grande necessidade de processar sinais, principalmente quando se fala em Qualidade de Energia Elétrica. Nestes casos o monitoramento da rede deve ser constante para observar falhas e desvios de conformidades. O armazenamento dos dados crus, adquiridos a altas taxas de amostragem requer um grande espaço de armazenamento dos sinais, tornando os *hardwares* mais caros por requerer uma memória maior, uma vez que o sinal é processado utilizando o critério de Nyquist-Shannon [Melo et al. 2020].

Com o aumento do uso de equipamentos com interfaces de eletrônica de potência (e.g. inversores e conversores de energia elétrica) nas redes elétricas de potência ao longo dos últimos anos, cada vez mais encontramos sinais harmônicos e não harmônicos na rede elétrica em frequências mais altas. Isto se deve, sobretudo, às frequências de chaveamento das chaves eletrônicas maiores que 9 kHz, dando origem às componentes que têm sido denominadas de Supra-harmônicas [Espín-Delgado et al. 2021], [Falco e Varilone 2021], [Basta e Morsi 2021], [Mendes et al. 2018] e [Mendes et al. 2020]. O processamento desses sinais requer taxa de amostragem muito superior à que vem sendo utilizada em análise harmônica tradicional, gerando com isso a necessidade de um grande espaço de

armazenamento. Deste modo a técnica de CS vem como possível solução para análise desses sinais, que têm ganhado muita atenção no estudo atual da QEE [Guevara et al.] e [Amaripadath et al. 2018].

O objetivo deste trabalho é utilizar a técnica de CS em sinais da área de engenharia elétrica, nesse caso sinais sintéticos que tem a mesma forma de sinais com harmônicos, com e sem ruídos. Avaliar a taxa de compressão denominada K , e o uso de uma matriz determinística feita de códigos de *chirp*.

A escolha apropriada da matriz de CS tem uma enorme diferença no resultado final. Sabe-se que matrizes aleatórias são incoerentes com quaisquer outras bases e satisfazem a condição de RIP, permitindo uma recuperação perfeita do sinal processado. Como as matrizes são aleatórias, não é possível reproduzi-las para a reconstrução do sinal, assim teriam que ser enviadas junto ao sinal do emissor para o receptor e essa prática é inviável, pois elas são geralmente de um tamanho grande e isso geraria uma necessidade de uma banda de transmissão muito grande.

Por esse motivo buscou-se matrizes determinísticas cujo formato conhecemos, que atendem aos requisitos para serem utilizadas no CS. Alguns exemplos desse tipo de matrizes são as matrizes de *Chirp*, matrizes de segunda ordem de Reed-Muller, matrizes binárias e matrizes construídas a partir de curvas algébricas sobre campos finitos.

Também pode-se utilizar matrizes *randômicas* estruturadas, elas têm a vantagem de ter uma rápida aquisição, um menor espaço de armazenamento requerido, facilidade em se reproduzi-la sem a necessidade de enviá-la, e transmissão de *overhead* reduzida, mas uma desvantagem dessas matrizes é que elas precisam de um número maior de medidas, quando comparadas com as matrizes *random* sem ser estruturadas. Alguns exemplos desse tipo de matrizes são as matrizes circulares e matrizes de toeplitz.

Neste trabalho serão estudados os métodos de *Compressive sensing* de matrizes determinísticas, pois o custo computacional de se gerar uma matriz *random* que satisfaça as condições necessárias para ser utilizada no CS geram um enorme custo computacional. Então optou-se pelo uso das matrizes determinísticas. Dentre as opções encontradas, foi escolhida a matriz determinística de *Chirp*, por ter sua forma conhecida, ser simples de se construir e oferecer todas as vantagens

de uma matriz determinística.

1.2 DIVISÃO DO TRABALHO

O Capítulo 2 apresenta a revisão bibliográfica, apresentando informações sobre o estado da arte relacionada a *Compressive Sensing*. Dentro deste capítulo serão abordados temas como: espaço vetorial, norma, bases e *frames*, condições de espaço nulo, propriedade de isometria restrita, métodos de otimização e *Compressive Sensing*. O Capítulo 3 irá apresentar a metodologia utilizada nesse trabalho e suas limitações.

No Capítulo 4 serão vistos os resultados gerados por esse trabalho e a discussão do método e suas aplicações. E o Capítulo 5 encerrará o trabalho explicando as aplicações, limitações e proposta para trabalhos futuros.

2 REVISÃO TEÓRICA DE *COMPRESSIVE SENSING*

Neste capítulo serão abordados os conceitos básicos para entendimento do método *Compressive Sensing*: diferentes tipos de formas dele e os métodos de otimização empregados para solucioná-lo.

2.1 ESPAÇO VETORIAL

Espaço vetorial é um espaço linear, que pode ser definido como uma coleção de objetos chamados vetores, que podem ser manipulados matematicamente. Em processamento de sinais, geralmente podemos modelar os sinais como um sistema linear, e portanto modelamos os sinais como vetores dentro de um espaço vetorial apropriado[Eldar e Kutyniok 2012].

Espaços vetoriais nos permitem aplicar ferramentas da geometria em um espaço real tridimensional, como comprimentos, distâncias e ângulos, para descrever e comparar sinais de interesse.

Nessa dissertação, sinais serão considerados funções de valor real com domínios discretos ou contínuos e também infinito ou finito. Iremos focar nos espaços vetoriais normatizados, ou seja, espaços vetoriais dotados de normas.

2.2 NORMA

Além de espaços vetoriais é importante frisar a definição de norma, que é uma função em que cada vetor de um espaço vetorial associa um número real não-negativo. Iremos focar nas normas l_p , que são definidas por [Christensen et al. 2003] $p \in [1, \infty]$ como:

$$\|\mathbf{x}\|_p = \begin{cases} (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}, & \text{se } p \in [1, \infty] \\ \max_{i=1,2,\dots,n} |x_i|, & \text{se } p = \infty \end{cases} \quad (2.1)$$

Em um espaço Euclidiano, podemos considerar o produto interno em \mathbb{R}^n , denotado por [Candes e Tao 2006]:

$$\langle \mathbf{x}, \mathbf{z} \rangle = \mathbf{z}^T \mathbf{x} = \sum_{i=1}^n x_i z_i$$

E esse produto interno nos leva à norma l_2

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} \quad (2.2)$$

Existem ainda os casos em que a norma l_p tem p menor que 1, ou seja, $0 < p < 1$, então a Equação 2.2 não satisfaz a igualdade, pois ela só é válida para $p \leq 1$, então essas normas são chamadas de quasinorma [Candes e Tao 2006]. E para esse caso é usado a notação:

$$\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})| \quad (2.3)$$

onde

$$\text{supp}(x) = \{i : x_i \neq 0\} \quad (2.4)$$

e no caso de $p = 0$ temos a norma 0, e ela é a cardinalidade do vetor \mathbf{x} .

As normas são utilizadas para medir a força do sinal ou o tamanho do erro [Hendrickx e Olshevsky 2010]. Ao medir a aproximação de um erro usando uma norma l_p , nós observamos que para um p grande, o erro tende a se espalhar mais uniformemente entre dois coeficientes, enquanto que para um p menor leva a um erro distribuído menos uniforme e tende a ser esparso.

2.3 BASES E FRAMES

Um conjunto $\{\phi_i\}_{i=1}^n$ é chamado Base de \mathfrak{R}_n se os vetores são linearmente independentes e se qualquer vetor ϕ puder ser escrito como uma combinação linear de ϕ_1, ϕ_2, \dots [Christensen 2008]. Isso implica que cada vetor no espaço tem apenas uma única representação como uma combinação linear dos vetores da base [Christensen et al. 2003]. Para qualquer $\mathbf{x} \in \mathfrak{R}_n$, existe um único coeficiente $\{c_i\}_{i=1}^n$ tal que:

$$\mathbf{x} = \sum_{i=1}^n c_i \phi_i \quad (2.5)$$

Se Φ representa uma matriz $n \times n$ com as colunas formadas por ϕ_i e \mathbf{c} representa o vetor de comprimento n com entradas c_i , então podemos representar essa relação da seguinte forma:

$$\mathbf{x} = \Phi c. \quad (2.6)$$

Se tivermos uma base ortonormal, definida como um conjunto de vetores $\{\phi_i\}_{i=1}^n$ satisfazendo:

$$\langle \phi_i, \phi_j \rangle = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad (2.7)$$

Uma base ortonormal tem a vantagem que os coeficientes c podem ser facilmente calculados como:

$$c_i = \langle x, \phi_i \rangle, \quad (2.8)$$

ou

$$c = \Phi^T \mathbf{x} \quad (2.9)$$

Em notações de matrizes. E isso pode ser facilmente verificado desde que a ortonormalidade das colunas de Φ signifique que $\Phi^T \Phi = I$, onde I é a matriz identidade $n \times n$. E isso é útil para generalizar o conceito de base, pois assim um conjunto de vetores possivelmente linearmente dependentes junto a base pode ser utilizado e ele é denominado *frame*, ou seja, é a generalização de uma base de um espaço vetorial [Kovacevic e Chebira 2007]. Para processamento de sinais o *frame* é uma maneira redundante e estável de representar um sinal. Podemos representá-lo da seguinte maneira:

$$A \|\mathbf{x}\|_2^2 \leq \|\Phi^T \mathbf{x}\|_2^2 \leq B \|\mathbf{x}\|_2^2 \quad (2.10)$$

Quando A é escolhido como o maior valor possível e B como o menor, chamamos eles de limites ótimos de *frame*. Se A e B podem ser escolhidos tais que $A = B$, então o *frame* é chamado *A-tight* e se $A = B = 1$ então Φ é um *frame* de Parseval ou *tight frame* [Mallat 1999]. Ele satisfaz uma versão generalizada da identidade de Parseval.

Um *frame* $\{e_k\}_{k \in N}$ para \mathbf{v} é *tight* com limite A se e somente se:

$$\mathbf{v} = \frac{1}{A} \sum_k \langle v, e_k \rangle e_k \quad (2.11)$$

para todo $\mathbf{v} \in V$.

Além disso, temos que um *frame* é chamado norma equivalente se existe algum $\lambda > 0$ tal que $\|\phi_i\|_2$ para todo $i = 1, \dots, n$, e ele é chamado norma unitária se $\lambda = 1$. *Frames* podem permitir que os dados sejam representados de várias formas dado a sua redundância [Bodmann e Casazza 2011], para um dado sinal \mathbf{x} , existe uma infinidade de coeficientes de vetores c tal que $\mathbf{x} = \Phi c$. Existem também os dual *frames*, para ter essa definição eles devem satisfazer a seguinte Equação:

$$\Phi \bar{\Phi}^T = \bar{\Phi} \Phi^T = I \quad (2.12)$$

A escolha de $\bar{\Phi} = (\Phi \Phi^T)^{-1} \Phi$ é chamada de Dual *frame* canônico e também é conhecido como Pseudoinversa Moore-Penrose [Eldar e Kutyniok 2012]. Com $A > 0$, Φ deve ter linhas independentes e isso garante que $\Phi \Phi^T$ é invertível, então $\bar{\Phi}$ é bem definido. Assim, uma maneira de se obter coeficientes viáveis é através da equação 2.13.

$$c_d = \bar{\Phi}^T \mathbf{x} = \Phi^T (\Phi \Phi^T)^{-1} x \quad (2.13)$$

E assim pode-se ver que essa sequência é a menor sequência de coeficientes na norma l_2 , $\|c_d\|_2 \leq \|c\|_2$ para todo c tal que $\mathbf{x} = \Phi c$ [Bodmann e Casazza 2011].

No caso de sinais esparsos, é comum que uma base ou um *frame* seja chamado de dicionário, e os elementos do dicionário são chamados de átomos.

2.4 CONDIÇÕES DE ESPAÇO NULO (NSP)

Ao buscar recuperar sinais esparsos a partir de medidas, para qualquer par de vetores distintos \mathbf{x} , \mathbf{x}' temos que $A\mathbf{x} \neq A\mathbf{x}'$, ou seja, a matriz deve obedecer à condição de espaço nulo. Pois não seria possível distinguir tais vetores baseados nas medidas y . Assim podemos concluir que A representa unicamente todo

$x \in \Sigma_k$ se e somente se o espaço nulo de A , $\aleph(A)$ não contém vetores em Σ_{2k} [Donoho e Elad 2003].

É importante definir o conceito de posto, que é dado por [Cohen e Dahmen 2009]:

Definição 2.1 *O posto de uma dada matriz A é o menor número de colunas de A que são linearmente dependentes.*

E com essa definição temos o seguinte teorema:

Teorema 2.1 *Para qualquer vetor $\mathbf{y} \in \mathfrak{R}^m$ existe pelo menos um sinal $\mathbf{x} \in \Sigma_k$ tal que $\mathbf{y} = A\mathbf{x}$ se e somente se $\text{posto}(A) > 2k$. [Donoho e Elad 2003]*

Quando o vetor é exatamente esparsos, o posto caracteriza bem se a recuperação do sinal é possível, mas para sinais aproximadamente esparsos é necessário uma condição mais restritiva no espaço nulo de A [Cohen e Dahmen 2009]. É importante que $\aleph(A)$ não tenha nenhum vetor que seja muito compressível além de vetores esparsos.

Para isso supõe-se que $\Lambda \subset \{1, 2, \dots, n\}$ é um subconjunto de índices e seja $\Lambda^c = \frac{\{1, 2, \dots, n\}}{\Lambda}$. Então temos \mathbf{x}_Λ que é um vetor de comprimento n feito a partir de entradas de \mathbf{x} indexadas por Λ^c a zero. Similarmente temos que A_Λ é uma matriz $m \times n$ obtida definindo as colunas de A indexadas por Λ^c a zero. Assim pode-se definir que [Eldar e Kutyniok 2012]:

Definição 2.2 *Uma matriz A satisfaz a propriedade do espaço nulo (NSP) de ordem k , se existe uma constante $C > 0$ tal que:*

$$\|h_\Lambda\|_2 \leq C \frac{\|h_{\Lambda^c}\|_1}{\sqrt{k}} \quad (2.14)$$

para todo $h \in \aleph(A)$ e para todo Λ tal que $|\Lambda| \leq k$. [Eldar e Kutyniok 2012]

A propriedade do espaço nulo quantifica a noção de que vetores no espaço nulo de A não devem ser muito concentrados em um pequeno subconjunto de índices.

2.5 PROPRIEDADE DE ISOMETRIA RESTRITA (RIP)

A propriedade de isometria restrita (RIP) é muito importante quando se tem ruído no sinal.[Candes e Tao 2005]

Definição 2.5.1 *Uma matriz A satisfaz a propriedade de isometria restrita de ordem k se existe $\delta_k \in (0, 1)$ tal que:*

$$(1 - \delta_k)\|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1 + \delta_k)\|\mathbf{x}\|_2^2 \quad (2.15)$$

para todo $\mathbf{x} \in \Sigma_k$.

Ou seja, A preserva aproximadamente a distância entre qualquer par de vetores K esparsos. Foram considerados limites simétricos, mas eles podem ser arbitrários tais que:

$$\alpha\|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq \beta\|\mathbf{x}\|_2^2 \quad (2.16)$$

onde, $0 < \alpha \leq \beta \leq \infty$. E isso nos leva ao seguinte Lema[Deedell e CoSaMP 2008]:

Lema 2.5.1 *Suponha que A satisfaça a condição RIP de ordem k com constante δ_k . Seja γ um número inteiro positivo. Então A satisfaz a condição de RIP de ordem $k' = \gamma\lfloor \frac{k}{2} \rfloor$ com $\delta_{k'} < \gamma\delta_k$, onde $\lfloor \cdot \rfloor$ é a parte inteira.*

2.5.1 Relação entre RIP e NSP

Se uma matriz satisfaz a propriedade RIP, então ela satisfaz a propriedade de espaço nulo, podemos ver pelo teorema[Eldar e Kutyniok 2012]:

Teorema 2.5.1 *Suponha que A satisfaça a RIP de ordem $2k$ com $\delta_{2k} < \sqrt{2} - 1$. Então A satisfaz a propriedade de espaço nulo de ordem $2k$ com uma constante:*

$$C = \frac{2}{1 - (1 + \sqrt{2})\delta_{2k}} \quad (2.17)$$

A prova desse teorema é feita com dois lemas, o primeiro lema se inicia das desigualdades das normas padrão, relacionando um vetor k -esparso a um vetor em \mathbb{R}^k . Assim tem-se o primeiro Lema:

Lema 2.5.1.1 *Suponha que $u \in \Sigma_k$, assim:*

$$\frac{\|u\|_1}{\sqrt{k}} \leq \|u\|_2 \leq \sqrt{k}\|u\|_\infty \quad (2.18)$$

Já o segundo Lema tem um resultado mais generalista, que serve para todo h . Embora para provar o Teorema anterior nós só precisamos de $h \in \mathfrak{R}(A)$, poderemos usá-lo quando tiver um problema de recuperação de sinal esparso com ruído. O Lema diz que:

Lema 2.5.1.2: *Suponha que A satisfaça a RIP de ordem $2k$, e seja $h \in \mathfrak{R}^n$, $h \neq 0$ seja arbitrário. Se Λ_0 é qualquer subconjunto de $\{1, 2, \dots, n\}$ tal que $|\Lambda_0| \leq k$. Defina Λ_1 como o conjunto de índices correspondentes as entradas $k_{\Lambda_0^c}$ com a maior magnitude, e seja $\Lambda = \Lambda_0 \cup \Lambda_1$. Assim:*

$$\|\mathbf{h}_\Lambda\|_2 \leq \alpha \frac{\|\mathbf{h}_{\Lambda_0^c}\|_1}{\sqrt{k}} + \beta \frac{|\langle A\mathbf{h}_\Lambda, Ah \rangle|}{\|\mathbf{h}_\Lambda\|_2}, \quad (2.19)$$

onde,

$$\alpha = \frac{\sqrt{2}\delta_{2k}}{1 - \delta_{2k}}, \beta = \frac{1}{1 - \delta_{2k}}. \quad (2.20)$$

Com esses dois Lemas podemos fazer a prova do Teorema que relaciona NSP com RIP.

2.6 MÉTODOS DE OTIMIZAÇÃO

Existem vários métodos para recuperar um sinal esparso \mathbf{x} , alguns deles são: algoritmos de minimização de ℓ_1 , algoritmo *Greedy* e algoritmos combinatórios. Este trabalho focou nos algoritmos de minimização de ℓ_1 . Dados as medidas de \mathbf{y} e sabendo que o sinal original \mathbf{x} é esparso ou compressível, é natural que tentemos recuperá-lo resolvendo o problema de otimização [Candes et al. 2011] da equação 2.21.

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z}} \|\mathbf{z}\|_0; \text{ sujeito a } \mathbf{z} \in B(\mathbf{y}), \quad (2.21)$$

Onde $B(\mathbf{y})$ garante que $\hat{\mathbf{x}}$ é consistente com as medidas de \mathbf{y} . Se as medidas forem exatas e livres de ruído, podemos fazer $B(\mathbf{y}) = \{\mathbf{z} : A\mathbf{z} = \mathbf{y}\}$, mas se

as medidas tem ruído aí temos que considerar $B(\mathbf{y}) = \{\mathbf{z} : \|\mathbf{Az} - \mathbf{y}\|_2 \leq \epsilon\}$. Nos dois casos encontramos o \mathbf{x} mais esparsa consistente com as medidas de \mathbf{y} [Baraniuk et al. 2011].

Para o caso de \mathbf{x} não ser esparsa, fazemos $\mathbf{x} = \Phi\mathbf{c}$ e assim temos[Davenport et al. 2012] a equação 2.22.

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{z}} \|\mathbf{z}\|_0 \text{ sujeito a } \mathbf{z} \in B(\mathbf{y}), \quad (2.22)$$

Onde $B(\mathbf{y}) = \{\mathbf{z} : \mathbf{A}\Phi\mathbf{z} = \mathbf{y}\}$ e $\Phi = \mathbf{I}$.

A norma ℓ_0 é não convexa e é muito difícil de resolver essa minimização, para resolver esses problemas substituímos a norma ℓ_0 pela norma ℓ_1 que é uma aproximação convexa, de acordo com a equação 2.23[Vidyasagar 2019].

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z}} \|\mathbf{z}\|_1 \text{ sujeito a } \mathbf{z} \in B(\mathbf{y}), \quad (2.23)$$

Como $B(\mathbf{y})$ é convexo esse problema é solucionável, e quando $B(\mathbf{y}) = \{\mathbf{z} : \mathbf{Az} = \mathbf{y}\}$ o problema pode ser apresentado como linear.

2.7 COMPRESSIVE SENSING

Um dos maiores problemas em processamento de sinais é o armazenamento dos dados, isso é algo que encarece a tecnologia pois requer *hardwares* cada vez mais robustos e conseqüentemente mais caros[Davenport et al. 2012]. Para amostrar sinais, geralmente é utilizado o Teorema de Nyquist-Shannon, e ele necessita de pelo menos o dobro da frequência do sinal para realizar o processamento[Baraniuk et al. 2011]. Com a evolução das tecnologias o número de dados armazenados para processar tem sido cada vez maiores, e as frequências também, tornando impraticável esse método de processamento de sinais.

O método *compressive sensing* (CS) vem para revolucionar essa área da engenharia, onde para processar sinais esparsos, a taxa utilizada é inferior a de Nyquist-Shannon, e a recuperação do sinal é perfeita, desde que o sinal seja esparsa e atenda a alguns outros requisitos[Melo et al. 2020]. Isso proporciona soluções para diversos problemas e barateamento de diversas tecnologias, como por exemplo

no caso de processamento de imagens, principalmente médicas, onde normalmente o sinal é processado com uma taxa altíssima, por ter característica esparsa, nesse novo método o seu processamento é melhorado por utilizar menos amostras e recuperar com pouca ou nenhuma perda o sinal inicial[Bryan e Leise 2013].

Este trabalho será focado no processamento de sinais elétricos, pois em certo espaço amostral o sinal é esparsa e aproveitando dessa esparsidade poderemos trazer grandes melhorias nos processamentos de sinais da área de Engenharia Elétrica[Boche et al. 2015]. Para isso será explicado como funciona esse método, e suas diversas formas de soluções para variados tipos de problemas.

Como vimos, para utilizar CS precisamos de um sinal esparsa no domínio, utilizado em alguma base ortonormal, e devido a essa esparsidade podemos comprimir o sinal com poucas amostras. O nível de compressão depende do nível de esparsidade do sinal e esse método tem o nome de CS porque processa o sinal ao mesmo tempo que o comprime, a Equação básica é [Baraniuk 2007] :

$$\mathbf{y} = \Phi \cdot \mathbf{x} \quad (2.24)$$

onde \mathbf{y} é o vetor do sinal comprimido e processado, Φ é a matriz de CS e \mathbf{x} é o sinal já no seu modo esparsa. Tem-se que \mathbf{x} é um sinal de comprimento n Φ , uma matriz de tamanho $m \times n$ onde $n \gg m$. Essa matriz retangular m por n , dá m medidas através da combinação linear de n amostras do sinal [DeVore 2007]. Assim tem-se a taxa de compressão sendo $\frac{n}{m}$ que acontece ao mesmo tempo das medidas e por isso esse método é chamado *compressive sensing*. O sinal a ser detectado pode não ser esparsa no domínio original, nesses casos pode-se vê-lo em uma base ortonormal Ψ como um sinal \mathbf{s} , representação esparsa do sinal \mathbf{x} , e isso é representado de acordo com a equação 2.25.

$$\mathbf{x} = \Psi \cdot \mathbf{s} \quad (2.25)$$

Essas equações e ideias são a base do CS, e a partir delas tem-se que escolher um método para obter \mathbf{y} a partir da recuperação de \mathbf{x} . Nesse método a matriz Φ pode ser aleatória ou determinística.

Essa matriz deve ter uma construção de forma que não destrua nenhuma informação contida no sinal original. Mas como $\Phi \in \mathfrak{R}^{m \times n}$ e $m < n$ não é possível, pois teria-se um problema indeterminado para resolver. Ao restringir a um sinal k -esparso, pode-se resolver essa questão. Se conhecer as posições das k entradas não nulos, pode-se formar a matriz $m \times k$ onde $m \geq k$ e a solução pode ser feita por mínimos quadrados.

Mas será difícil existirem problemas onde se conheça as posições dos elementos não nulos, mas existem condições para solucionar o problema como a propriedade de isometria restrita, RIP definida em (2.15). A constante dessa propriedade dá a medida de quanto pode-se mudar o comprimento do vetor k -esparso. E também relaciona ao *kernel* da matriz. Também pode-se definir uma versão mais simplificada dessa propriedade, chamada NSP (2.14). E pode-se ver a relação dessas duas propriedades na Equação (2.17).

A questão mais importante para se criar a matriz de CS, dada uma base Ψ , tal que essa matriz tenha um RIP de ordem alta. Uma maneira é fazendo que essa matriz seja aleatória e gerada da seguinte forma:

Formar Ψ através da amostragem de vetores com n colunas uniformemente na esfera unitária em \mathfrak{R}^m , sejam as entradas de Ψ independente e identicamente distribuídas normal com média 0 e variância $\frac{1}{m}$, ou seja as entradas de Ψ independente e identicamente distribuídas pela distribuição simétrica de Bernoulli, $\Psi_{ij} = \pm \frac{1}{\sqrt{m}}$ com probabilidade igual ou outra distribuição subgaussiana, todas as condições citadas obedecem a propriedade RIP de ordem k [Pope 2009].

Outra maneira de garantir a estabilidade de matrizes de medida é garantindo um alto grau de incoerência entre a matriz de medida Φ e a matriz da base Ψ .

Para as matrizes aleatórias seguem-se essas condições para criá-las, mas para matrizes determinísticas, como é conhecido seu comportamento apenas escolhem-se modelos que atendam as restrições necessárias para resolver o problema de CS.

Algumas matrizes determinísticas conhecidas que atendem a essas demandas são: Matrizes de *Chirp*, matrizes de segunda ordem de Reed-Muller, matrizes Binárias e matrizes criadas via curvas algébricas através campos finitos. Será demonstrado um pouco sobre o funcionamento dessas matrizes.

2.7.1 Matrizes de códigos de *Chirp*

O problema de recuperação do sinal pode ser aproximado através da observação de encontrar o sinal que corresponde a pequenas combinações lineares das colunas de $\Theta = \Phi \cdot \Psi$ que formam \mathbf{y} . A matriz de códigos de Chirps é construída da seguinte forma, os sinais de *Chirp* são colocados em colunas, cada um deles com comprimento K , e eles têm a seguinte forma [Applebaum et al. 2009] vista na equação 2.26.

$$\mathbf{v}_{m,r}(l) = \alpha \cdot e^{\frac{j2\pi ml}{K} + \frac{j2\pi r l^2}{K}}, \quad m, r \in Z_k \quad (2.26)$$

onde tem-se as frequências bases definidas por m_i e as taxas de *chirp* definidas por r_i . As taxas de *chirp* podem ser recuperadas de \mathbf{y} a partir da função $f(l) = \bar{\mathbf{y}}(l)\mathbf{y}(l+T)$, onde $\bar{\mathbf{y}}(l)$ é, onde para o índice $l+T$ é feita a operação *mod* K . E ao substituir $\mathbf{y}(l)$ nessa função é gerada a equação 2.28 [Applebaum et al. 2009].

$$\mathbf{y}(l) = s_1 \cdot e^{\frac{j2\pi m_1 l}{K} + \frac{j2\pi r_1 l^2}{K}} + s_2 \cdot e^{\frac{j2\pi m_2 l}{K} + \frac{j2\pi r_2 l^2}{K}} + \dots \quad (2.27)$$

E ao aplicarmos esse $\mathbf{y}(\mathbf{l})$ na equação anterior, tem-se a equação 2.28.

$$f(l) = \sum_i |s_i|^2 \cdot e^{\frac{j2\pi}{K} \cdot (m_i \cdot T + r_i T^2)} \cdot e^{\frac{j2\pi r_i l \cdot T}{K}} + CT \quad (2.28)$$

Onde CT são os termos cruzados e eles são dados pela equação 2.29.

$$\mathbf{s}_p \cdot \bar{\mathbf{s}}_q \cdot e^{\frac{j \cdot 2\pi}{K} \cdot (m_p \cdot T + r_p \cdot T^2)} \cdot e^{\frac{j \cdot 2 \cdot \pi}{K} \cdot l \cdot (m_p - m_q + 2 \cdot T \cdot r_p)} \cdot e^{\frac{j \cdot 2 \cdot \pi}{K} \cdot l^2 \cdot (r_p - r_q)}, \quad (2.29)$$

que são *chirps*. É possível perceber que a Equação (2.28) tem a forma de senoides com frequências discretas dadas por $2 \cdot r_i \cdot T \bmod K$. Para se ter uma bijeção entre as taxas de *chirp*, K deve ser um número primo [Zhang et al. 1995]. É importante observar que os termos cruzados são desprezíveis.

Como \mathbf{x} é esparso, \mathbf{y} tem o número suficiente de *chirps*. Então uma Transformada rápida de Fourier (FFT) obtida a partir de (2.28) terá um espectro com os picos em $2 \cdot r_i \cdot T \bmod K$ onde serão obtidas as taxas de *chirp*.

Com as taxas de *chirp* r_i pode-se transformar os sinais com essa taxa em senoides. Para fazer isso basta multiplicar (2.27) por $e^{\frac{-j.2.\pi.r_i.l^2}{K}}$. Ao aplicar a FFT no sinal resultante, tem-se os valores de m_i e s_i . Dessa forma escolhendo Θ como:

$$\Theta_{k,l} = e^{\frac{j.2.\pi.m.l}{K} + \frac{j.2.\pi.r.l^2}{K}} : k = K.r + m \in Z_{K^2} \quad (2.30)$$

Pode-se escrever Θ como uma função de m conjuntos de matrizes $U_t(t = 1, \dots, k)$ tal que:

$$\Theta = [U_{r1}, U_{r2}, \dots, U_{rk}] \quad (2.31)$$

Onde tem-se que U_{rk} é uma matriz quadrada de tamanho K , que as colunas são formadas por vetores de sinais de *chirp* com uma taxa $r = r_t$ e uma frequência base de m variando de 0 a $K-1$. E assim pode-se perceber que $\mathbf{y} = \Theta.s$ tem a forma de (2.27) e \mathbf{y} é dado e Θ pode ser formado de acordo com o algoritmo abaixo [Applebaum et al. 2009]:

1º Passo Selecionar um $T \in \mathbf{Z}_k$, onde $T \neq 0$.

2º Passo Selecionar um critério de parada ϵ .

3º Passo Obter $f(l) = \bar{\mathbf{y}}(l)\mathbf{y}(l + T)$.

4º Passo Obter comprimento K da FFT a partir de $f(l)$.

5º Passo Encontrar a localização do pico na FFT obtida. Tomar isso como $2.r_i.T \bmod K$ e manter o único r_i combinando com a localização.

6º Passo Multiplicar $y(l)$ por $e^{\frac{-j.2.\pi.r_i.l^2}{K}}$, e então obter o comprimento K da FFT.

7º Passo Encontrar a localização do pico e manter ele como m_i . O valor do pico pode ser utilizado para a recuperação de \mathbf{s}_i .

8º Passo Atualizar \mathbf{y} através da subtração de $\mathbf{s}_i.e^{\frac{j.2.\pi}{k} \cdot (m_i.l + r_i.l^2)}$ a partir de \mathbf{y} .

9º Passo Repetir o 3º ao 8º passo até que se obtenha $\|\mathbf{y}\|_2^2 < \epsilon$ ou que tenha-se feito M interações.

2.7.2 Matrizes de segunda ordem de Reed-Muller

Os códigos de Reed-Muller são bem antigos e possuem diversas aplicações. Eles foram descobertos por Muller e Reed, que fizeram um algoritmo de decodificação deles. Os códigos são inicialmente dados como binários, mas existem generalizações de códigos q-nários [Candes e Tao 2005].

No caso de matrizes de detecção, serão usados como binários. Existem diversas maneiras de descrever esses códigos e nesse trabalho foi escolhido uma delas.

No mundo existe uma dualidade entre tempo e frequência, as senóides são autofunções de mudanças no tempo. Com isso inicia-se a descrição do espaço binário de Hamming em que existe mudanças discretas e equivalentes no tempo e na frequência. As funções de Walsh são análogas as senóides. Já as funções de segunda ordem de Reed-Muller são análogas aos *chirps*.

É importante ressaltar que as funções de Walsh são funções de primeira ordem de Reed-Muller e elas são funções $Z_2^m \implies R$ definidas por:

$$\Phi_{0,b}(a) = \frac{1}{\sqrt{2^m}} \cdot (-1)^{b^T \cdot a} \quad (2.32)$$

As funções de Walsh criam uma base ortonormal para R^{2^m} e portanto H_m é unitária. Elas podem ser vistas como as senóides do mundo binário, onde b define a frequência binária.

Existem 2^m funções Walsh em Z_2^m . Elas levam para as linhas da matriz de Hadamard H_m e formam uma base ortonormal, então H_m é unitária. Essas funções são autofunções com os seguintes deslocamentos:

$$\Phi_{0,b}(a + \varepsilon) = (-1)^{b^T \cdot \varepsilon} \cdot \Phi_{0,b}(a). \quad (2.33)$$

E esses deslocamentos podem ser comparados aos *chirps* no mundo binário.

Uma senóide é chamada linearmente *chirped* de acordo com que a frequência muda linearmente no tempo. Pode-se construir funções no caso binário onde a matriz P é simetricamente binária como visto na equação (2.34).

$$\Phi_{P,b}(a) = \frac{(-1)^{w.t.a}}{\sqrt{2^m}} \cdot i^{(2.b+P.a)^T \cdot a} \quad (2.34)$$

Essas são funções de segunda ordem Reed-Muller, o parâmetro P representa as matrizes binárias simétricas e o \mathbf{b} os vetores binários. As funções de Walsh são associadas quando $P = 0$.

As funções de segunda ordem Reed-Muller possuem importantes propriedades, uma delas é que, como P é uma matriz binária fixada simétrica, o conjunto $F = \{\Phi_{P,b} | b \in Z_2^m\}$ faz uma base ortonormal para R^{2^m} . Cada função nessa base é uma autofunção de um grupo comutativo de operadores tempo-frequência definidos por P [Howard e Calderbank 2006].

Focando nas funções reais de segunda ordem de Reed-Muller, as matrizes P tem zeros na diagonal principal.

Seja U_P uma matriz unitária correspondendo a uma base ortonormal F_p . Em [Howard e Calderbank 2008] foi proposto que $2^m \times 2^{\frac{m \cdot (m+1)}{2}}$ matrizes de *compressive sensing* são da forma em:

$$\Phi_{RM} = (U_{P_1}, U_{P_2}, \dots, U_{P_{\frac{m \cdot (m+1)}{2}}}). \quad (2.35)$$

Normalizando a Equação por $\frac{1}{\sqrt{2^m}}$, os elementos na matriz se tornam ± 1 . As colunas de Φ_{RM} formam um *tight frame* para \Re^{2^m} com redundância de $2^{\frac{m \cdot (m-1)}{2}}$, ou seja, temos o resultado na equação 2.36.

$$\Phi_{RM} \cdot \Phi_{RM}^T = 2^{\frac{m \cdot (m-1)}{2}} \cdot I_{2^m \times 2^m} \quad (2.36)$$

Sendo m par, então pode-se provar que um limite adicional para o conjunto de matrizes P é DG_{2h} . Eles são conjuntos binários de matrizes simétricas de diagonal zero $m \times m$ com a propriedade que para qualquer conjunto de matrizes $P, Q \in DG_{2h}$ o posto de $P+Q$ é pelo menos $m-2h$. Assim DG_{2h} são subconjuntos, de tal maneira que $DG_0 \subseteq DG_2 \dots$ sendo relacionados com sub-códigos do código de segunda ordem de Reed-Muller.

A propriedade RIP é essencial para CS. Nessa situação onde os auto-valores das matrizes de Gram são definidas pelos conjuntos de k colunas das matrizes de

processamento que estão no intervalo $[1 - \delta, 1 + \delta]$, entretanto essa característica é comum para uma matriz *randômica* e em [Howard e Calderbank 2008] foi visto que uma matriz em que $m = 6$ é quase idêntica a uma matriz Gaussiana.

2.7.2.1 Algoritmo de reconstrução rápida

A proposta é criar uma matriz determinística de CS explorando a formação das funções de segunda ordem de Reed-Muller para construir um algoritmo de recuperação computacionalmente prático.

Uma característica essencial dessas funções para esse estado é: Pegar qualquer $\epsilon \in \mathcal{Z}_2^m$, e então temos a equação 2.37.

$$\phi_{P,b}(a + \epsilon) \cdot \phi_{P,\hat{b}}(a) = \frac{(-1)^{b^T \cdot \epsilon} \cdot (-1)^{\epsilon^T \cdot P \cdot a}}{2^m} \quad (2.37)$$

Isso produz uma função de Walsh com frequência P_e .

Com uma função de segunda ordem de Reed-Muller desconhecida φ , tem-se um algoritmo de recuperação rápida dos parâmetros. Para cada unidade do peso $(\epsilon_j)_k = \delta_{j,k}$, calcula-se o produto na Equação 2.37 e implementa-se a transformada rápida de Hadamard.

O maior elemento dessa transformada estará em $P_{\epsilon,j}$, ou seja, coluna j de P . Então para realizar o *dechirp*, multiplicamos $\phi_{(P,0)}$ e uma transformada de Hadamard extra nos leva a b . Os parâmetros desconhecidos da função de Reed-Muller podem ser obtidos com uma complexidade de $(m + 1)^2 \cdot 2^m$ multiplicações e a busca de 2^m termos em cada transformada.

A partir da Equação 2.26 e temos a 2.38.

$$\mathbf{y} = \sum_{i=1}^k c_i \cdot \phi_{P_i}, b_i. \quad (2.38)$$

O algoritmo rápido é estabelecido com a utilização de multiplicação por partes dos dados com o resultado visto na equação (2.39).

$$\mathbf{y}(a + \epsilon) \cdot \hat{\mathbf{y}}(a) = \frac{1}{2^m} \sum_{i=1}^k c_i^2 \cdot (-1)^{b_i^T \cdot \epsilon} \cdot (-1)^{\epsilon^T \cdot P_i \cdot a} + chirps. \quad (2.39)$$

Onde o termo a direita de (2.39) é uma combinação linear de funções de Walsh adicionadas de *chirps* consistindo nos termos cruzados. Esses termos são funções de Reed-Muller de segunda ordem com P_s não-nulos. Implementando a transformada de Hadamard em (2.39) termina nos picos das frequências $P_{l,\epsilon}$ para cada P_l . Como essas funções com P não-zeros possuem as transformadas distribuídas na frequência, isso resulta em picos com interferências. Concluindo que P_l é uma das componentes do sinal, então uma estimação robusta do b_l correspondente é possível através do *de-chirp* dos dados. Assim tem-se a equação 2.40.

$$\mathbf{y}(a) \cdot \phi_{P,\hat{0}}(a) = c_i (-1)^{wt(b)} \cdot (-1)^{b^T \cdot a} + chirps. \quad (2.40)$$

E assim continua-se aplicando a transformação de Hadamard.

O algoritmo passo a passo é dado por:

1º Passo Obtenha o próximo P_j com a componente de maior energia ($|c_j|^2$) do dado comprimido e processado \mathbf{y}' .

2º Passo Obtenha a frequência b_j .

3º Passo Obtenha c_i que minimiza $\|\mathbf{y}\|_2$ e o substitui para obter os dados comprimidos:

$$\mathbf{y}' = \mathbf{y} - \sum_{i=0}^j \hat{c}_i \cdot \phi_{P_i}, b_i. \quad (2.41)$$

4º Passo Repetir os passos acima até que $\|\mathbf{y}'\| < \epsilon$.

Com esse método básico encontra-se um problema quando duas componentes tem P_s com uma coluna em comum. Se isso ocorrer, duas funções Walsh com a mesma frequência são encontradas em (2.40). Elas podem se somar ou se cancelar, e no caso de cancelamento isso pode gerar uma coluna perdida.

A carga computacional desse algoritmo não é dependente do comprimento n do sinal para a recuperação, mas depende do número de medidas n .

Baron et al. [Baron et al. 2005] propôs uma regra de acordo com o estudo de Donoho e Tanner [Donoho e Tanner 2005],[Donoho 2006] da probabilidade de uma recuperação eficiente através de BP. Isso é que o número de medidas n é essencial

para uma reconstrução confiável, e isso satisfaz $n > k \log_2(1 + \frac{N}{k})$. Entretanto, a única mudança requerida para o algoritmo é o critério de parada.

2.7.3 Aproximação de Amini para matrizes que satisfazem RIP

A condição RIP no *compressive sensing* é um requerimento essencial que garante a recuperação de um vetor de sinal esparsos em uma condição de ruído. Entretanto, existe um problema da confirmação dessa propriedade para um dado conjunto de detecção: o problema de natureza NP-hard, que requer otimização metaheurística evolucionária [Khosravy et al. 2020][Khosravy et al. 2020]. Como resultado, a matriz mais popular que cumpre RIP é de estrutura aleatória.

Amini e Marvasiti fizeram uma aproximação de como fazer uma matriz que satisfaz RIP como explicado em [Amini e Marvasti 2011]. Considerando $A_{m \times n}$ como uma matriz real composta por colunas normalizadas de tal maneira que o produto escalar de duas colunas não superem um certo valor λ . E com $B_{m \times k}$ uma matriz feita de quaisquer k colunas de A . Então tem-se a matriz de Gramian $G_{k \times k} = B^T B$. Como as colunas de B são normalizadas, os valores na diagonal de G são 1, e os outros elementos têm valores absolutos menores que λ . Por isso tem-se a equação 2.42.

$$\forall 1 \leq i \leq k : \sum_{j, j \neq i} |G_{i,j}| \leq \lambda \cdot (k - 1) \quad (2.42)$$

onde $G_{i,j}$ são os elementos de G . Por meio do teorema circular de Gershgorin, os autovalores de G são limitados a uma faixa de $[1 - \lambda(k - 1), 1 + \lambda(k - 1)]$. Portanto, existe suficientemente pequenos k de maneira que $\sigma_k = \lambda(k - 1) < 1$, e a matriz $A_{m \times n}$ satisfaz a condição RIP. A aproximação de Amini-Marvasti para satisfazer essa condição pode ser resumida a seguir.

Aproximação de Amini-Marvasti para satisfazer a propriedade RIP: Uma matriz de CS A composta por colunas normalizadas \mathbf{a}_i s e de tamanho $m \times n$ onde $\forall i, j \neq i, \langle \mathbf{a}_i, \mathbf{a}_j \rangle \leq \lambda$ tem RIP de ordem k se:

$$\lambda < \frac{1}{k - 1} \quad (2.43)$$

2.7.4 Matrizes binárias Amini-Marvasti

Amini e Marvasti introduziram matrizes binárias que satisfazem RIP [Amini e Marvasti 2011]. Uma matriz de amostragem binária que satisfaz a condição de RIP têm os elementos 0 e 1 antes de suas colunas serem normalizadas. Um subconjunto dessas matrizes tem sido usado anteriormente como códigos óticos ortogonais (OOC). Considerando um vetor binário composto de 0 e 1, e o peso w como o número 1. Se $R(m, w, \lambda)$ representa o número máximo de vetores $m \times 1$ para qual seus pesos são w e o produto interno de quaisquer dois deles não é maior que λ ($\lambda \in Z$), então temos a equação 2.44.

$$R(m, w, \lambda) \leq \left[\frac{m}{w} \cdot \left[\frac{m-1}{w-1} \cdot \left[\dots \left[\frac{m-\lambda}{w-\lambda} \right] \dots \right] \right] \right] \quad (2.44)$$

Onde $[\cdot]$ representa o maior número inteiro que não é maior que o operador. Na aplicação desses códigos em telecomunicações multi-acessos, o menor valor do produto interno de diferentes símbolos é a chave para diferenciar entre os usuários quando os símbolos são sincronizados; e quando eles não são sincronizados, a autocorrelação e a correlação cruzadas possui esse papel importante [Amini e Marvasti 2011]. Portanto ao invés de λ , dois parâmetros λ_a e λ_c são usados, e eles são respectivamente a máxima auto correlação e a máxima correlação cruzada.

Alternativamente, existe uma propriedade calculável rápida para que a matriz de compressão seja a coerência mútua $\mu(\Phi)$. Para uma matriz $\Phi_{m \times M}$, a coerência mútua é o máximo absoluto do produto interno de quaisquer duas colunas normalizadas de $\Phi_{m \times M}$.

$$\mu(\Phi) = \max_{1 \leq i, j \leq M, i \neq j} \left[\frac{|\phi_i^T \cdot \phi_j|}{[\|\phi_i\|]_2 \cdot [\|\phi_j\|]_2} \right] \quad (2.45)$$

onde ϕ_i é a i -ésima coluna de Φ . A proposição 1 relaciona δ_s com $\mu(\Phi)$. [Bourgain et al. 2011]

Proposição 1. *Suponha que $\phi_1, \phi_2, \dots, \phi_M$ são colunas da matriz Φ . Eles têm norma unitária, e Φ tem coerência μ . Então, Φ cumpre o RIP de ordem s com $\delta_s = (s-1) \cdot \mu$ constante, quando $\delta_s < 1$.*

O resultado dessa proposição é que podem-se aproximar projetos de matrizes sensoriais construindo elas com baixa coerência mútua. Na área de construção de matrizes binárias de detecção, existem técnicas variantes, elas usam álgebra, teoria de gráficos e teoria de codificação. Nesta seção serão discutidas três técnicas que são: Código BCH, Teoria dos conjuntos extremos e Expansão de gráficos desbalanceados.

2.7.4.1 Código BCH

Os códigos de BCH, (Bose - Chaudhuri - Hocquenghem) fazem dois conjuntos de algoritmos para correção cíclica de erros. Eles são feitos utilizando polinômios em um campo finito (campo de Galois).[Bose e Ray-Chaudhuri 1960] [Amini e Marvasti 2011]

O principal destaque desses códigos é que por todo o projeto, existe um alto nível de controle de precisão sobre o código de erro de frequência, e como isso é corrigível por ele. É possível criar esses códigos com a capacidade de correção de erros de bit. Eles também têm a vantagem de decodificar através de um método algébrico.

Para a aplicação dos códigos BCH em matrizes binárias destinadas ao CS pode-se usar vetores OOC caracterizados por $m, w, \lambda_a, \lambda_c$. E existe a possibilidade que os parâmetros λ_a e λ_c sejam iguais ($\lambda_a = \lambda_c = \lambda$). Nesse caso o OOC é apontado como código (m, w, λ) .

Seja A um conjunto de vetores OOC de comprimento m . Com peso w e parâmetros iguais ($\lambda_a = \lambda_c = \lambda$). Aplicando as determinações de OOC é observado que o produto interno de quaisquer dois vetores de A é restrito por λ . Então é possível construir uma matriz $A_{m \times n}$ através das colunas normalizadas em A onde $n = |A|$. Sobre o limite superior do produto interno de vetores em A é fácil de verificar que a matriz A satisfaz a RIP de ordem $k < 1 + \frac{w}{\lambda}$. [Ding e Xing 2003]

Considerando $q = 16^a$ onde $a \in \mathbb{N}$. $F = GF(q)$ como α é a raiz primitiva. Embora [Amini e Marvasti 2011] considere a relação de $5|q - 1$, indicando que 5 é um fator de $q - 1$ como um assunto claro. Isso pode ser provado no Lema a seguir.

Lema 1

$$\forall a \in N, 5|(16^a - 1), \quad (2.46)$$

Onde N é o conjunto dos números naturais, e $|$ indica "sendo fator de".

Prova A prova deste Lema é através de indução matemática. Como a afirmação é para um número natural a , primeiro mostramos que é verdade para $a = 1$, depois mostramos que se a afirmação é verdadeira para $a = k$ então é verdade para $a = k + 1$ onde k é um número particular mas é um número arbitrário natural. Matematicamente provamos que:

$$a = 1 \longrightarrow 5|(16^a - 1), \quad (2.47)$$

e

$$k \in N \implies 5|(16^k - 1) \longrightarrow 5|(16^{k+1} - 1) \quad (2.48)$$

A Equação 2.47 é facilmente vista quando substituimos $a = 1$, o resultado é $5|15$, que é verdade. Já para provar a Equação 2.48, se $5|(16^k - 1)$, assim claramente $5|(16 \times (16^k - 1))$, pode ser reescrita como:

$$\begin{aligned} & 5|(16^{k+1} - 16) \\ & 5|(16^{k+1} - 15 - 1) \\ & 5|(16^{k+1} - 1) \end{aligned} \quad (2.49)$$

E assim o Lema é provado.

Como $5|(16^a - 1)$, existe um inteiro d onde $q = 5d + 1$. Definindo:

$$D_i = \{\alpha^{d+1}, \alpha^{2 \cdot d+1}, \dots, \alpha^{5 \cdot d+1}\}, 0 \leq i \leq d - 1. \quad (2.50)$$

Porque a maioria dos casos o número 1 é consideravelmente menor que o número 0, os vetores OOC são comumente representados pela localização do 1. Nessa configuração, o comprimento do código é $q - 1$, o que significa que $m = 16^a - 1$ e as localizações dos 1 em cada código é dada por $C_i^{d-1}_{i=1}$ [Amini e Marvasti 2011]:

$$C_i = \log a(D_i - 1), 1 \leq i \leq d - 1. \quad (2.51)$$

Uma matriz com estrutura independente dos códigos OOC é dado em [DeVore 2007]. É feita de uma matriz binária $p^2 \times p^{r+1}$. As colunas tem pesos p (antes da normalização). O produto interno de quaisquer duas colunas não é maior que r , e depois da normalização não se tornam maiores que $\frac{r}{p}$. Onde p é uma potência de um número primo. A estrutura da matriz é baseada em polinômios em $GF(p)$. Essas matrizes são assintoticamente ótimas quando $\frac{p}{r^2} \rightarrow \infty$ [Amini e Marvasti 2011] :

$$\begin{aligned} \lim_{\frac{p}{r^2} \rightarrow \infty} \frac{p^{r+1}}{R(p^2, p, r)} &\geq \lim_{\frac{p}{r^2} \rightarrow \infty} \prod_{i=0}^r \frac{p \cdot (p - i)}{p^2 - i} \\ &= \lim_{\frac{p}{r^2} \rightarrow \infty} \frac{p - r^{r+1}}{p} \\ \lim_{\frac{p}{r^2} \rightarrow \infty} e^{-\frac{r(r+1)}{p}} &= e^0 = 1. \end{aligned} \quad (2.52)$$

Tendo $C(\tilde{n}, ; 2)$ como um bloco de código linear e $1_{\tilde{n} \times 1}$ um vetor de coluna unitária de $\tilde{n} \times 1$. Se $1_{\tilde{n} \times 1} \in C$ então C é simétrico. Portanto para códigos simétricos podem-se definir $a_{n \times 1}$ como vetor código, e por isso através da linearidade do código o complemento de $a_{n \times 1}$ é definido como $a_{n \times 1} \oplus 1_{\tilde{n} \times 1}$ e é um código de vetor válido também. Então os vetores são feitos de pares complementares.

O Teorema 1 de [Amini e Marvasti 2011] dá uma matriz binária de CS. Isso considera $C(\tilde{n}, ; 2)$ como um código simétrico de mínima distância \tilde{d}_{min} . Então isso compõe a matriz $\tilde{A}_{\tilde{n} \times 2^k - 1}$ através dos códigos nas colunas. A matriz envolve exatamente um de cada casal complementar. Então define:

$$A_{\tilde{n} \times 2^k - 1} \triangleq \frac{1}{\sqrt{\tilde{n}}} \left[2 \cdot A_{\tilde{n} \times 2^k - 1} - (1)_{\tilde{n} \times 2^k - 1} \right] \quad (2.53)$$

A matriz definida acima cumpre a RIP com a constante $\delta_k = (k - 1)(1 - 2 \cdot \frac{\tilde{d}_{min}}{\tilde{n}})$ para $k < \frac{\tilde{n}}{\tilde{n} - 2 \cdot \tilde{d}_{min}}$. Onde k é a ordem de RIP.

A prova do Teorema 1 pode ser encontrado em [Amini e Marvasti 2011]. Ele diz que \tilde{d}_{min} deve ser próximo do valor de $\frac{\tilde{n}}{2}$, normalmente não é comum isso nos sistemas binários devido ao envio de bits de paridade para proteger os dados enviados. Mas usando o código BCH pode-se fazer esse tipo de algoritmo.

É importante lembrar que o código BCH está na classe de códigos cíclicos tal que $\tilde{n} = 2^{\tilde{m}-1} + 1$ feito por um $g(x) \in GF(2)[x]$ respeitando $g(x) | x^{2^{\tilde{m}-1}} + 1$ [Lin 2004]. Usando a teoria de Galois, temos:

$$x^{2^{\tilde{m}-1}} + 1 = \prod_{\substack{r \neq 0 \\ r \in GF(2^{\tilde{m}})}} (x - r) \quad (2.54)$$

Dessa maneira podem-se decompor o polinômio gerador BCH em produtos dos fatores lineares de $GF(2^{\tilde{m}})$. Portanto seja $\alpha \in GF(2^{\tilde{m}})$ uma raiz primitiva do campo e α^i a raiz de $g(x)$. Sendo $g(x) \in GF(2)[x]$ todos os elementos conjugados de α^i então todos serão também raízes de $g(x)$. E por isso, se $\alpha^{i_1}, \alpha^{i_2}, \dots, \alpha^{i_d}$ são diferentes raízes de $g(x)$, tal que i_1, i_2, \dots, i_d forme uma progressão geométrica, então $d_{min} \geq d + 1$.

Deve-se então fazer um polinômio $h(x)$ para checagem de paridade. Então, sendo $l < \tilde{m} - 1$ tem-se a equação 2.55.

$$G_{\tilde{m}}^{(l)} = \{\alpha^0, \alpha^1, \dots, \alpha^{2^{\tilde{m}-1} + 2^l - 1}\} \quad (2.55)$$

Em [Amini e Marvasti 2011] tem-se a maneira de construir a matriz compressiva seguindo os seguintes passos:

1º Passo Enquanto k é dado, $i = \lceil \log_2(k) \rceil$ e obtém $\tilde{m} \geq i$. O comprimento do vetor do sinal comprimido será $m = 2^{\tilde{m}-1}$.

2º Passo Considere H_{seq} como o conjunto de todos vetores binários de comprimento \tilde{m} tendo pelo menos i zeros. Considere H_{dec} como um conjunto de números decimais. A representação binária correspondente de H_{dec} está contida em H_{seq} .

3º Passo Seja α a raiz primitiva de $GF(2^{\tilde{m}})$. E então defina:

$$H = \{\alpha^r | r \in H_{dec}\}. \quad (2.56)$$

Determine a checagem de paridade. E defina o código produzido polinômios:

$$h(x) = \prod_{r \in H} (x - r) \quad (2.57)$$

e

$$g(x) = \frac{x^{2^{\tilde{m}}} - 1}{h(x)} \quad (2.58)$$

4º Passo Considere $\tilde{A}_{(2^{\tilde{m}}-1) \times 2^{\deg(h)-1}}$ como uma matriz binária feita de códigos pares em colunas. Enquanto as colunas são coeficientes polinomiais ($\epsilon GF(2)[x]$), cada polinômio será dividido por $(x+1).g(x)$.

5º Passo Coloque -1 ao invés dos elementos nulos de A . E então normalize cada coluna. Nesse estágio a matriz de compressive sensing $\tilde{A}_{(2^{\tilde{m}}-1) \times 2^{\deg(h)-1}}$ é obtida.

2.7.4.2 Teoria dos conjuntos extremos

A teoria dos conjuntos extremos é uma teoria de conjuntos. Seja r, k e m como inteiros positivos de forma que $r < k < m$. Seja X um conjunto de m elementos, tal que $X = \{1, 2, \dots, m\}$. E também sejam as seguintes definições dadas por [Naidu e Murthy 2016]:

- $2^X := \{H, H \subseteq X\}$,
- $[X]^k := \{H \subseteq X, |H| = k\}$.

O subconjunto $F \subset [X]^k$ é chamado de família k -uniforme. Assim temos as seguintes definições:

Definição 1. O subconjunto $F_d(r, k, m) \subset [X]^k$ é r -denso se qualquer subconjunto de X com r elementos tiver pelo menos um pedaço de F_d

Definição 2. $F_s(r, k, m) \subset [X]^k$ é r -esparso se qualquer subconjunto de X com r elementos tiver no máximo um pedaço de F_s , tal que, $|F_i \cap F_j| \leq r - 1, \forall F_i, F_j \in F_s$.

Definição 3. Um subconjunto $F_s(r, k, m) \subset [X]^k$ é um sistema Steiner se cada subconjunto de X com r elementos cabe perfeitamente em um dos pedaços de F_s

Devemos considerar uma condição de divisibilidade vista em [Keevash 2014] dada por:

$$\forall 0 \leq i \leq r-1 \binom{k-1}{r-i} \text{ divide } \binom{m-1}{r-i} \quad (2.59)$$

Desse modo o sistema Stein $F_S(r, k, m)$ é um subconjunto de um conjunto r -esparso $F_s(r, k, m)$. Para aplicar a Teoria dos conjuntos extremos, devem-se verificar a existência da matriz com tamanho da coluna otimizado, para cada par de (r, k) onde $r < k$. Para atingir o mesmo objetivo da seguinte proposição [Naidu e Murthy 2016] relaciona a esparsidade r dos conjuntos e a matriz de detecção binária com estrutura $(r-1, k)$.

Proposição 1. Suponha que ϕ_1, \dots, ϕ_M sejam colunas formadas por normas unitárias da matriz Φ com coerência μ . Então, Φ satisfaz a RIP de ordem s com constante $\delta_s = (s-1)\mu$, quando $\delta_s < 1$. [Naidu e Murthy 2016]

Proposição 2. Existe uma bijeção um pra um dentro do conjunto contendo as famílias r -esparsas e k -uniformes e o conjunto de matrizes de detecção binárias de estrutura $(r-1, k)$. [Naidu e Murthy 2016]

Proposição 3. Seja F uma família r -esparsa k -uniforme de cardinalidade M no m elemento do conjunto X . Então $\Phi_{m \times M}$, ou seja, a matriz de incidência de F , tem no máximo a coerência de $\frac{r-1}{k}$. Também, $\Phi = \frac{1}{\sqrt{k}} \cdot \Phi$ cumpre a RIP com $\delta_s = (s-1) \cdot \binom{r-1}{k}$, para cada $s < \frac{k}{r-1} + 1$. [Naidu e Murthy 2016]

O processo para construir uma família 2-esparsa e k uniforme de um conjunto $X = \{0, 1, 2, \dots, kp-1\}$ usando polinômios infinitos $F_p = \{f_1 = 0, f_2, \dots, f_p\}$, enquanto p é uma potência primária, e $2 \leq k \leq p-1$, pode ser resumido desse modo.

Proposição 4. Seja F_p um campo finito de ordem p . Seja $\Gamma(x)$ um conjunto de polinômios com grau máximo de $\frac{1}{F_p}$. Então, o conjunto F_Γ formado acima é uma família 2-esparso e k uniforme $(2, k, kp)$ em X . E sua cardinalidade é p^2 . [Naidu e Murthy 2016]

Através da junção das proposições 2 e 4 temos que a incidência da matriz F_Γ tem a $(2, k)$ -estrutura e tamanho de $(kp \times p^2)$. Além disso pode-se observar que qualquer $(r-1, k)$ -estrutura pertence ao conjunto de famílias k -uniformes e

r-esparsas no conjunto X com cardinalidade m .

Finalmente, usando os resultados da Teoria dos conjuntos extremos para um caso r-esparso deve-se limitar o tamanho máximo de cada possível coluna com essas matrizes. Esse resultado foi provado por Katona e Nemetz.[Brown e Erdős 1984]

Proposição 5. Seja $F_s \subseteq [X]^k$ e F_s uma família r-esparsa. Então:

$$|F_s| \leq \frac{\binom{m}{r}}{\binom{k}{r}} \quad (2.60)$$

Portanto, o tamanho da coluna da matriz em construção de $(r-1, k)$ -estrutura é limitada por $\frac{\binom{m}{r}}{\binom{k}{r}}$ onde m é o número de linhas na estrutura, e k é o número de 1 nas colunas, e $r-1$ representa o produto interno entre duas colunas.

O teorema para a existência de tal matriz de detecção com o tamanho da coluna otimizado é dado abaixo.

Teorema 1.

$$\forall(r, k), r < k, \lim_{m \rightarrow \infty} n(m, k, r) \cdot \frac{\binom{k}{r}}{\binom{m}{r}} = 1 \quad (2.61)$$

onde $n(m, k, r)$ indica a máxima cardinalidade possível de F_s .

Portanto, para um m suficientemente grande, usando a construção de Rödl [Rödl 1985], é possível a detecção de uma estrutura (r, k) -binária.

2.7.4.3 Expansão de gráficos desbalanceados

Em combinatória a Expansão de gráficos é um gráfico esparso com fortes propriedades de conectividade são quantificados usando vértices, bordas ou expansão

espectral. Essas construções expansivas geram pesquisas em várias áreas tais como teoria da complexidade, design de redes de computadores que é robusto, e teoria de códigos de correção de erros [Hoory e Linial 2006]. É similar aos códigos BCH, e nessa subseção será mostrado como utilizara expansão de gráficos desbalanceados para gerar uma matriz binária de amostragem.

Para isso seja, $\|x\|_0$ define o número de entradas diferentes de zero do sinal x , e a notação $\{n\} = \{1, \dots, n\}$ é usada para qualquer gráfico G formado pelos vértices V e $v \in V$. Os conjuntos vizinhos de v serão representados em G por $\Gamma_G(v)$ e o conjunto incluindo as bordas de v em X por $\Gamma(v, X) = \{v \in V, X \subset V\}$ [Indyk 2008].

Seja $A = \{n\}$ e $B = \{k\}$. Tem-se um gráfico bipartido $G = (A, B, E)$. E d_A é o grau dos nós em A . No gráfico adquirido, o ϵ -extrator [Shaltiel 2004] $\forall \epsilon > 0$ será especificado depois. Portanto $\forall X \subset A, |X| \geq k$ considerando D uma distribuição de B que é obtido de uma aquisição aleatória de $a \in X$ e $b \in \Gamma(a)$. Então:

$$\sum_{b \in B} |Pr_D(b) - \frac{1}{|B|}| \leq \epsilon \quad (2.62)$$

Para todo $\epsilon > 0$, deve-se construir um ϵ -extrator tal que $d_A = 2^{O.(\log \log n)^E}$, $\forall E > 1$. Para as demonstrações será usado $E = 2$ [Indyk 2008].

Já para $t \in N$, definimos:

$$overflow_t(X) = \{b \in B, |\Gamma(b, X)| > t\}. \quad (2.63)$$

Escolhendo $t = 2.d_A$, percebemos que $\forall b \in overflow_t(X)$, e temos $Pr_D(b) \geq \frac{2}{|B|}$. E isso significa que:

$$|overflow_t(X)| \leq \epsilon . |B| \quad (2.64)$$

Então seja H um gráfico da matriz adjacente a G . Considere $X = \{i \in \{n\} : x_i \neq 0\}$ onde $|X| \leq r$. Usa-se as propriedades de G [Indyk 2008] para qualquer $i \in overflow_t(X)$, o valor de $(Hx)_i$ é a soma do máximo dos elementos t de X que não sejam zeros. Usa-se os índices dessas entradas definidos como $\{j_1^i, j_2^i, \dots, j_t^i\}$ pode-se recuperar o conjunto $J_i = \{j_1^i, \dots, j_t^i\}$, e assim fica claro que i pertence a $overflow_t(X)$. Para melhorar a recuperação deve-se aumentar as medidas de

H para uma medida maior de H' , assim tendo $H'x$, será possível recuperar J_i , $\forall i \in \text{overflow}_t(X)$.

Para isso, pode-se usar a noção de produto tensorial de linha [Cormode e Muthukrishnan 2006]

Definição 4. Tendo matrizes $V_{v \times n}$ e $W_{w \times n}$, então o produto $V \otimes W$ é uma matriz $v.w \times n$ de forma que $(V_{v \times n} \text{ e } W_{w \times n})_{i.v+l,j} = V_{i,j}.W_{l,j}$, onde \otimes é um produto tensorial.

Na verdade, $(V_{v \times n} \text{ e } W_{w \times n})$ é feito através da cópia de cada linha de V a w vezes e implementando a multiplicação coordenada de cada cópia com a linha correspondente de W. Assim pode-se definir $H' = (H \otimes B)$ que garante que pode-se recuperar o conjunto J_i e a sequência U_i para todo i fora do $\text{Overflow}_t(X)$ [Indyk 2008].

Então para recuperar x a partir de J_i tem-se que seguir os seguintes passos:

1. Deve-se construir um vetor y que aproxima bem de perto de x e então $\|x - y\|_0 = O(\epsilon.k)$.
2. Aplicar recursivamente a redução do parâmetro de esparsidade k para recuperar completamente x .

Os algoritmos utilizados nos itens 1 e 2 são chamados REDUÇÃO e RECUPERAÇÃO, respectivamente, e eles podem ser encontrados em mais detalhes em [Indyk 2008].

2.7.5 Matrizes com RIP estatístico

Em CS a matriz de amostragem $\Phi N \times C$ a princípio possui isometria nos conjuntos de k sinais esparsos. Isso garante a propriedade de isometria restrita (RIP), se ϕ garante RIP, então os algoritmos *basis pursuit* ou *matching pursuit* são aplicáveis para a recuperação de vetores k -esparsos α de N medidas Φ_a .

Alguns métodos probabilísticos fazem matrizes $N \times C$ que cumprem RIP. Entretanto existe uma falta de algoritmos práticos que pesquisam se a matriz Φ possui RIP. E isso é crucial para a construção de algoritmos de recuperação. Mas na referência [Calderbank e Howard 2010] tem várias regras simples para matrizes determinísticas que cumprem a RIP.

Candès e Tao definiram RIP da seguinte maneira: [Candes e Tao 2006] uma matriz que atende a uma k -propriedade de isometria restrita se sua performance na isometria é para todos os k -esparsos vetores.

Ambos algoritmos BP e MP dependem da multiplicação da matriz e do vetor. Eles são super lineares quanto a dimensão dos dados. O algoritmo de recuperação de códigos de Reed-Muller que será mostrado, precisa de multiplicação de vetor a vetor.

Pode-se comparar CS com vários códigos estruturados da prática da comunicação, pois ambos precisam de uma rápida encriptação e desencriptação.

Quando trabalha-se com matrizes determinísticas, faz sentido trabalhar com uma RIP estatística mais fraca, então será feita a definição a seguir.

Uma matriz Φ de $N \times C$ tem (k, ϵ, γ) -estatística RIP, se ela é como a equação 2.65.

$$\forall k\text{-esparsos } \alpha \in \mathfrak{R}^C (1 - \epsilon) \|\alpha\|^2 \leq \left\| \frac{\Phi \alpha}{\sqrt{N}} \right\|^2 \leq (1 + \epsilon) \|\alpha\|^2 \quad (2.65)$$

Considerando a probabilidade maior que $1 - \delta$, relativo a uma distribuição uniforme para vetores α k -esparsos em \mathfrak{R}^C [Calderbank e Howard 2010].

Uma matriz Φ $N \times C$ é uma (k, ϵ, δ) -singularidade-garantida estatisticamente RIP, se Φ tem (k, ϵ, δ) -StRIP, e tem-se a equação 2.66.

$$\{\beta \in \mathfrak{R}^C : \Phi \cdot \alpha = \Phi \cdot \beta\} = \{\alpha\} \quad (2.66)$$

Com probabilidade maior que $1 - \delta$, relativo a uma distribuição uniforme para vetores k -esparsos α em \mathfrak{R}^C da mesma norma [Calderbank e Howard 2010].

Sobre esse tema, existem regras fáceis. As colunas da matriz de detecção fazem uma coleção sob multiplicação pontual em que todas as somas das linhas desaparecem. Diferentes linhas são ortogonais ao limite superior do valor absoluto de qualquer soma de colunas.

No artigo [Calderbank e Howard 2010] encontra-se definições, lemmas e explicações sobre esse RIP mais fraco e suas aplicações em: Matrizes de detecção de

Chirp discretas, Kerdock, Delsarte-Goethals, Matrizes de detecção de Reed-Muller e BCH.

Com o critério de um RIP mais fraco, conforme preenchido pela matriz, só garante a recuperação do sinal; Só a pequena parte exponencial dos sinais k -esparsos podem não ser recuperados. Esse critério é realizado por várias famílias de matrizes determinísticas e isso inclui os subcódigos dos códigos binários de segunda ordem de Reed-Muller. E também se aplica aos conjuntos aleatórios de Fourier.

2.7.6 Matrizes determinísticas de curvas algébricas através de campos finitos

Essas matrizes são motivadas pelos códigos de geometria algébricos. É uma novidade a estrutura determinística através de curvas algébricas através de campos finitos. Uma curva algébrica real é um conjunto de pontos no plano Euclidiano. Suas coordenadas são zeros de alguns polinômios com duas variáveis. Eles têm uma diversidade que nos dá escolha ao construir uma matriz de detecção.

Ao escolher as curvas apropriadas, pode-se criar uma matriz de detecção binária preferido a matrizes de DeVore[Li et al. 2012].

Em [DeVore 2007], DeVore usa polinômios em campos finitos \mathcal{F}_p para construir $p^2 \times p^{r+1}$ matrizes de detecção binárias, onde p é a potência principal. A matriz construída tem a coerência $\frac{r}{p}$. Isso satisfaz o RIP de ordem $k < \frac{r}{p} + 1$. Essa aproximação pode ser utilizada para qualquer campo finito \mathcal{F}_p . Cada polinômio produz um aumento na coluna da matriz. Lembre que todo polinômio desenvolve uma palavra-chave de Reed-Solomon.

Através das curvas algébricas e seus campos de função, existe um grande potencial para se fazer matrizes de CS determinísticas.

2.7.7 Matrizes construídas a partir de curvas algébricas sobre campos finitos

Essas matrizes de CS são motivadas pelos códigos algébricos geométricos. É uma nova estrutura determinística através de curvas algébricas sobre campos finitos. Uma curva algébrica real é um conjunto de pontos no plano Euclidiano. Suas coordenadas são zeros de algum polinômio com duas variáveis. Eles têm uma diversidade que dá a escolha de construir matrizes de detecção.

Através da escolha correta de curvas, pode-se criar uma matriz de detecção binária preferida às matrizes de Devore.

Em [DeVore 2007], Devore emprega polinômios em um campo finito F_p para a construção de matrizes de detecção binárias $p^2 \times p^{(r+1)}$, onde p é uma potência e prima. A matriz construída tem uma coerência $\frac{r}{p}$. E satisfaz o RIP de ordem $k < \frac{p}{r} + 1$. Essa aproximação pode ser utilizada para qualquer campo finito F_p . Cada polinômio produz um aumento na coluna da matriz. E cada polinômio desenvolve uma palavra chave de Reed-Solomon.

Através de curvas algébricas e seus campos de funções, existe um grande potencial para se fazer matrizes determinísticas de CS.

3 PROPOSTA DA METODOLOGIA

Compressive Sensing é um método amplamente utilizado em diversas aplicações como foi visto anteriormente. O método escolhido para ser estudo deste trabalho são os códigos de *Chirp*. O presente capítulo apresenta a metodologia aplicada em sinais sintéticos de potência, bem como alguns códigos MATLAB[®] utilizados para a análise da metodologia, esses códigos estão presentes no Anexo 6.1.

3.1 Apresentação da metodologia

O método de códigos de *Chirp* é uma aproximação determinística. Para um código de *chirp* de comprimento K , existem K^2 possibilidades de códigos de *chirp*, ou seja, o código tem um comprimento K , e existem K^2 possibilidades de formar diferentes códigos com esse comprimento e a matriz de detecção é composta de códigos de *chirp* de comprimento K em K^2 colunas.

Resumidamente, sabe-se que no problema geral de CS, um sinal de comprimento n em um vetor \mathbf{x} no espaço vetorial R^n é comprimido na detecção para um vetor de comprimento m , \mathbf{y} , no espaço vetorial R^m , através de m combinações lineares de n amostras do vetor do sinal, ou seja, o sinal de n amostras é comprimido em um sinal com m amostras.

Em CS determinístico utilizando códigos *chirps*, tem-se a matriz A que é composta dos códigos de *chirp* em suas colunas, e sua dimensão é $K \times K^2$. O desempenho e a capacidade da técnica depende das características desses códigos.

Na próxima seção será implementado um código de *chirp* no MATLAB[®].

3.2 Códigos de Chirp

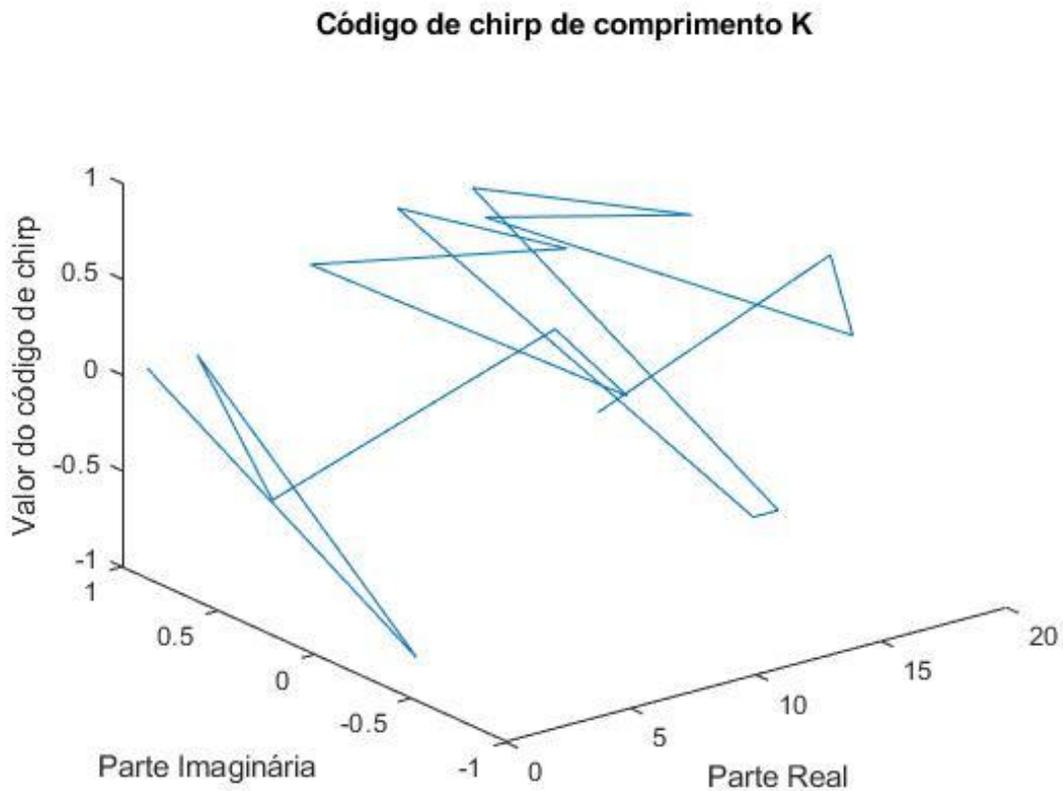
Um código de *chirp* é na verdade um vetor de comprimento K . No que se refere a frequência ele tem dois aspectos. O primeiro que é uma parte periódica de tom único, e isso é chamado frequência de base. E o outro que é um tom passando todas as frequências, e isso é chamado tom de *chirp*. As duas frequências são multiplicadas entre si. O código de *chirp* de comprimento K é dado na equação 3.1.

$$\mathbf{v}_{m,r}(l) = \alpha \cdot \exp(j2\pi \frac{ml}{K} + j2\pi \frac{rl^2}{K}), l, m, r \in Z_K \quad (3.1)$$

em que $\exp(j2\pi \frac{ml}{K})$ é a componente multiplicativa de um tom e $\exp(j2\pi \frac{rl^2}{K})$ é a componente de tom de *chirp*. A primeira componente contém a frequência base m , e a componente de *chirp* contém a taxa de *chirp* r , que é a taxa de *chirp* entre as frequências.

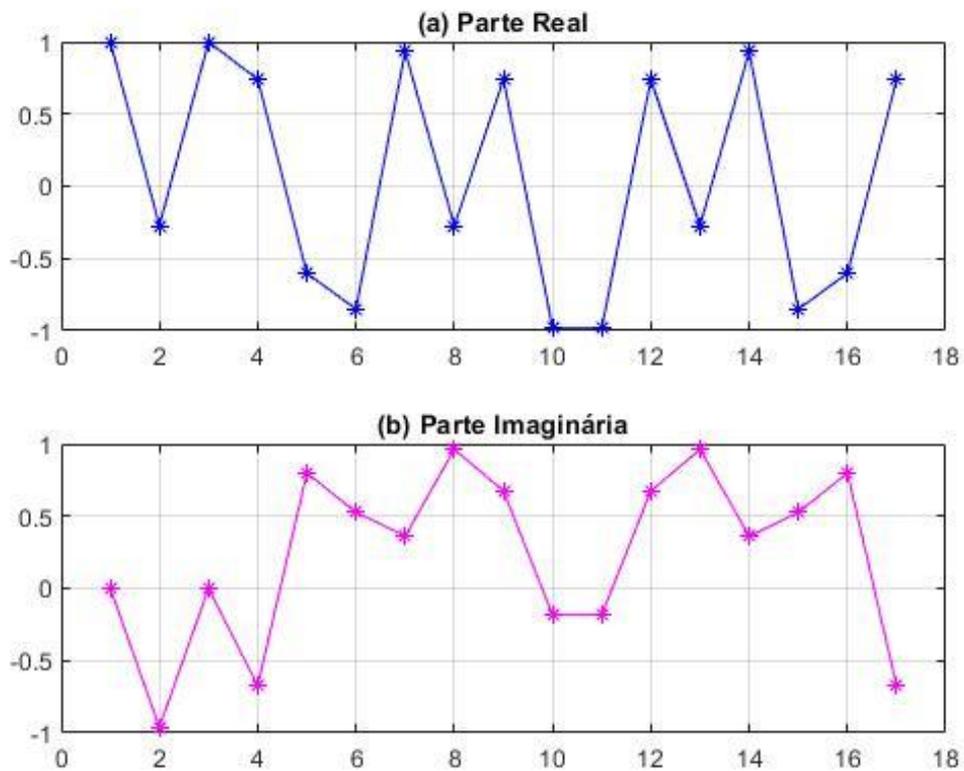
Como visto na Figura 1, ele é uma sequência de números complexos. As Figuras 2a e 2b mostram as partes real e imaginária da 1, ou seja do mesmo código de *chirp*.

Figura 1 – Código de *Chirp* como uma sequência de número complexo



Fonte: Elaborado pelo autor.

Figura 2 – Parte real e imaginária do código de *Chirp* da imagem 2.



Fonte: Elaborado pelo autor.

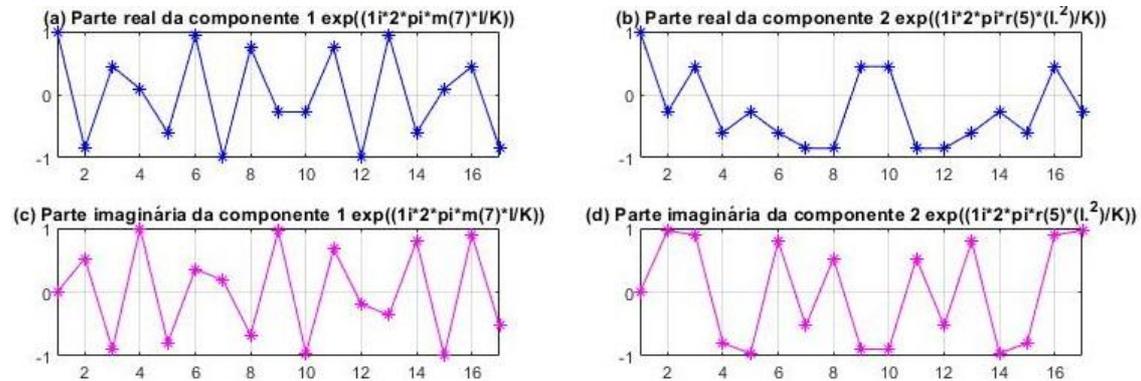
O código de MATLAB ®implementou a Equação 3.1, e o resultado do código é mostrado na Figura 3.

As componentes multiplicativas do mesmo código de *chirp* são:

- $e^{j2\pi\frac{ml}{K}}$
- $e^{j2\pi\frac{rl^2}{K}}$

As figuras referentes ao código são as Figuras 3, onde as Figuras 3.A e Figura 3.B são as partes reais das componentes e as Figura 3.C e Figura 3.D são as partes imaginárias. Com isso pode-se perceber a parte principal da frequência base e da frequência de *chirp* e que não existem diferenças significativas entre elas.

Figura 3 – Componentes multiplicativas do código de *chirp* parte real (A e B) e imaginária (C e D) 3.

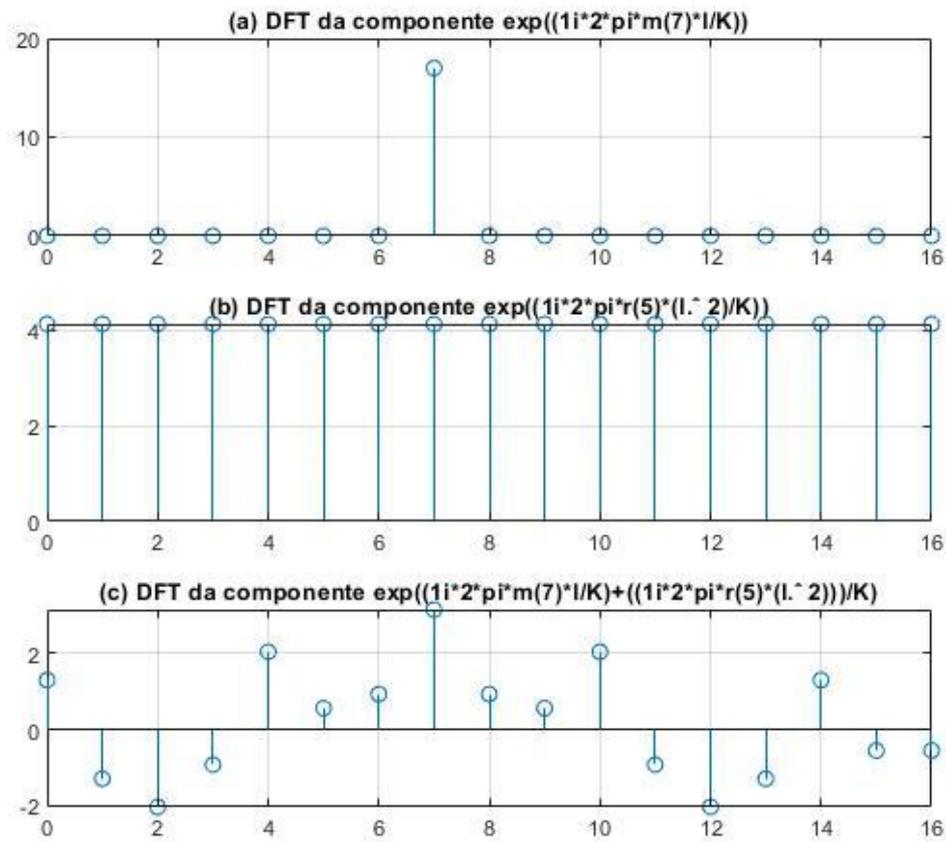


Fonte: Elaborado pelo autor.

Ou seja, no processo de detecção a diferença entre elas não é relevante. E isso é esperado, pois como vimos anteriormente a forma de onda que as formam é muito próxima.

Foi observado como a Transformada Discreta de Fourier (DFT) dessas componentes se comportam. A Figura 4 mostra que a energia da parte com a frequência de *chirp* está espalhada ao longo do espectro e a componente da frequência de base tem sua energia concentrada. Entretanto o código de *chirp* que é composto por ambas frequências, possui um espectro com a energia espalhada. Embora o pico de energia possa ser observado em m_i , deve-se lembrar que esse sinal de *chirp* é somente um, e no caso de termos uma combinação linear de códigos de chirp, a energia espalhada pelo espectro ficará mais clara.

Figura 4 – DFT das componentes do código de *chirp* sendo (A) DFT da componente da frequência de base (B) DFT da componente da frequência de *chirp* e (C) DFT de ambas as componentes⁴.



Fonte: Elaborado pelo autor.

3.3 Aproximação para extração dos parâmetros

O vetor \mathbf{f} tem um papel fundamental na recuperação de sinais comprimidos utilizando CS com códigos de *chirp*, ele pode ser visto na equação 3.2. Com isso podemos ver como ele auxilia a extrair a frequência de base m_i a partir de um código de *chirp*.

$$\mathbf{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_l \\ \vdots \\ f_K \end{bmatrix}. \quad (3.2)$$

Onde f_l é visto na equação 3.3.

$$f_l = \bar{y}_1^* y_1 + T = |x_1|^2 e^{\frac{j2\pi}{K}(m_1 T + r_1 T^2)} e^{j2\pi \frac{2r_1 T}{K} l} + |x_2|^2 e^{\frac{j2\pi}{K}(m_2 T + r_2 T^2)} e^{j2\pi \frac{2r_2 T}{K} l} + \dots + CTs, \quad (3.3)$$

Onde \bar{y} é o conjugado complexo de y , e T é o valor diferente de zero em Z_K . O asterisco indica uma multiplicação circular entre \bar{y}_1 e y_1 em que o último mostra um deslocamento de $l + T$. Então devido a essa natureza circulante, o termo $l + T$ pode ser indicado como $l + T \bmod K$ onde $\bmod K$ indica módulo K . E na Equação 3.3 o termo CTs significa termos cruzados, que são dados pela equação 3.4.

$$CTs = \sum_{i,j \neq i} x_i \bar{x}_j e^{\frac{j2\pi}{K}(m_i T + r_i T^2)} e^{\frac{j2\pi}{K} l (m_i - m_j + 2Tr_i) + \frac{j2\pi}{K} l^2 (r_i - r_j)}. \quad (3.4)$$

f_l é feito por senoides com frequência de $\frac{2r_i T}{K}$ e componentes cruzadas de *chirp*. Essas componentes tem energia espalhada ao longo do espectro, embora a componente senoidal tenha uma energia concentrada em $\frac{2r_i T}{K}$. Isso é usado para detecção de r_i utilizando transformada discreta de Fourier de y_l .

O passo a passo para a recuperação do sinal utilizando f_l é apresentado a seguir:

1º Passo Escolhendo um T diferente de zero e $T \in Z_K$ e um limite mínimo de energia de f_l depois de sua atualização.

2º Passo Atualizando ou obtendo f_l através da última atualização de y_l .

3º Passo Fazendo a Transformada Rápida de Fourier (FFT) de comprimento K de f_l .

4º Passo Obtendo o r_i dominante de $2r_i T \bmod K$ no ponto de pico da FFT de f_l .

5º Passo Obter y_l como y_l^d após o processo de recuperação do sinal usando códigos *chirp* utilizando o r_i obtido.

6º Passo Obter a FFT de comprimento K de sinal recuperado y_l .

7º Passo Obter m_i como a localização do pico FFT de comprimento K do sinal recuperado y_l .

8º Passo Obter m_i como a amplitude do pico da FFT de comprimento K do sinal recuperado y_l .

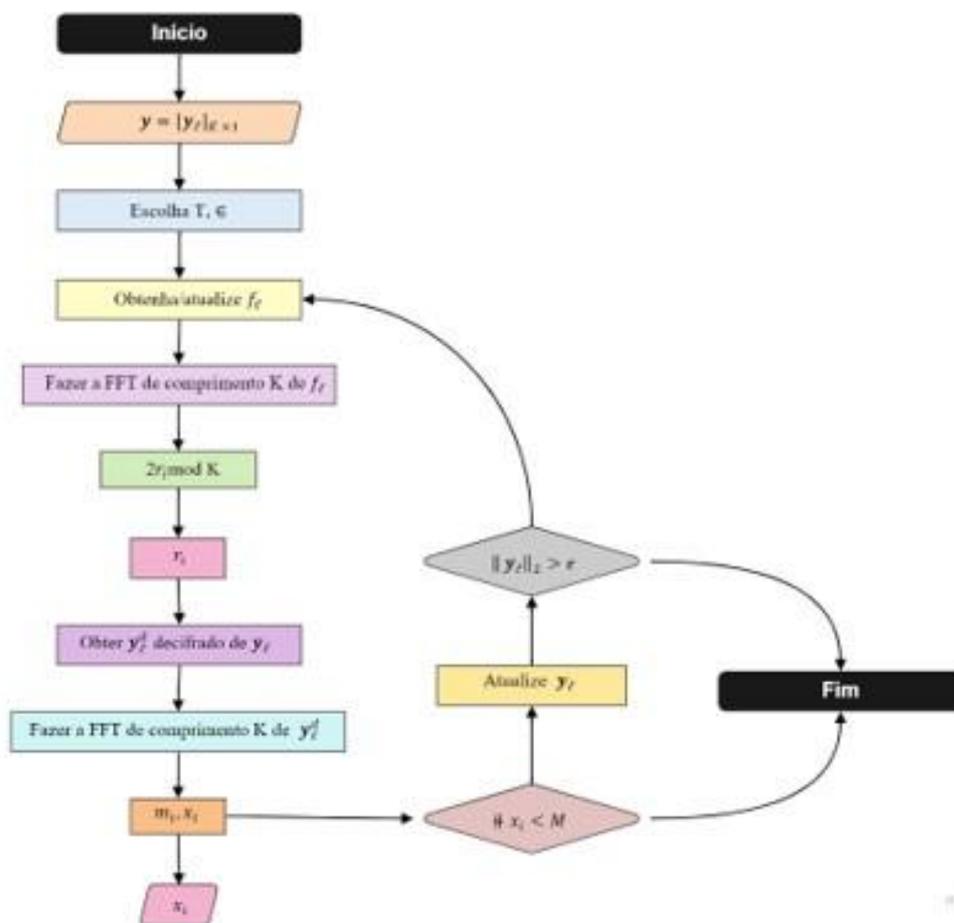
9º Passo Atualizar y_l através da remoção da componente correspondente da amostra de \vec{x} .

10º Passo Se a interação do looping tiver sido repetida M vezes ou a energia de y_l for menor que ϵ o processo será finalizado, caso contrário deve-se repetir os passos de 2 a 9.

A energia de y_l é a sua norma 2.

Para ilustrar esse algoritmo pode-se observar no Fluxograma da Figura 5 os passos.

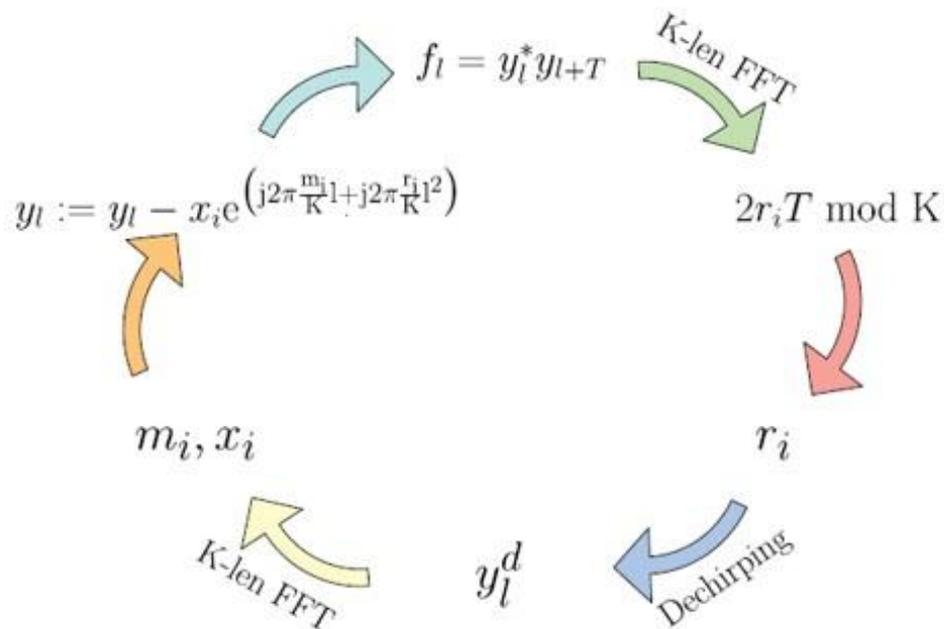
Figura 5 – Fluxograma do algoritmo de recuperação do sinal.



Fonte: Elaborado pelo autor.

Também pode-se ver essa recuperação como um ciclo e isso pode ser visto na Figura 6.

Figura 6 – Fluxograma do ciclo de recuperação do sinal.



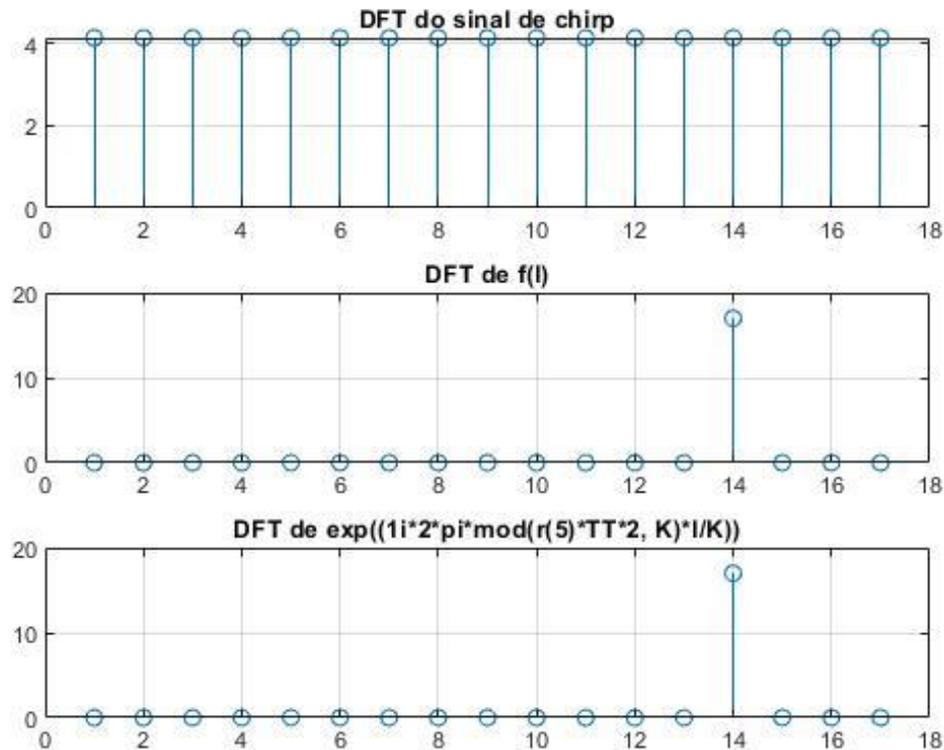
Fonte: Elaborado pelo autor.

3.3.1 Um único chirp

Inicialmente foi visto as características de \mathbf{f} quando \mathbf{y} é composto por um único chirp, isso significa que o vetor do sinal \mathbf{x} antes de ser comprimido por CS tem apenas uma amostra com valor representativo que é representada por uma única componente de *chirp*. Viu-se também uma mistura de códigos de *chirp* correspondente a múltiplas amostras de sinal.

Na Figura 7, tem-se f_l de um único *chirp* tem a energia de sua DFT espalhada pelo espectro, com a concentração em $r_i 2T \text{ mod } K$. Entretanto isso pode ser utilizado para se obter a taxa de *chirp* r_i . Pode-se observar isso teoricamente na Eq.3.2. Para utilizar essa propriedade de f_l , existe uma bijeção entre o conjunto de r_i e o conjunto de $\{2r_i T \text{ mod } K\}$, de outra forma, tendo $2r_i T \text{ mod } K$ pela DFT de f_l , não se encontra a correspondente r_i .

Figura 7 – Módulo da DFT do código de *Chirp* no topo, Módulo da DFT de $f(l)$ no meio e Módulo da DFT de uma senóide com frequência $r_5 2T \bmod K$ em baixo.



Fonte: Elaborado pelo autor.

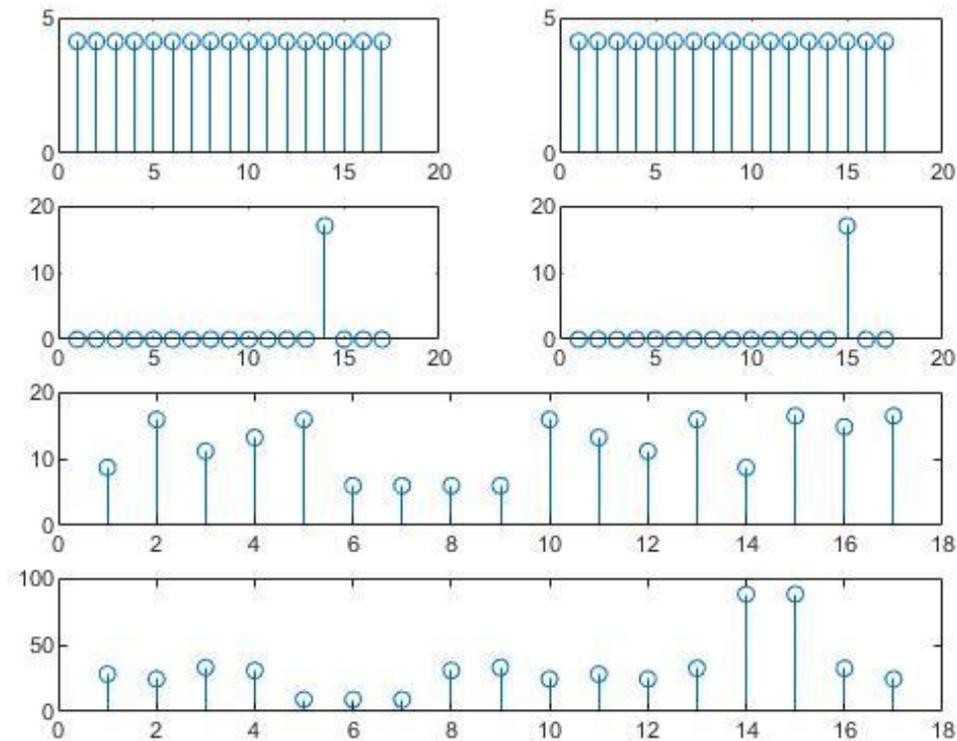
3.3.2 Mistura de 2 chirps

Nesta seção a FFT de uma mistura de dois *chirps* e o vetor resultante \mathbf{f} serão analisados.

Na Figura 8, tem-se f_l dos chirps misturados tem a energia no espectro concentrada em $r_1 2T \bmod K$ e $r_2 2T \bmod K$. Então pode-se detectar o r_i correspondente. Mas no caso da DFT de cada *chirp* assim como a DFT da mistura de chirps, é visto que a energia da DFT é espalhada ao longo das frequências.

Figura 8 – Módulo da DFT dos códigos de *chirp* separadamente no topo, Módulo da DFT da mistura dos códigos de *chirp* no meio e Módulo da DFT de \tilde{f} em baixo.

8



Fonte: Elaborado pelo autor.

3.3.3 Bijeção entre dois conjuntos $\{r_i\}$ e $\{2r_i T \bmod K\}$

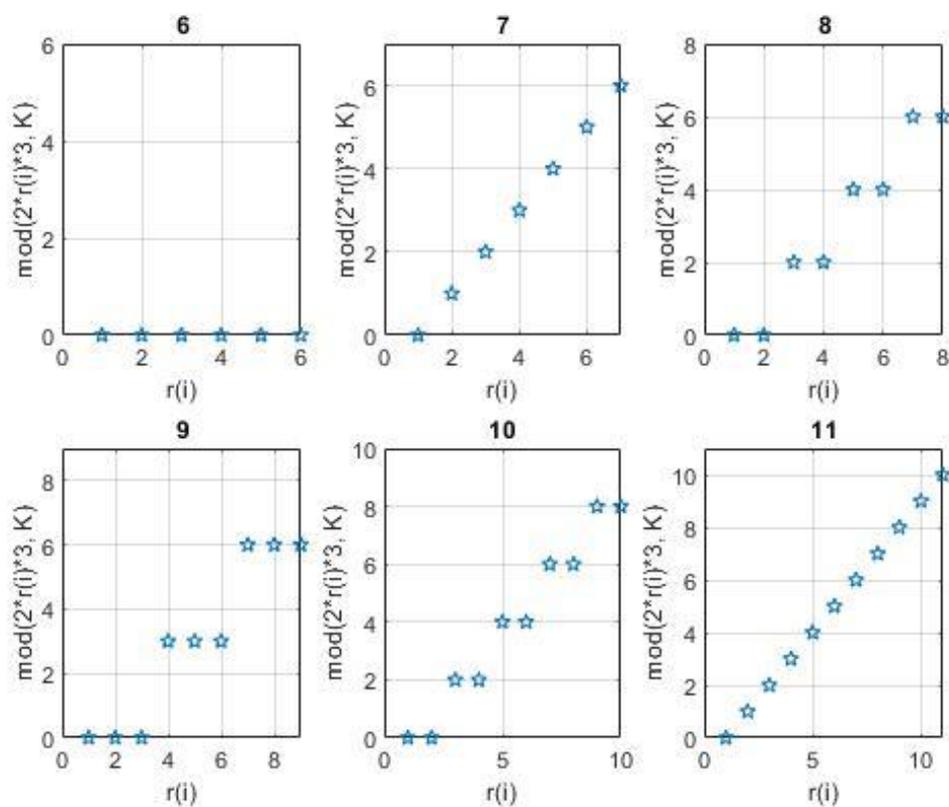
A bijeção entre r_i e $2r_i T \bmod K$ é essencial para a extração da taxa de chirp, que é um importante elemento na identificação de um código de *chirp*. Como é indicado na referência [Applebaum et al. 2009], para se cumprir a bijeção, é necessário que K seja um número primo.

Nas Figuras 9 e 10, a correspondência entre dois conjuntos é mostrado para diferentes valores de K . Para K igual a 6, 8, 9, 10, 12, 14, 15, 16 não tem uma bijeção. Quando K é igual a 6, todos os valores correspondentes a r_i são 0. Já para K igual aos números 8, 10, 14 e 16, os dois valores de r_i são o mesmo valor

daqueles do outro conjunto.

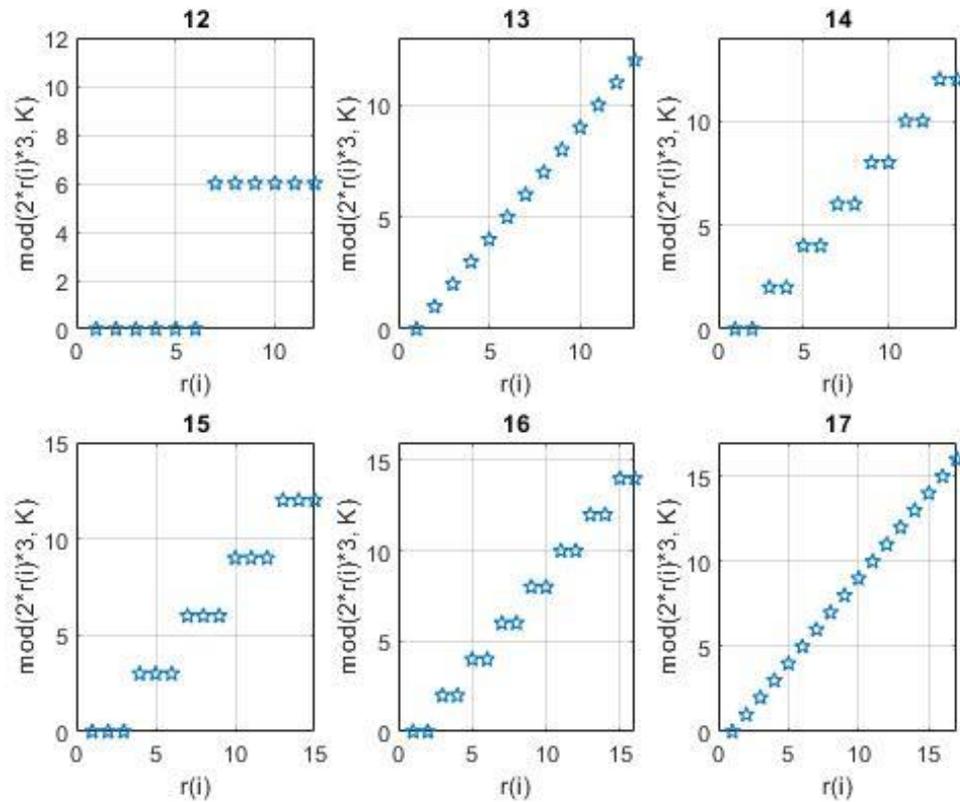
Já para K igual a 9 e 15 cada um dos três valores de r_i correspondem ao mesmo valor daqueles do outro conjunto. E para K igual a 12, cada cinco valores de r_i são correspondentes aos valores do outro conjunto. Apesar de todos os casos em que a bijeção é desejada ela não existe, quando K é um número primo, ou seja, 7, 11, 13 e 17 a correspondência desejada é encontrada.

Figura 9 – Ilustração da bijeção entre dois conjuntos $\{r_i\}$ e $\{2r_i \bmod K\}$ para os valores de K de 6, 7, 8, 9, 10, 11.



Fonte: Elaborado pelo autor.

Figura 10 – Ilustração da bijeção entre dois conjuntos $\{r_i\}$ e $\{2r_i T \bmod K\}$ para os valores de K de 12, 13, 14, 15, 16, 17.



Fonte: Elaborado pelo autor.

3.3.4 Decifrando (Dechirping)

Após a detecção de r_i a partir da frequência dominante de f_l , o vetor do sinal y_1 é decifrado pelo correspondente dominante r_i de acordo com Equação 3.5.

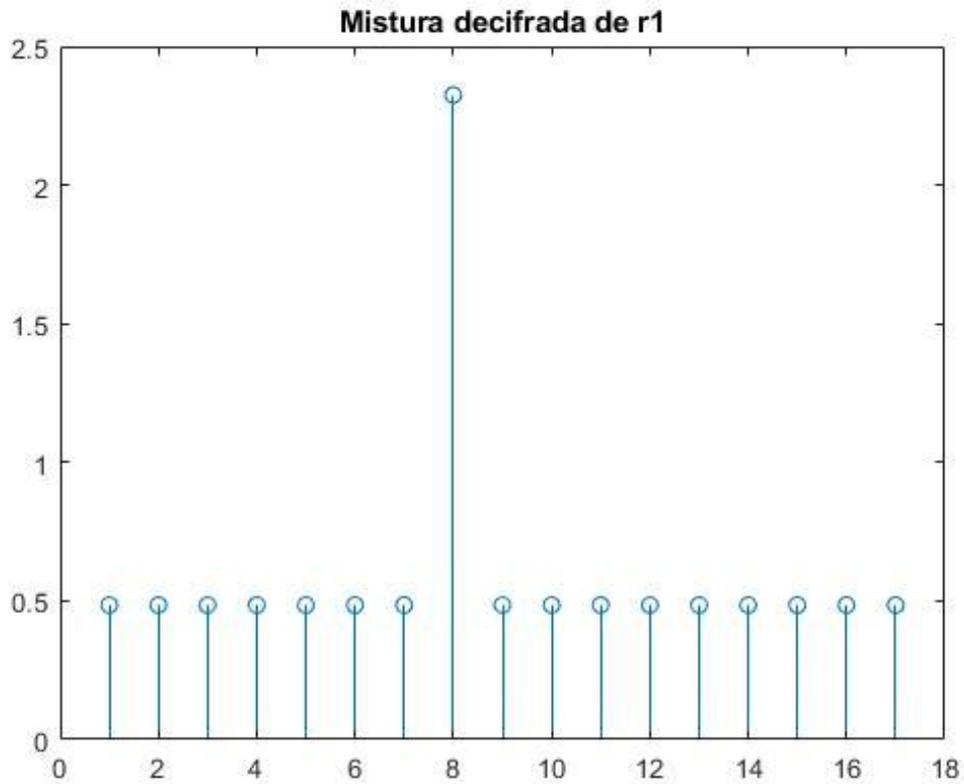
$$y_i^d = y_l e^{-\frac{j2\pi r_i l^2}{K}}. \quad (3.5)$$

Após decifrar o vetor do sinal y_l , desde que a taxa de *chirp* r_i seja removida, a energia correspondente não se espalha no espectro, isso é observável em m_i .

Na Figura 11, o vetor do sinal decifrado tem a concentração da energia ao

redor do 8 ponto da DFT. Ou seja, $m_i = 8$ e sua amplitude indica a amostra do sinal recuperado correspondente que nesse caso é aproximadamente $s_1 \approx 2$.

Figura 11 – Mistura decifrada de r_1



Fonte: Elaborado pelo autor.

3.4 Detecção e recuperação do sinal

É importante reforçar que a bijeção entre o conjunto de taxas de *chirp* $\{r_i | r_i \in Z_K\}$ e o conjunto de pontos da FFT $\{\frac{2r_i T \text{mod} K}{K} | r_i \in Z_K\}$. No código de *chirp* de CS, a matriz de detecção é feita por vários sinais de *chirp* em suas colunas, são K^2 códigos de *chirp* de comprimento K .

O sinal de comprimento K^2 pode ser comprimido durante sua detecção como um vetor de um sinal de comprimento K , ele é a combinação linear de códigos

de *chirp* de comprimento K . Enquanto as amostras do sinal inicial de comprimento K^2 tem o papel de pesos da combinação linear.

A recuperação do sinal é um processo iterativo, em que em cada loop, \mathbf{f} é atualizado a partir da última atualização de \mathbf{y} . A DFT de comprimento K de \mathbf{f} dá a taxa de *chirp* r_i correspondente a amostra dominante do sinal. Decifrando \mathbf{f} usando o r_i encontrado e utilizando a DFT de comprimento K , temos o m_i correspondente e também x_i .

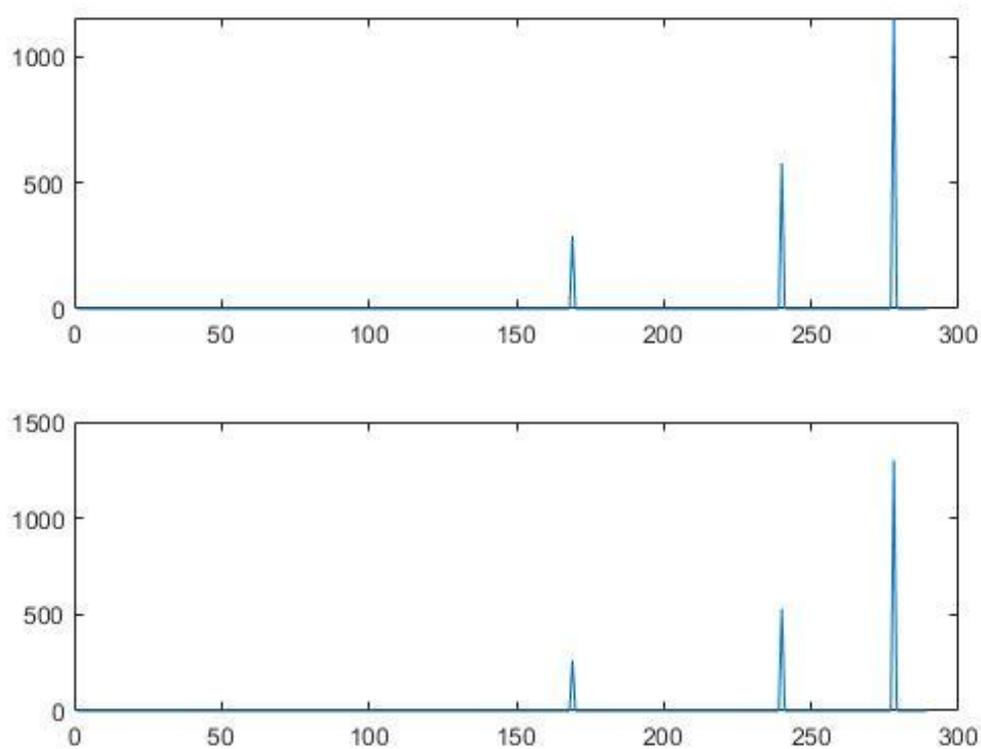
Com r_i , m_i e x_i temos o valor da amostra do sinal e sua localização. Então, com a amostra do sinal, \mathbf{y} pode ser atualizado através da remoção dos códigos de *chirp* correspondentes a partir da mistura de acordo com (3.6).

$$y_l := y_l - x_i \exp(j2\pi \frac{m_i}{K} l + j2\pi \frac{r_i}{K} l^2). \quad (3.6)$$

Isso se repete até que a energia da mistura de *chirp* atinja o valor mínimo estipulado. O código no Anexo 6.1 mostra todo esse processo de compressão durante a detecção e a recuperação de um sinal harmônico.

Podemos observar na Figura 13 que os sinais se aproximam muito mas não são idênticos, existem pequenos erros, embora a FFT de ambos visto na 12 tenha os bins localizados corretamente, as amplitudes possuem erros e por isso essa diferença entre o sinal original e reconstruído.

Figura 12 – Espectro do sinal original acima e reconstruído abaixo.



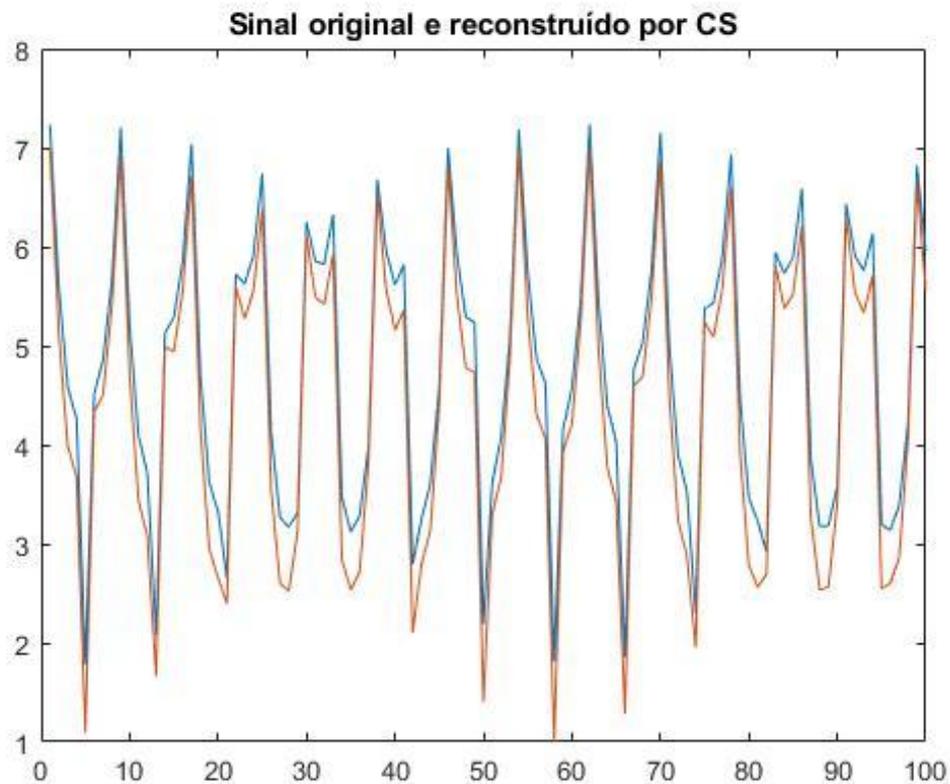
Fonte: Elaborado pelo autor.

3.5 Análise de K e taxa de compressão

Nessa metodologia apresentada neste trabalho, podemos observar o impacto que a variável K tem na análise dos sinais. Essa variável representa o comprimento do sinal esparso, o comprimento do sinal de *Chirp* que será usado para construir a matriz, e também é ela que determina a dimensão da matriz de *chirp* de CS, pois suas dimensões são $K \times K^2$, e isso gerará um vetor comprimido \mathbf{y} de comprimento K .

Com essas informações pode-se perceber que a taxa de compressão (TC) deste método é dada por :

Figura 13 – Sinal original e reconstruído sobrepostos.



Fonte: Elaborado pelo autor.

$$TC = \frac{K^2}{K} = K \quad (3.7)$$

E com essa equação 3.7, vemos que a taxa é de K , ou seja, essa variável representa o valor da compressão do sinal, quanto maior o seu valor, maior será a compressão do sinal. É importante lembrar que esse número deve ser primo, para que a construção da matriz de *chirp* seja única.

O equilíbrio desse número será avaliado no próximo capítulo com o uso de sinais sintéticos de várias formas.

3.6 Energia do sinal

Para uma melhor análise da reconstrução do sinal, será calculada a energia do sinal, ela é dada por:

$$E = \int x^2 dx \quad (3.8)$$

No próximo capítulo serão analisadas as energias dos sinais originais e reconstruídos e comparadas para análise da reconstrução.

3.7 Limitações

Neste capítulo fica claro que existem diversas limitações nesse método e no desenvolvimento em si. Várias exigências devem ser cumpridas para que o método funcione e ainda sim existem erros como visto na Figura 13.

Foram vistos apenas dois sinais misturados, a realidade é um número muito maior de sinais onde o erro aumenta a cada sinal aumentado tornando o método ineficaz. Mas ao comparar sua aplicabilidade com os métodos que utilizam Nyquist para processar seus sinais observa-se que têm ganhos tão grandes que podem baratear os softwares que faz com que existam diversos casos onde o método sendo aplicado trás benefícios que superam os pontos negativos.

No capítulo a seguir iremos demonstrar esse método com outros sinais e analisar cada um deles.

4 RESULTADOS

Neste capítulo será feita uma breve revisão do método e as regras utilizadas e alguns casos e seus resultados.

4.1 Método utilizado para gerar resultados

Nos capítulos anteriores, viu-se que o método de códigos de *chirp* para *compressive sensing* é basicamente multiplicar o sinal original x por uma matriz de CS composta por códigos de *chirp* A e obter o sinal comprimido e processado y :

$$\mathbf{y}_{m \times 1} = A_{m \times n} \mathbf{x}_{n \times 1} \quad (4.1)$$

Em que $m \ll n$ e por isso a técnica é tão interessante, ela transforma um sinal com m amostras em um sinal com n amostras, que é bem menor. É importante lembrar que $m = K$ e $n = K^2$. Sabemos também que a matriz de *chirp* A é feita da seguinte maneira:

$$A = \Psi \Phi \quad (4.2)$$

E Φ é a transformada de Fourier, que é onde o sinal harmônico utilizado é esparsos e Ψ são os códigos de *chirp*. No entanto para se obter um bom resultado essa matriz Φ deve atender a propriedade RIP e para satisfazê-la temos 4.3.

$$M < \frac{\sqrt{K} + 1}{2} \quad (4.3)$$

Em que M são as componentes não zero, que neste caso serão as componentes de frequência harmônicas do sinal e K é o comprimento dos códigos de *chirp*.

Serão demonstrados resultados dos seguintes casos:

Tabela 1 – Casos estudados neste capítulo

Caso	K	M (Máximo de frequências)	Frequências utilizadas (Hz)	Amplitudes
Caso 1	11	2	2 e 30	Todas 1
Caso 2	11	2	2 e 100	Todas 1
Caso 3	11	2	2 e 100 e 180	Todas 1
Caso 4	41	3	2 e 100 e 180	Todas 1
Caso 5	41	3	2 e 300 e 580	Todas 1
Caso 6	41	3	2 e 100 e 300 e 600	Todas 1
Caso 7	41	3	Aleatórias de 2 a 8 componentes	Todas 1
Caso 8	97	5	Aleatórias de 2 a 12 componentes	Todas 1
Caso 9	41	4	50 e 150 e -150 e -50	1, 0.3, 0.3, 50
Caso 10	103	5	2 e 100 e 300 e 600 e 800	Todas 1
Caso 11	137	6	2 e 100 e 300 e 600 e 800 e 750	Todas 1
Caso 12	179	7	2 e 100 e 300 e 600 e 800 e 750 e 1000	Todas 1
Caso 13	251	8	2 e 100 e 300 e 600 e 800 e 750 e 1000 e 1200	Todas 1

Fonte: Elaborado pelo autor.

4.2 Análise dos Casos 1, 2 e 3

Nesta seção serão analisados os casos 1, 2 e 3, para todos os casos teremos $K = 11$ e pela equação 4.3, obtida pela propriedade RIP, o número de componentes possíveis para esse algoritmo são duas. No caso 1 tem-se um sinal harmônico com 2 componentes de frequência, cujas frequências são 2Hz e 30Hz e as amplitudes são 1. Já no caso 2 tem-se um sinal harmônico também com 2 componentes de frequência, cujas frequências são 2 e 100 e as amplitudes são 1. Percebe-se que a diferença entre o caso 1 e caso 2 são que as frequências estão mais espaçadas, espera-se que nesse caso o erro aumente devido a esse maior espaçamento. No caso 3 o sinal tem 3 componentes de frequência, e elas são 2, 100 e 180 e as amplitudes são 1. Esse número de frequências é superior ao possível para que o sinal respeite a propriedade RIP, por isso espera-se que o erro seja maior que no caso 2.

A Figura 14a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 1. Observa-se que o algoritmo identificou corretamente a posição dos *bins*, porém as amplitudes estimadas apresentaram desvio. A Figura

14b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS. O MSE (Mean Square Error) para este caso foi de 0,0218. A Energia do sinal reconstruído sobre a Energia do sinal original Er/Eo foi de 0,1376.

A Figura 15a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 2. O algoritmo identificou corretamente a posição dos *bins*, porém também para esse caso as amplitudes estimadas apresentaram desvio, esse desvio foi maior que o caso anterior, pois as frequências estão mais espalhadas no espectro confirmando o esperado. A Figura 15b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS. O MSE para este caso foi superior ao caso 1 e tem o valor de 0,039. A Energia do sinal reconstruído sobre a Energia do sinal original Er/Eo foi de 0,0000006.

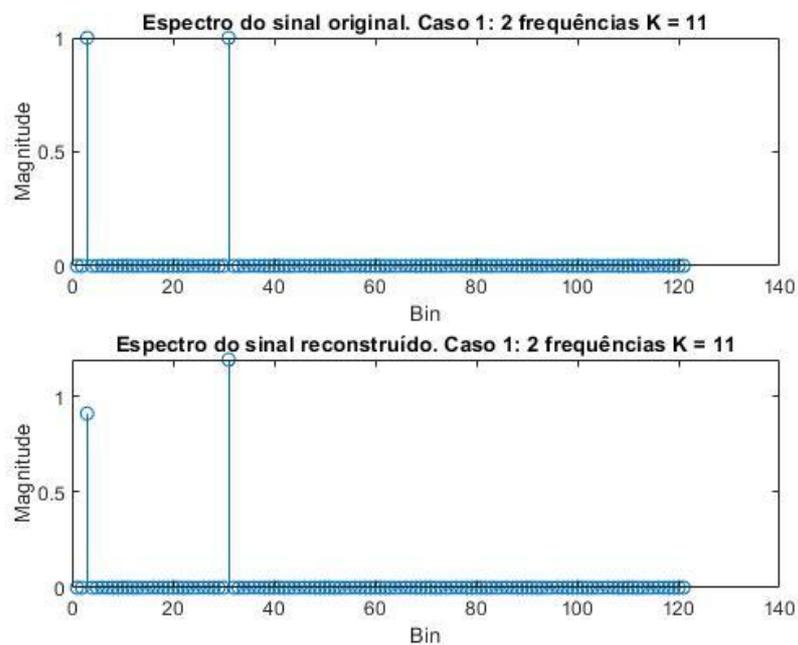
A Figura 16a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 3. Ainda que as componentes de frequência sejam superiores ao indicado pela equação 4.3 para que a propriedade RIP seja respeitada pelo sinal em análise, o algoritmo identificou corretamente a posição dos *bins*, e as amplitudes estimadas apresentaram desvio, maior que dos casos anteriores mas ainda sim pequeno. A Figura 16b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS, observa-se que a reconstrução já se desvia com maior intensidade em relação aos casos anteriores. E o MSE para este caso foi de 0,0898, cerca de 4 vezes maior que o caso 1 e mais que o dobro do caso 2. Analisando essa diferença podemos perceber que quando a propriedade não é respeitada podemos ter um erro bem maior. Os erros desses casos estão demonstrados na tabela 2. A Energia do sinal reconstruído sobre a Energia do sinal original Er/Eo foi de 0,00004.

Tabela 2 – Erros dos casos 1, 2 e 3

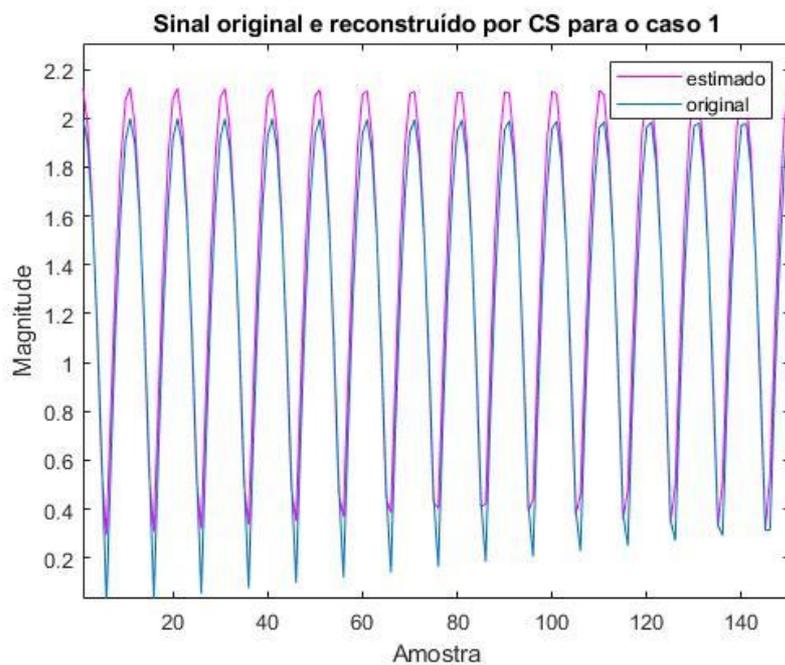
Caso	K	M (Máximo de frequências)	Frequências utilizadas	Amplitudes	MSE (Mean square error)	Er/Eo
Caso 1	11	2	2 e 30	Todas 1	0.0218	0.1376
Caso 2	11	2	2 e 100	Todas 1	0.0390	0.0000006
Caso 3	11	2	2 e 100 e 180	Todas 1	0.0898	0.00004

Fonte: Elaborado pelo autor.

Figura 14 – Resultados do Caso 1



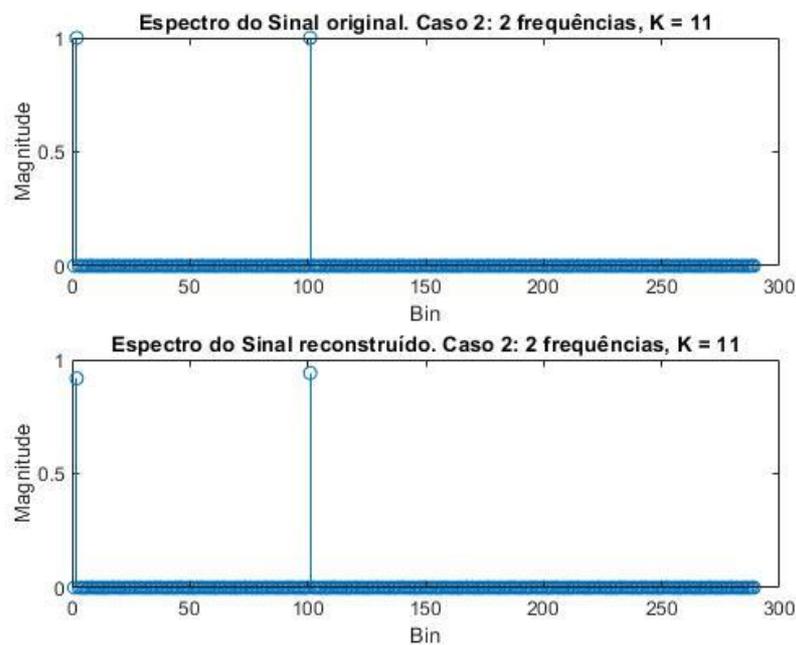
(a) Espectro do sinal para o caso 1. Superior: sinal original; Inferior: sinal estimado



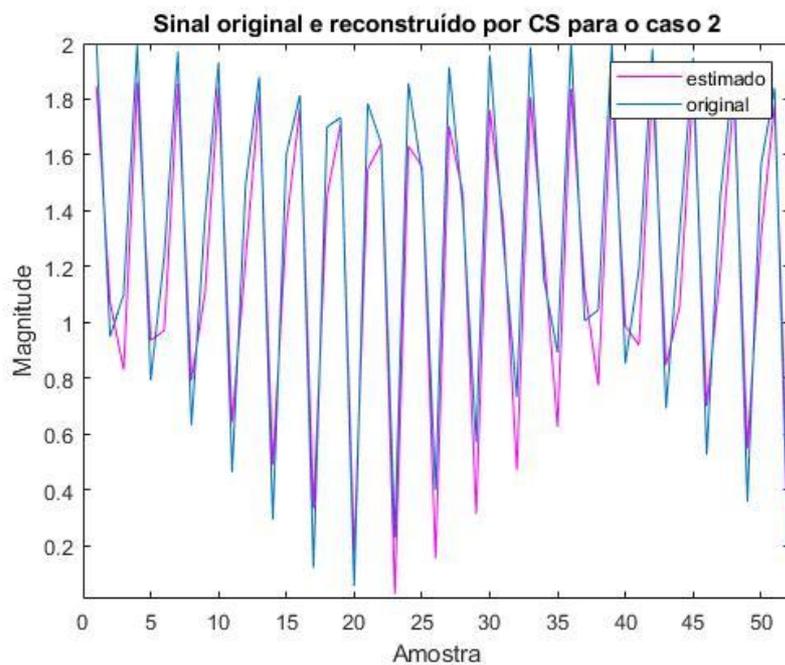
(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 1

Fonte: Elaborado pelo autor.

Figura 15 – Resultados do Caso 2



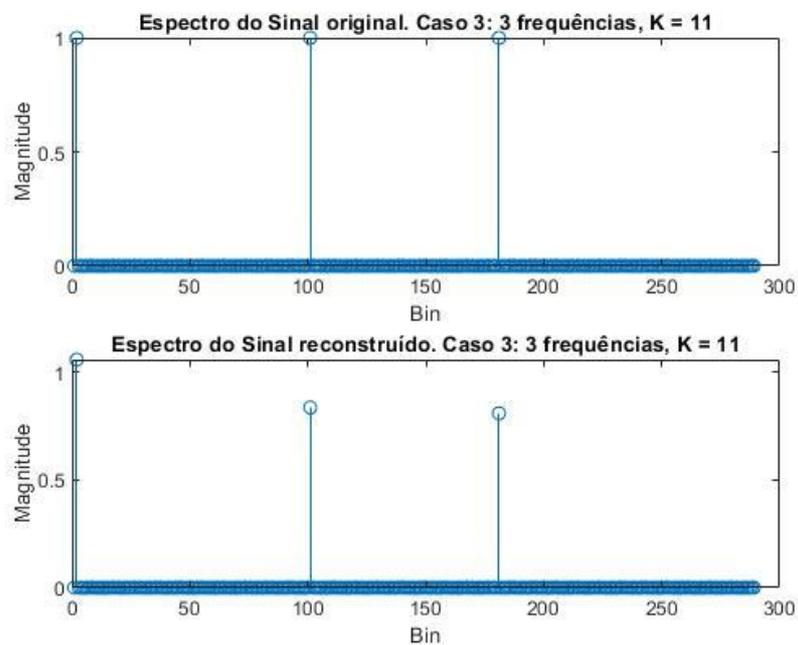
(a) Espectro do sinal para o caso 2. Superior: sinal original; Inferior: sinal estimado



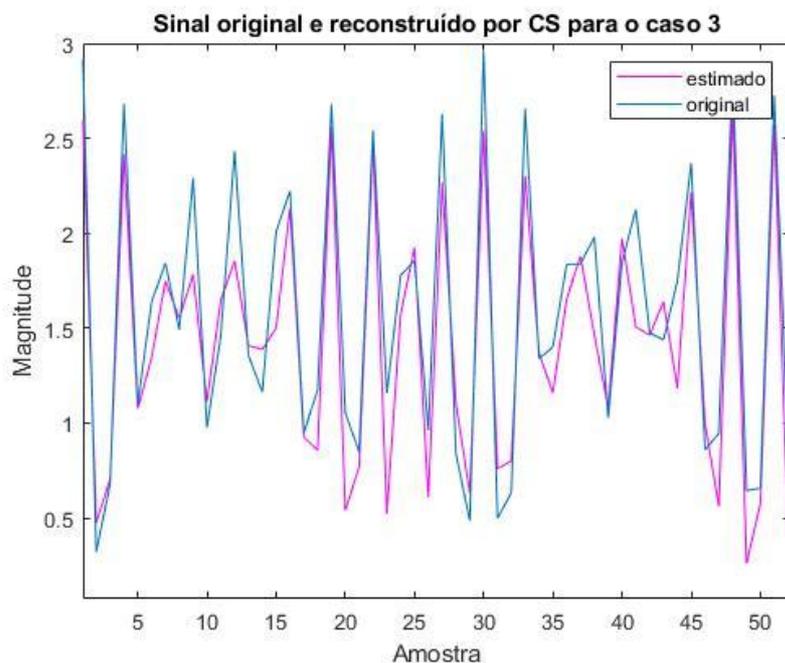
(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 2

Fonte: Elaborado pelo autor.

Figura 16 – Resultados do Caso 3



(a) Espectro do sinal para o caso 3. Superior: sinal original; Inferior: sinal estimado



(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 3

Fonte: Elaborado pelo autor.

4.3 Análise dos Caso 4, 5 e 6

Nesta seção serão analisados os casos 4, 5 e 6, para todos os casos teremos $K = 41$, como visto na seção anterior a equação 4.3, obtida pela propriedade RIP, define o número máximo de componentes e neste caso serão três. Nos casos 4 e 5 tem-se um sinal harmônico com 3 componentes de frequência, sendo que no caso 4 elas são 2, 100 e 180 e no caso 5 elas são 2, 300 e 580, ou seja, estão mais espalhadas no espectro. Para ambos os casos as amplitudes são 1. Espera-se que o erro do caso 5 seja maior que o caso 4. No caso 6 o sinal tem 4 componentes de frequência, elas são 2, 100, 300 e 600, não respeitando a propriedade RIP e com isso espera-se um erro maior.

A Figura 17a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 4. Observa-se que o algoritmo identificou corretamente a posição dos *bins*, porém as amplitudes estimadas apresentaram um pequeno desvio. A Figura 17b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS. O MSE para este caso foi de 0,0298.

A Figura 18a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 5, observa-se que em relação ao caso 4 já percebe-se que houve uma piora na reconstrução. Todavia o algoritmo identificou corretamente a posição dos *bins*, mas também para esse caso as amplitudes estimadas apresentaram desvio, esse desvio foi maior que o caso anterior, pois as frequências estão mais espalhadas no espectro confirmando o esperado. A Figura 18b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS. O MSE para este caso foi superior ao caso 4 e tem o valor de 0,0488.

A Figura 19a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 6. Ainda que as componentes de frequência não respeitem a propriedade RIP, o algoritmo identificou corretamente a posição dos *bins*, as amplitudes estimadas apresentaram desvio maior que dos casos anteriores. A Figura 19b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS, observa-se que a reconstrução já se desvia com maior intensidade em relação aos casos anteriores. E o MSE para este caso foi de 0,0674, cerca de 2 vezes maior que o caso 4. Analisando essa diferença podemos perceber que quando a propriedade

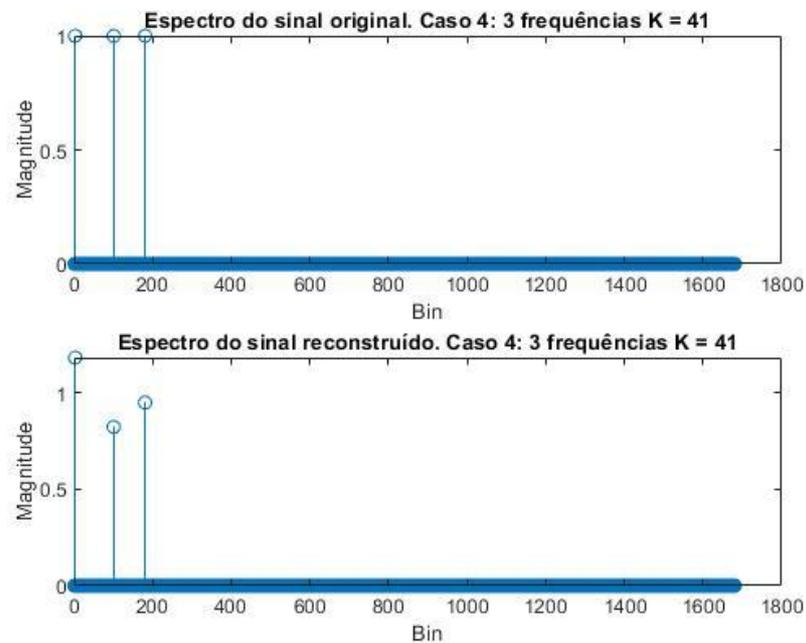
não é respeitada podemos ter um erro bem maior. Para esses casos, o erro não chega a ser tão maior como nos casos anteriores, pois quanto maior o valor de K, mais amostras se tem e o erro vai diminuindo. Confirma-se isso observando os casos 3 e 4, onde eles têm o mesmo número de componentes de frequências, com as mesmas frequências e amplitudes, variando apenas o valor de K, e como esperado o valor do erro do caso 4 é inferior ao do caso 3 pelo fato de K ser maior. Os erros desses casos estão demonstrados na tabela 4.

Tabela 3 – Erros dos casos estudados neste capítulo

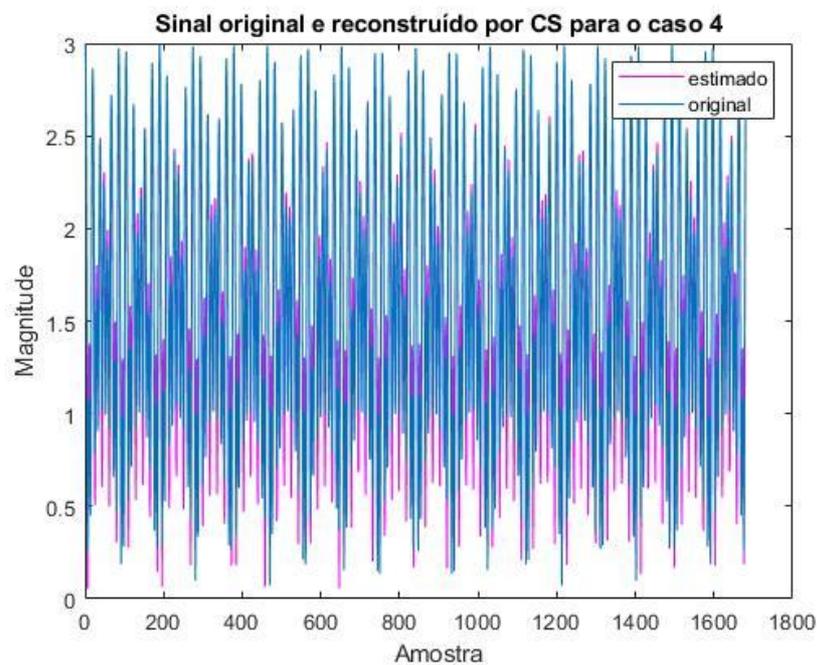
Caso	K	M (Máximo de frequências)	Frequências utilizadas	Amplitudes	MSE (Mean square error)	Er/Eo
Caso 4	41	3	2 e 100 e 180	Todas 1	0.0298	0.0011
Caso 5	41	3	2 e 300 e 580	Todas 1	0.0488	0.00017
Caso 6	41	3	2 e 100 e 300 e 600	Todas 1	0.0674	0.0025

Fonte: Elaborado pelo autor.

Figura 17 – Resultados do Caso 4



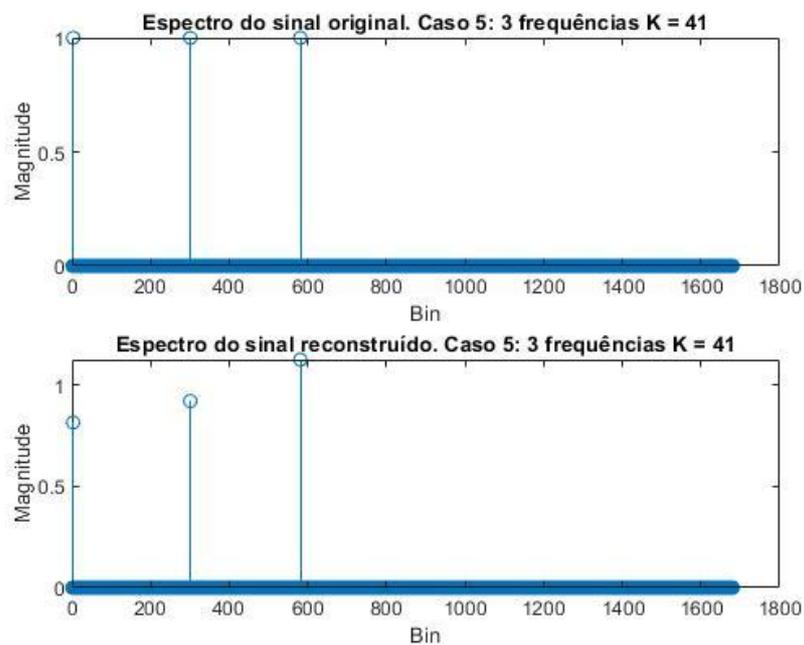
(a) Espectro do sinal para o caso 4. Superior: sinal original; Inferior: sinal estimado



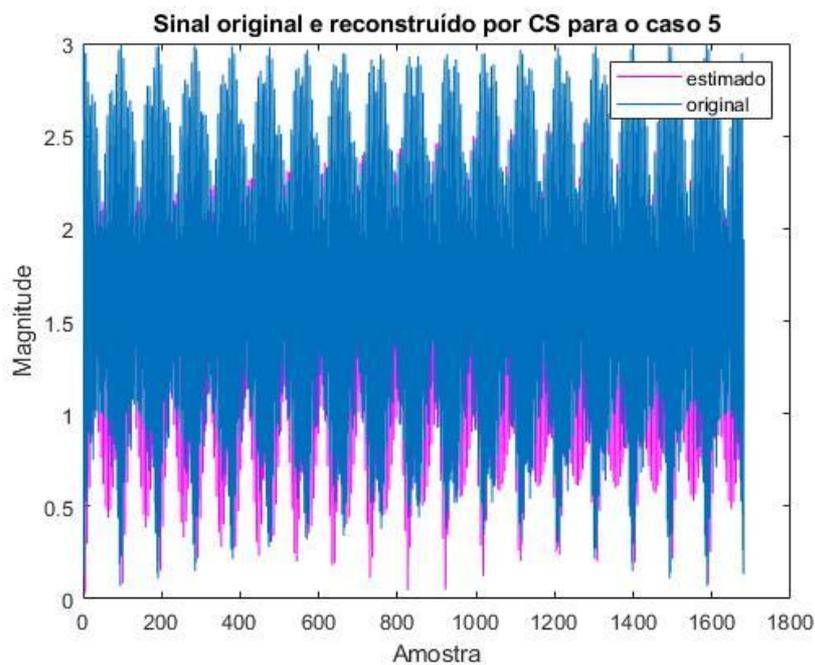
(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 4

Fonte: Elaborado pelo autor.

Figura 18 – Resultados do Caso 5



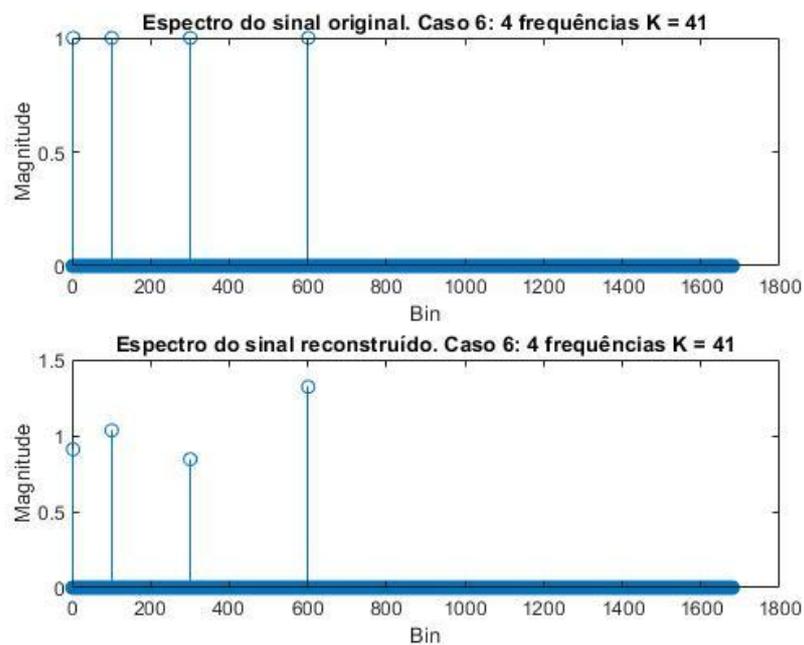
(a) Espectro do sinal para o caso 5. Superior: sinal original; Inferior: sinal estimado



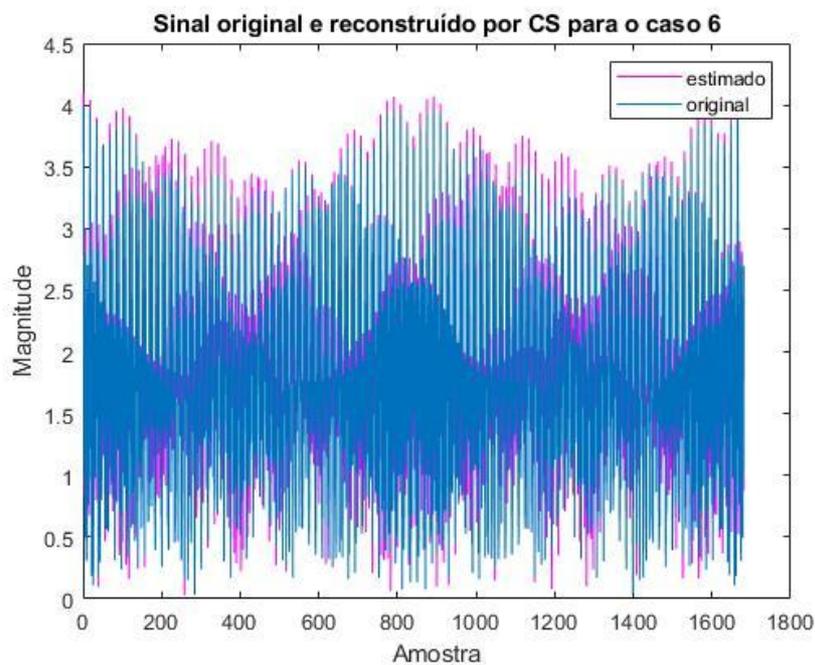
(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 5

Fonte: Elaborado pelo autor.

Figura 19 – Resultados do Caso 6



(a) Espectro do sinal para o caso 6. Superior: sinal original; Inferior: sinal estimado



(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 6

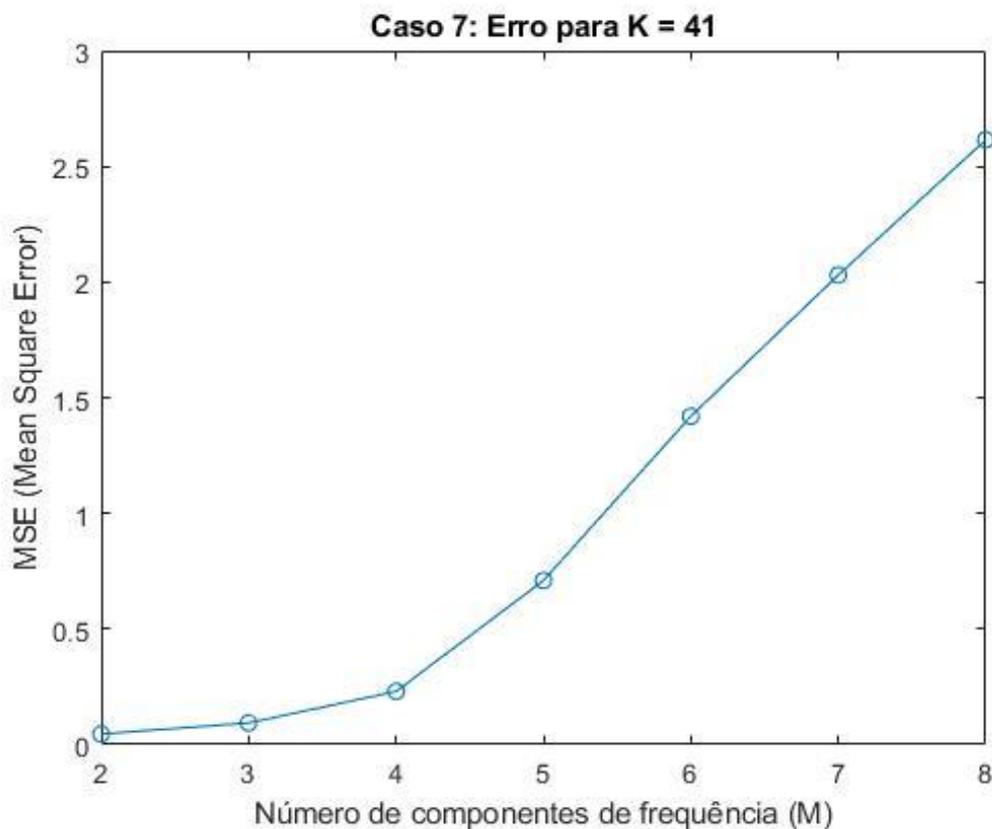
Fonte: Elaborado pelo autor.

4.4 Análise dos Casos 7 e 8

Nesta seção serão analisados sinais harmônicos com frequências geradas aleatoriamente, para o caso 7 $K = 41$ e foram geradas de 2 a 8 componentes de frequências aleatoriamente, para cada quantidade de componentes de frequências o código foi rodado por 100 vezes, o MSE foi armazenado e foi obtida a média dos valores. Já para o caso 8 foi utilizado $K = 97$ e foram geradas de 2 a 12 componentes de frequências aleatoriamente, para cada quantidade de componentes de frequências o código foi rodado por 100 vezes e o MSE armazenado e sua média obtida.

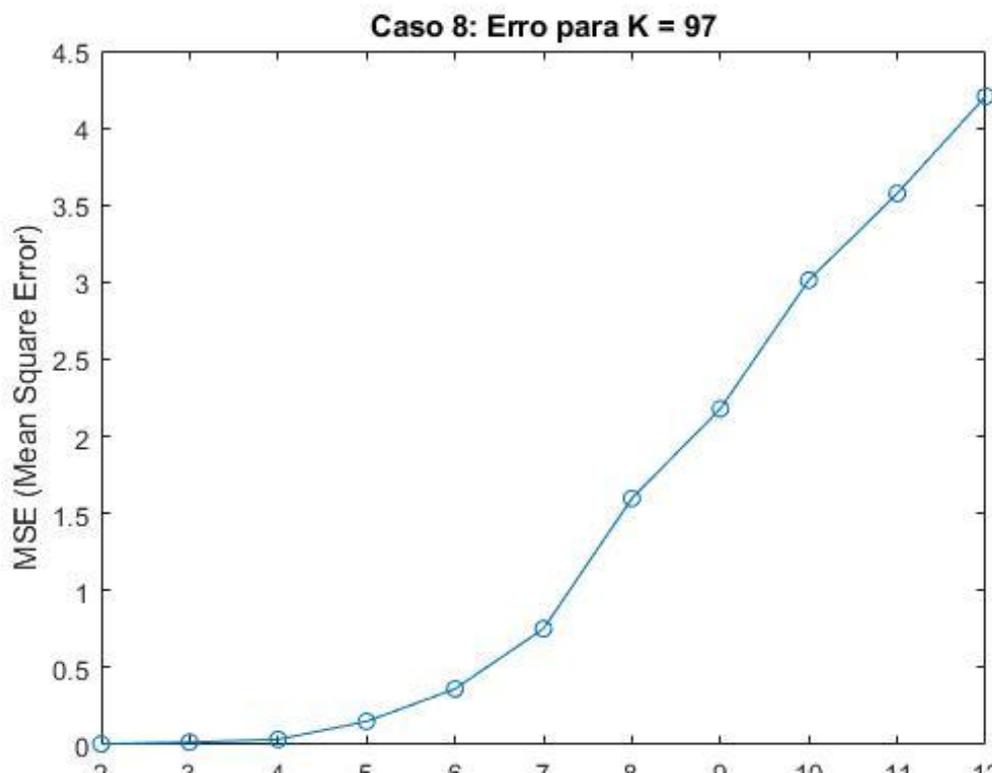
Para que o sinal satisfaça a equação 4.3 que foi obtida pela propriedade RIP, para $K = 41$ o número máximo de componentes que ele deve ter é 3, e para $K = 97$ o número máximo é 5. Na figura 20 observa-se que até 3 componentes o erro ainda é muito baixo, para 4 frequências ele ainda é pequeno mas já tem um leve salto, a partir de 5 componentes o erro fica muito grande, tornando completamente inviável a reconstrução do sinal. Já na figura 21 temos o comportamento similar, ou seja, para o número de componentes de frequências permitida pela propriedade RIP o erro é baixo, neste caso até 5 componentes, e a partir dos sinais com 6 componentes ou mais o erro cresce muito rapidamente tornando inviável a reconstrução.

Figura 20 – Caso 7: MSE para cada número de componentes de frequência para $K = 41$



Fonte: Elaborado pelo autor.

Figura 21 – Caso 8: MSE para cada número de componentes de frequência para $K = 97$



4.5 Análise do caso 9

Nesta caso um sinal real é utilizado para testar o algoritmo. O sinal é dado por $\mathbf{x}[\mathbf{n}] = 2\cos(10.\pi n) + 0.6\cos(300)$. As frequências utilizadas foram 50, 150, -150 e -50 e as amplitudes são 1, 0.3 0.3 e 50 respectivamente.

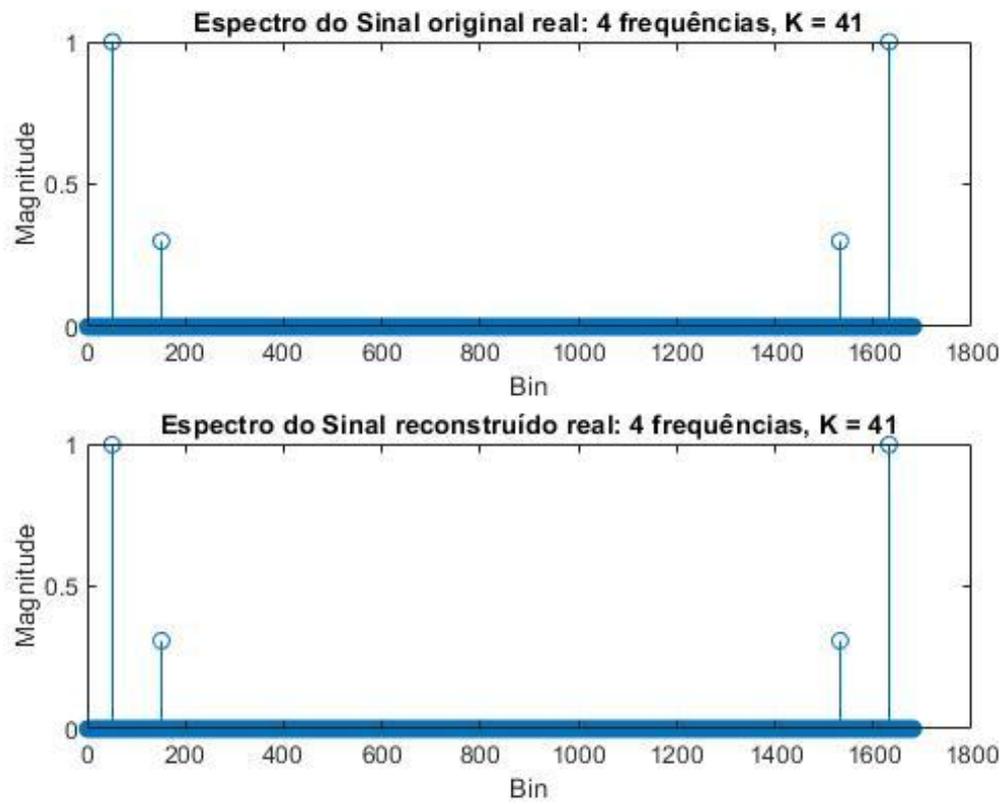
A Figura 22 apresenta o espectro em frequência do sinal original e estimado. Observa-se que as frequências negativas aparecem na parte superior dos *bins* da FFT, como ja esperado. Para garantir a reconstrução de um sinal real para o sinal estimado foi realizada uma manipulação dos valores estimados individualmente de modo que a garantir que o fasor das componentes de 50 e -50 fossem o conjugado um do outro. Para isso utilizou-se a manipulação vista na equação 4.4.

$$\begin{aligned} F(50) &= \frac{(F(50))+(F(-50))^*}{2} \\ F(-50) &= (F(50))^* \end{aligned} \tag{4.4}$$

A mesma correção foi aplicada para as componentes em 150 e -150.

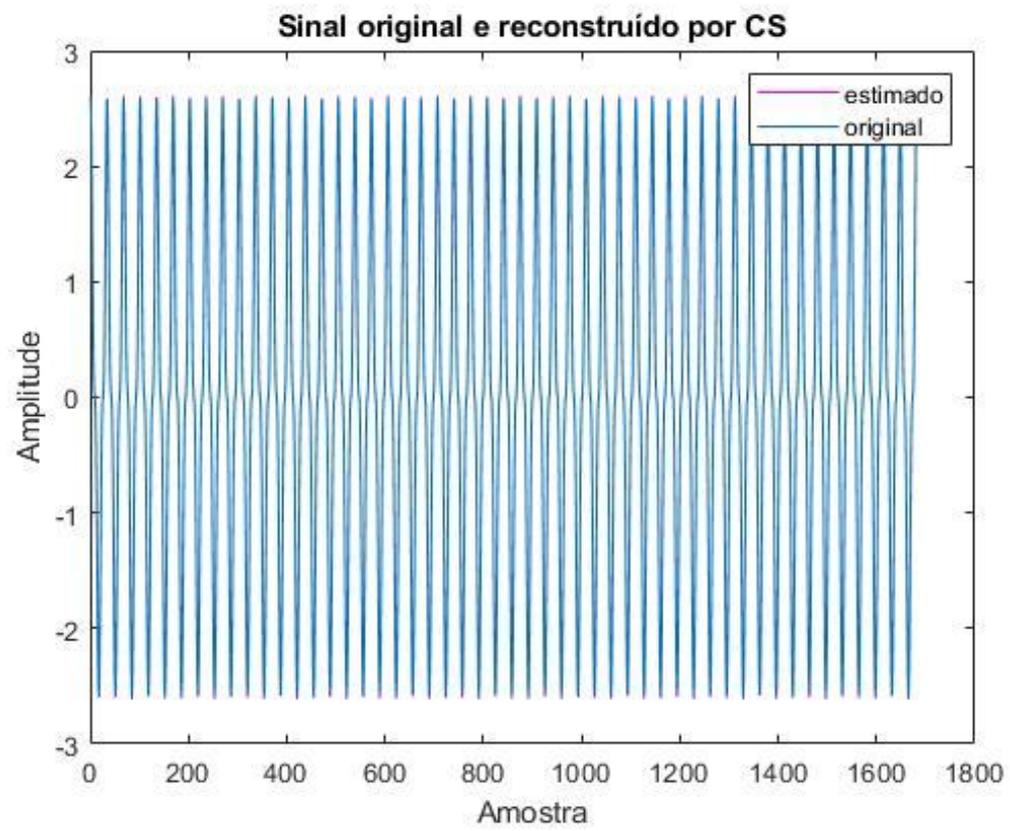
A Figura 23 mostra o sinal original e o sinal reconstruído no domínio do tempo. O MSE para esta reconstrução foi de 0.0018.

Figura 22 – Caso 9: Espectro do sinal original e do sinal estimado por CS



Fonte: Elaborado pelo autor.

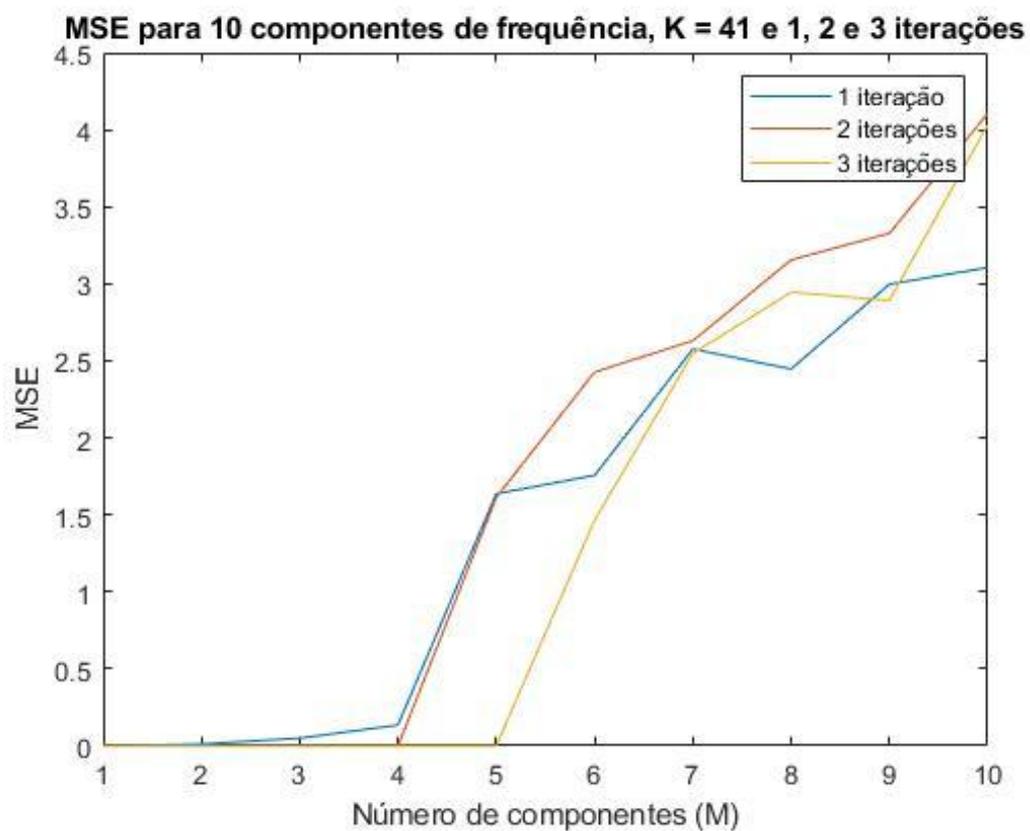
Figura 23 – Caso 9: Sinal real e estimado no domínio do tempo



Fonte: Elaborado pelo autor.

4.5.1 Sinal real com $K=41$ e frequências aleatórias

Foi gerado o mesmo sinal real com $K = 41$, porém dessa vez as frequências foram geradas de maneira aleatória, e foram geradas de 1 até 10 componentes de frequências aleatórias e esse código foi feito para 1,2 e 3 iterações. Para esse valor de K espera-se que o código tenha um bom resultado até 3 componentes de frequência, ou seja, $M = 3$, observa-se que na Figura 24, para o caso com 1 iteração, o algoritmo tem um erro baixo até 3 componentes, porém quando as iterações são duas e três, respectivamente, observa-se que com 2 iterações o MSE ainda fica pequeno até 4 componentes e com 3 iterações o MSE fica pequeno até 5 componentes. Isso significa que o resultado diminui o seu erro a cada nova iteração feita e esse resultado era esperado, pois o código foi feito para que se façam várias iterações até que chegue numa energia desejada.

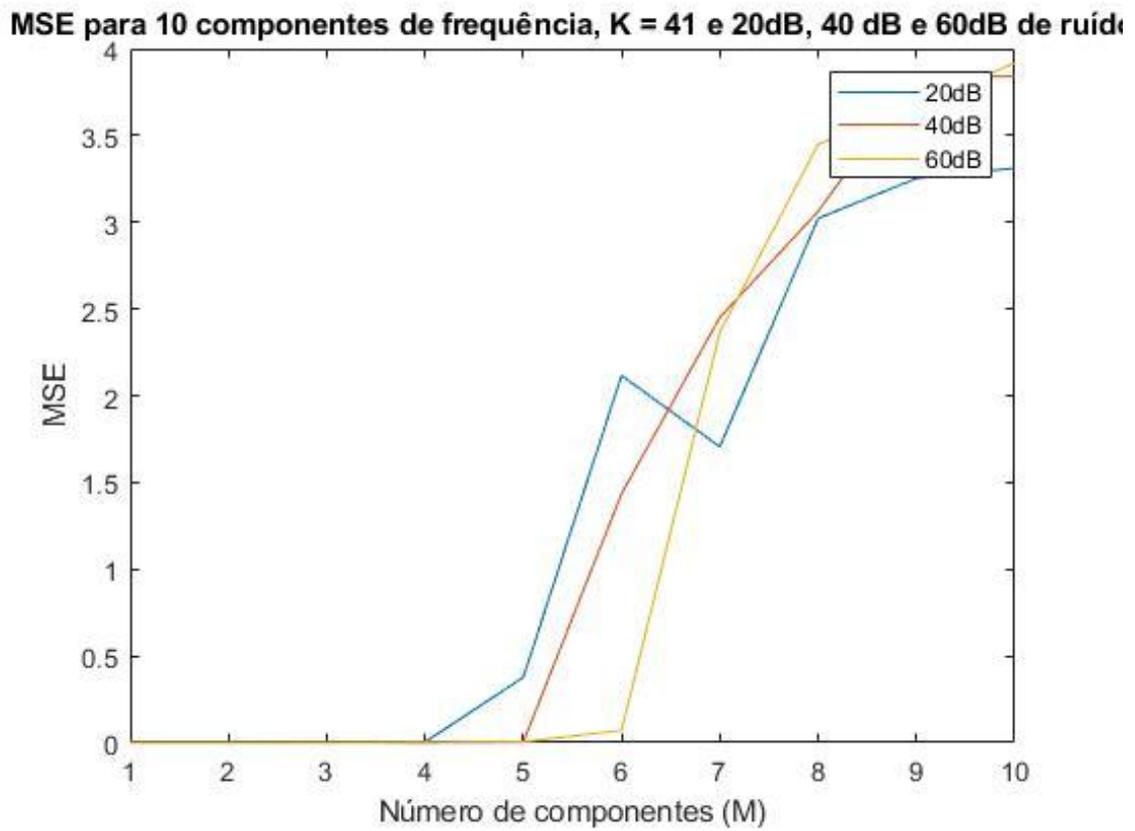
Figura 24 – MSE para sinal real com $K=41$ com uma, duas e três iterações

Fonte: Elaborado pelo autor.

4.5.2 Sinal real com $K=41$ e com adição de ruído e frequências aleatórias

Foi gerado o mesmo sinal real com $K = 41$, porém dessa vez as frequências foram geradas de maneira aleatória, e foram geradas de 1 até 10 componentes de frequências aleatórias, além disso, foi adicionado um ruído branco gaussiano ao sinal com relação sinal-ruído de 20dB, 40dB e 60dB e foram feitas 2 iterações para cada um desses casos. Para esse valor de K espera-se que o código tenha um bom resultado até 3 componentes de frequência, ou seja, $M = 3$, observa-se que na Figura 25, que é o caso com 60dB de ruído, o algoritmo tem um erro baixo até 6 componentes de frequência, o que já é um resultado bem satisfatório para um sinal com ruído e apenas duas iterações, porém na Figura 25 onde o ruído é aumentado para 40dB observa-se que o algoritmo tem um erro baixo para até 5 componentes de frequência, o que ainda é satisfatório, dado que o esperado eram apenas 3 componentes e em 25 observa-se que quando o ruído é aumentado para 20dB o algoritmo tem um erro baixo até 4 componentes de frequência, que ainda é superior ao esperado para a técnica empregada. Conclui-se que o resultado para vários tipos de sinais e problemas é satisfatório para sinais reais com ou sem ruído, fica dentro do esperado pela técnica.

Figura 25 – MSE para sinal real com $K=41$ com duas iterações e ruídos de 20dB,40dB e 60dB



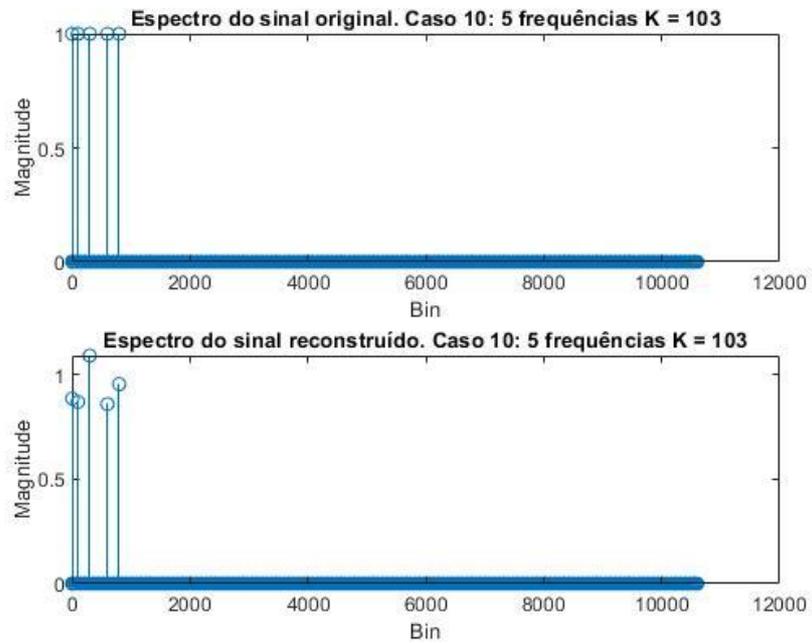
Fonte: Elaborado pelo autor.

4.6 Análise dos casos 10, 11, 12 e 13

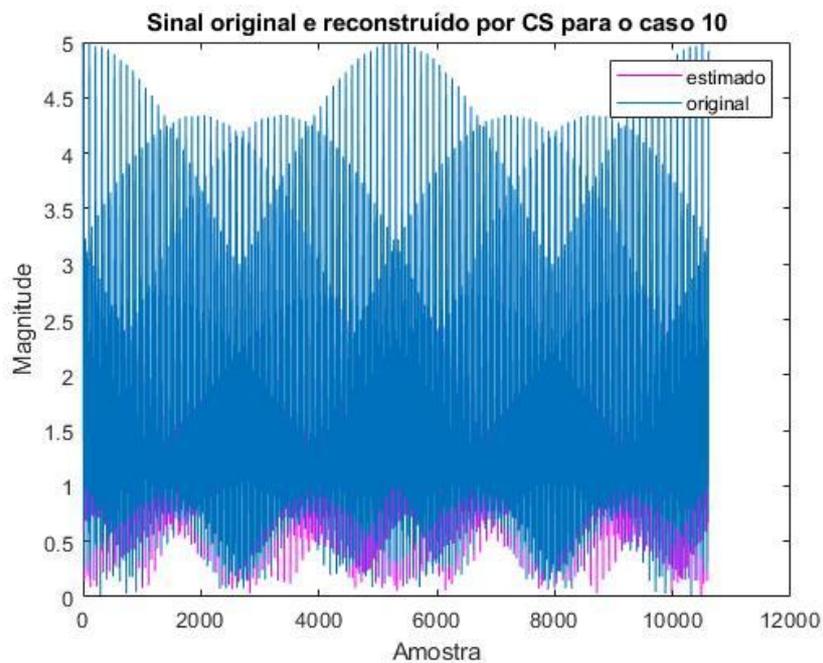
Nesta seção serão analisados os casos 10, 11, 12 e 13, teremos para os casos respectivamente $K = 103$, $K = 137$, $K = 179$ e $K = 251$, como visto na seção anterior a equação 4.3, obtida pela propriedade RIP, define o número máximo de componentes e neste caso serão respectivamente, 5, 6, 7 e 8. As amplitudes para todos os casos foi definida 1. Espera-se que o caso 13 tenha o melhor resultado, pois é o caso em que o sinal comprimido tem mais amostras. E é esperado que o caso 10 tenha o pior resultado, pois é o caso em que o sinal comprimido tem menos amostras.

A Figura 46 apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 10. Observa-se que o algoritmo identificou corretamente a posição dos *bins*, porém as amplitudes estimadas apresentaram um desvio. A Figura 26b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS. Foi calculada a relação entre a Energia do sinal reconstruído e a Energia do sinal original e foi obtido o valor de 0,2630, que significa que houve grande perda da energia do sinal na sua reconstrução.

Figura 26 – Resultados do Caso 10



(a) Espectro do sinal para o caso 10. Superior: sinal original; Inferior: sinal estimado

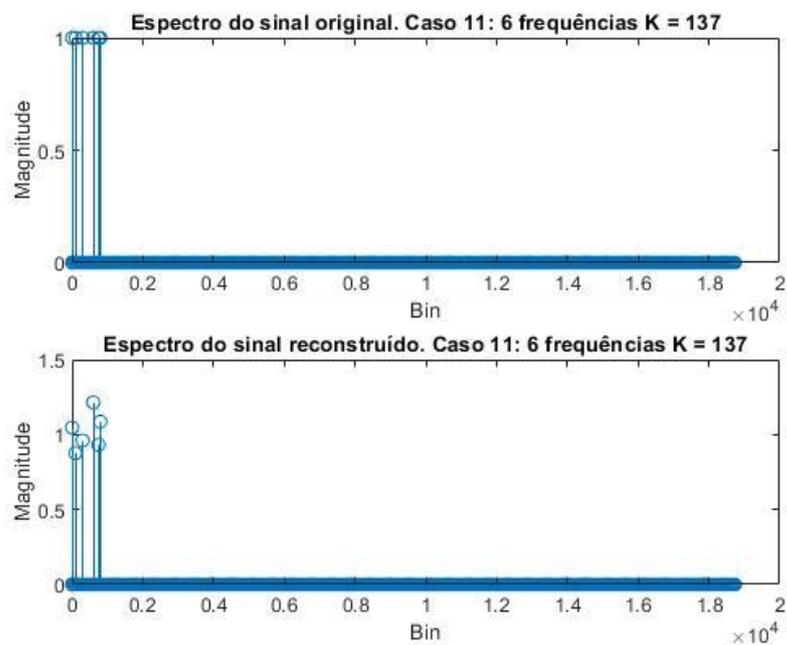


(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 10

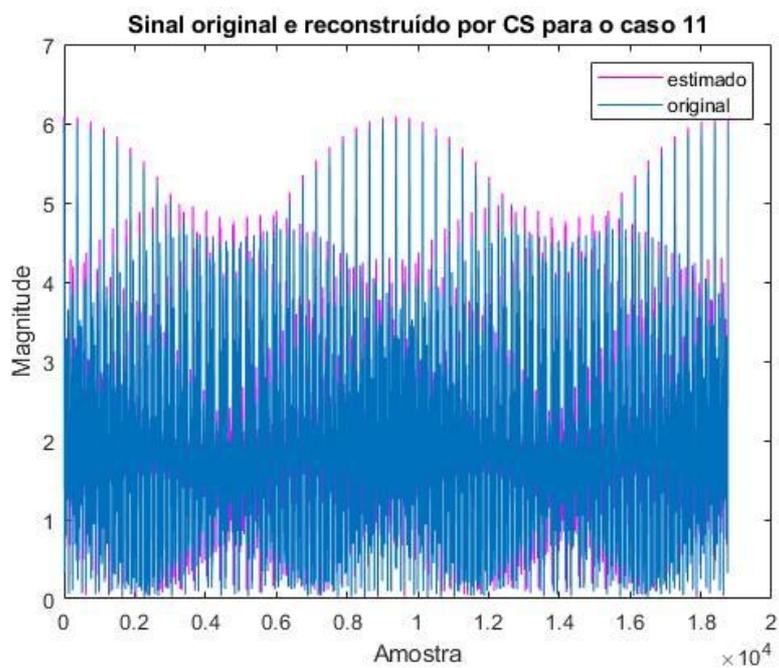
Fonte: Elaborado pelo autor.

A Figura 27a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 11, observa-se que em relação ao caso 10 já percebe-se que houve uma melhora na reconstrução. Todavia o algoritmo identificou corretamente a posição dos *bins*, mas também para esse caso as amplitudes estimadas apresentaram desvio, esse desvio foi menor que o caso anterior. A Figura 27b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS. Foi calculada a relação entre a Energia do sinal reconstruído e a Energia do sinal original e foi obtido o valor de 0,3692, que significa que houve grande perda da energia do sinal na sua reconstrução.

Figura 27 – Resultados do Caso 11



(a) Espectro do sinal para o caso 11. Superior: sinal original; Inferior: sinal estimado

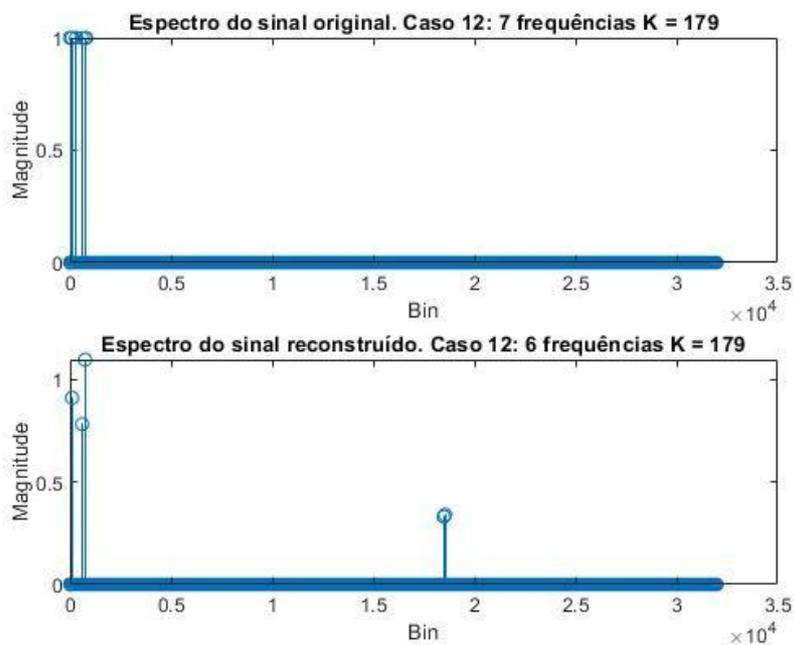


(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 11

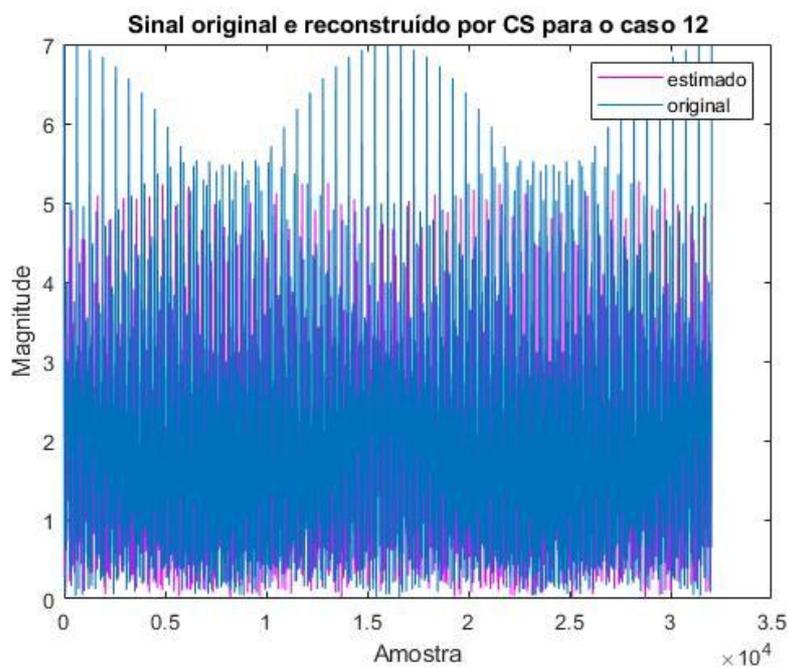
Fonte: Elaborado pelo autor.

A Figura 28a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 12. O algoritmo identificou corretamente a posição dos *bins*, as amplitudes estimadas apresentaram desvio menor que dos casos anteriores. A Figura 28b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS. Foi calculada a relação entre a Energia do sinal reconstruído e a Energia do sinal original e foi obtido o valor de 0,4849, que significa que houve grande perda da energia do sinal na sua reconstrução, ainda maior que 50%.

Figura 28 – Resultados do Caso 12



(a) Espectro do sinal para o caso 12. Superior: sinal original; Inferior: sinal estimado

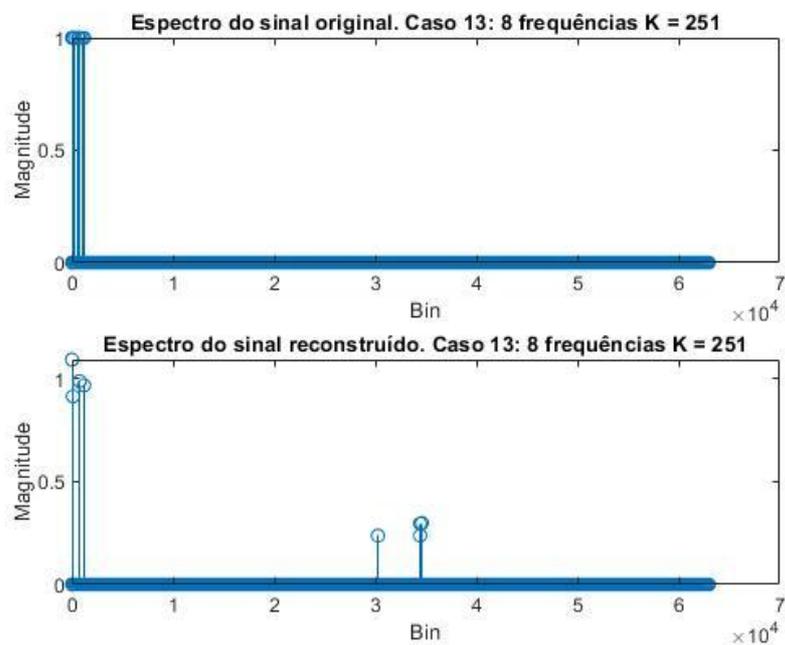


(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 12

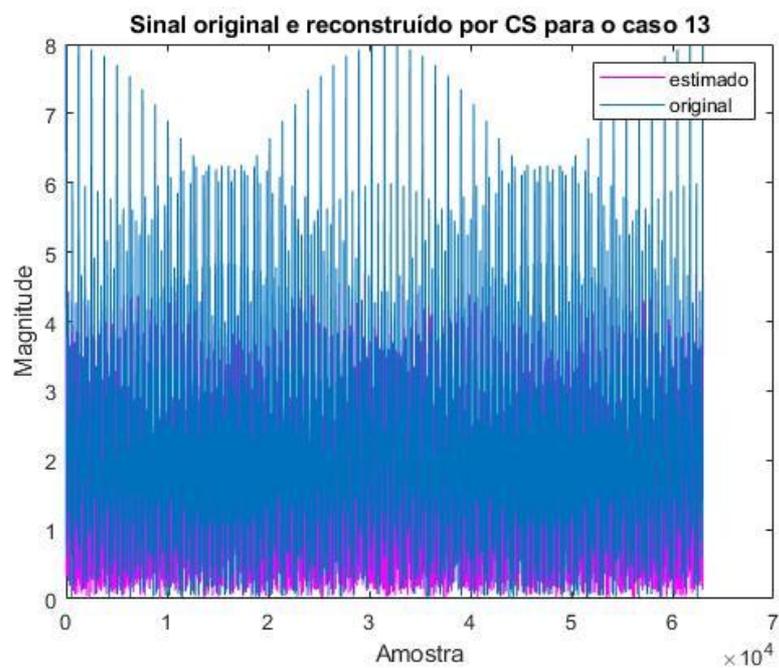
Fonte: Elaborado pelo autor.

A Figura 29a apresenta o espectro do sinal original e do sinal reconstruído utilizando CS para o caso 13. O algoritmo identificou corretamente a posição dos *bins*, as amplitudes estimadas apresentaram desvio menor que dos casos anteriores. A Figura 29b apresenta a magnitude dos sinais originais e do sinal estimado utilizando CS. Foi calculada a relação entre a Energia do sinal reconstruído e a Energia do sinal original e foi obtido o valor de 0,8524, e nesse caso a reconstrução foi bem feita, com pouca perda de energia. As relações das Energias desses casos estão demonstrados na tabela 4.

Figura 29 – Resultados do Caso 13



(a) Espectro do sinal para o caso 13. Superior: sinal original; Inferior: sinal estimado



(b) Magnitude do sinal original e estimado no domínio do tempo para o caso 13

Fonte: Elaborado pelo autor.

Tabela 4 – Erros dos casos estudados neste capítulo

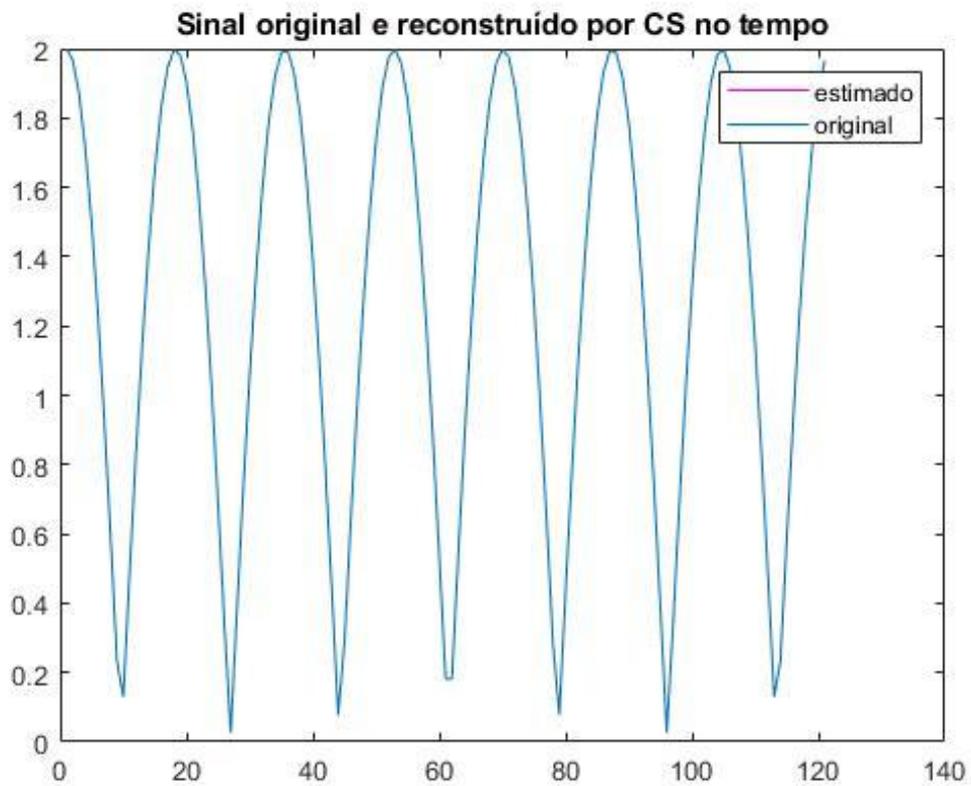
Caso	K	M	Frequências utilizadas	Amplitudes	Er/Eo
Caso 10	103	5	2 e 100 e 300 e 600 e 800	Todas 1	0.2630
Caso 11	137	6	2 e 100 e 300 e 600 e 800 e 750	Todas 1	0.3692
Caso 12	179	7	2 e 100 e 300 e 600 e 800 e 750 e 1000	Todas 1	0.4849
Caso 13	251	8	2 e 100 e 300 e 600 e 800 e 750 e 1000 e 1200	Todas 1	0.8524

Fonte: Elaborado pelo autor.

4.7 Análise dos casos com 100 interações

Foram feitas simulações para $K = 11, 41, 103, 137, 179e251$, as frequências foram geradas de forma aleatórias, respeitando o número máximo de componentes de frequências e foram feitas 100 interações para cada um desses casos. Pode ser observado que ao realizar 100 interações, pode-se obter simulações com uma boa reconstrução. Para os casos 1,2 e 3 temos $K = 11$, e para os casos 4, 5 e 6 o K temos $K = 41$.

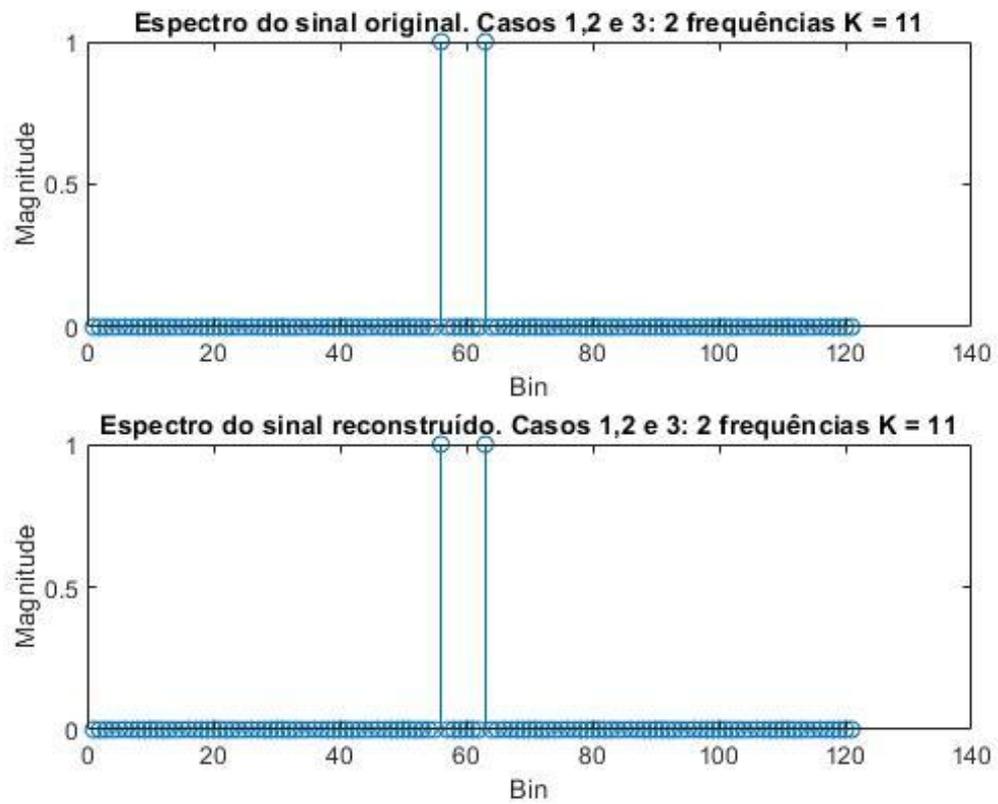
Figura 30 – Casos 1,2 e 3: Sinal original e sinal reconstruído no tempo



Fonte: Elaborado pelo autor.

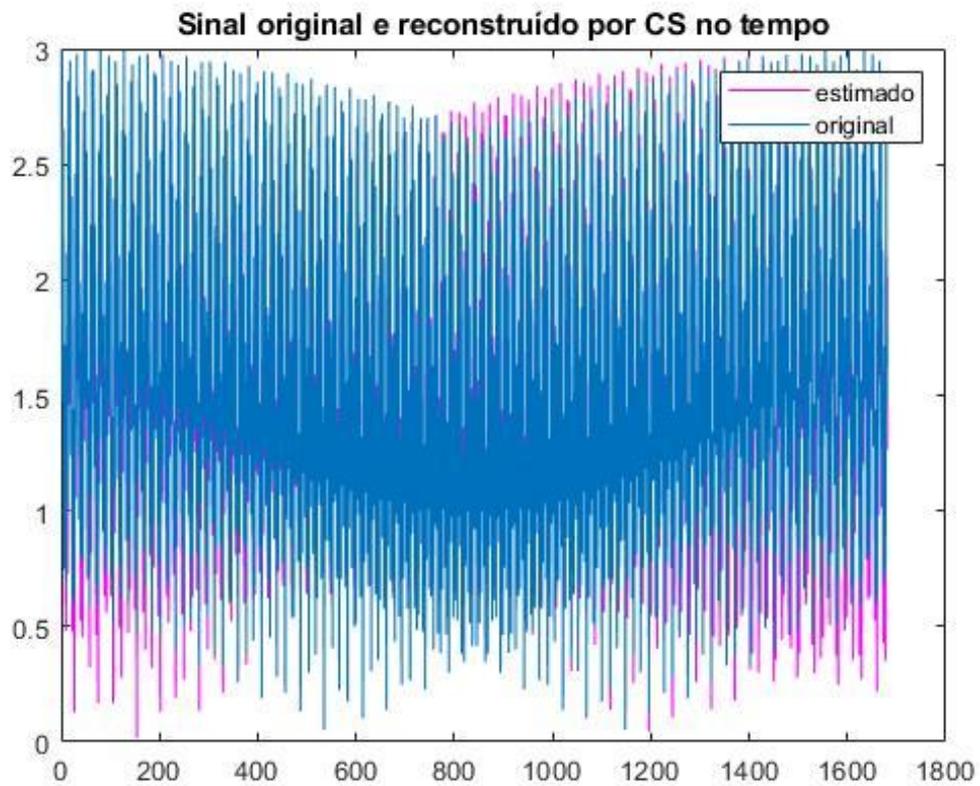
Observa-se na Figura 30, $K = 11$ o sinal original e o sinal reconstruído no tempo. E na Figura 31 tem-se o espectro na frequência do sinal original e reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 1,00 , e esse valor representa que o sinal foi reconstruído corretamente sem perdas de energia, e pode-se observar isso nas imagens, onde vê-se os bins das frequências localizados corretamente e também suas amplitudes.

Figura 31 – Casos 1,2 e 3: Sinal original e sinal reconstruído no espectro da frequência



Fonte: Elaborado pelo autor.

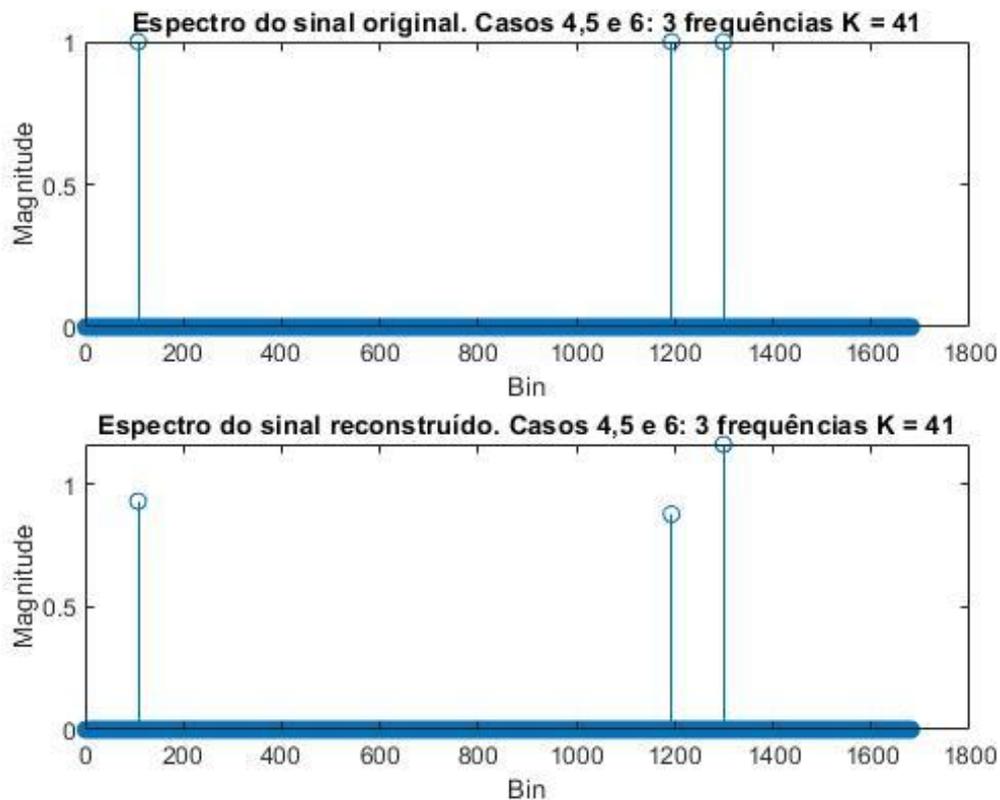
Figura 32 – Casos 4, 5 e 6: Sinal original e sinal reconstruído no tempo



Fonte: Elaborado pelo autor.

Observa-se na Figura 32, $K = 41$ o sinal original e o sinal reconstruído no tempo. E na Figura 33 tem-se o espectro na frequência do sinal original e reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9995, e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

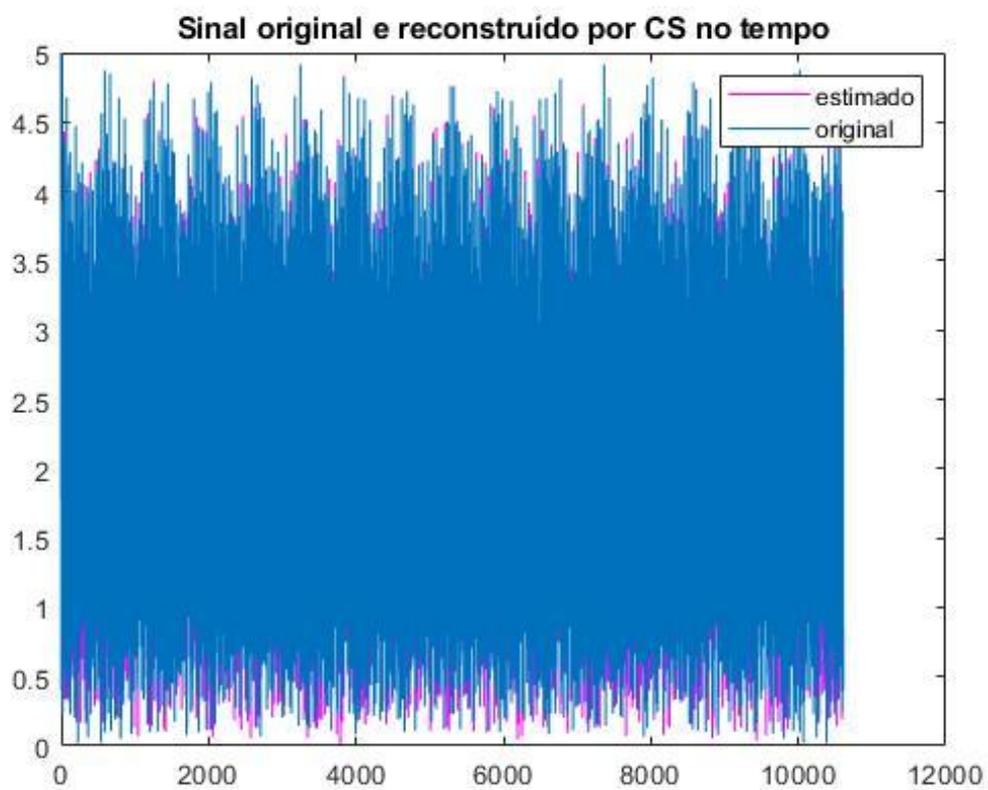
Figura 33 – Casos 4, 5 e 6: Sinal original e sinal reconstruído no espectro da frequência



Fonte: Elaborado pelo autor.

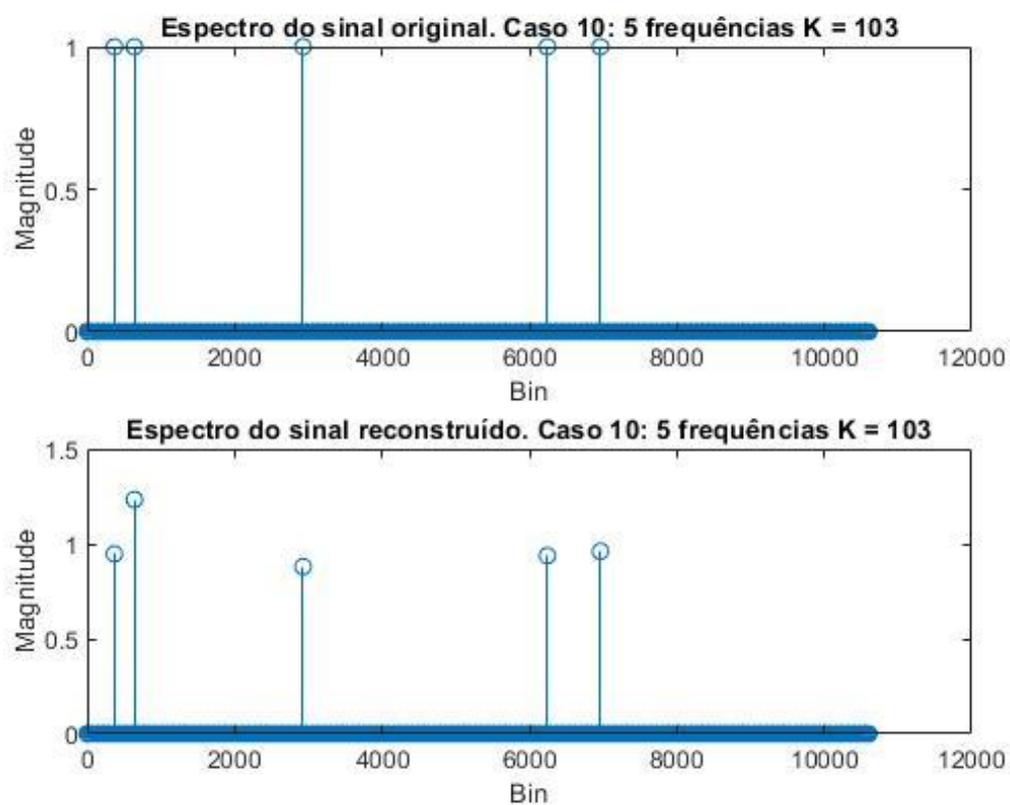
Para o caso 10, temos $K = 103$. Observa-se na Figura 34, o sinal original e o sinal reconstruído no tempo. E na Figura 35 tem-se o espectro na frequência do sinal original e reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9996, e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 34 – Caso 10: Sinal original e sinal reconstruído no tempo



Fonte: Elaborado pelo autor.

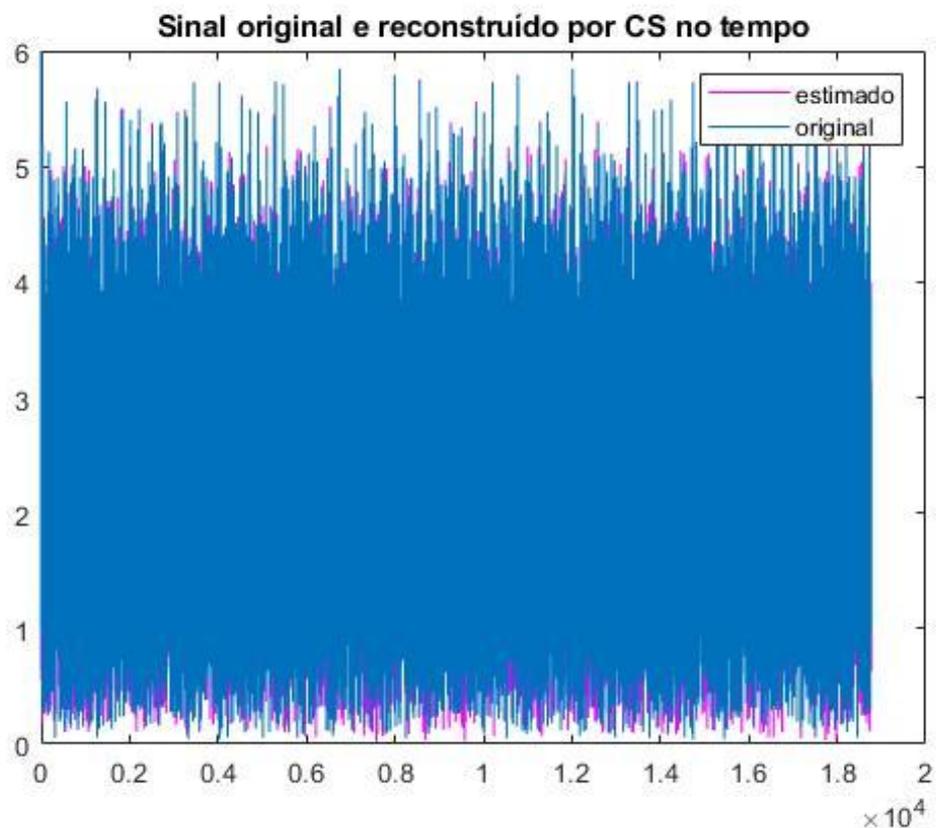
Figura 35 – Caso 10: Sinal original e sinal reconstruído no espectro da frequência



Fonte: Elaborado pelo autor.

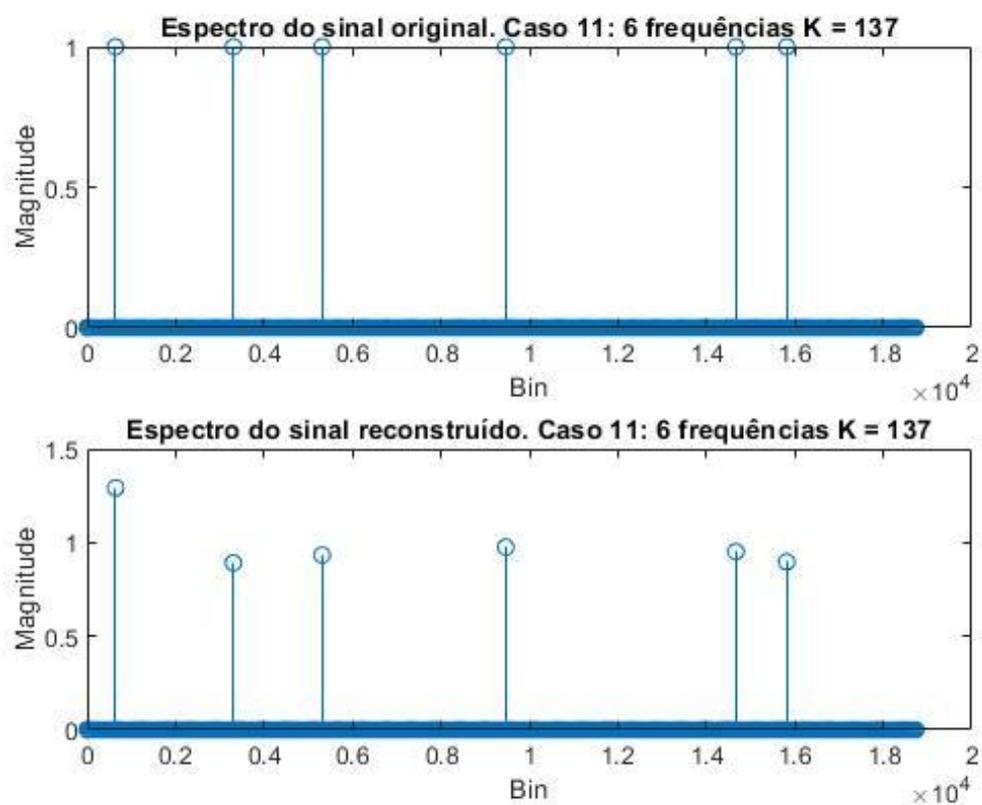
Para o caso 11, temos $K = 137$. Observa-se na Figura 36, o sinal original e reconstruído no tempo. E na Figura 37 tem-se o espectro na frequência do sinal original e reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9994 , e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 36 – Caso 11: Sinal original e sinal reconstruído no tempo



Fonte: Elaborado pelo autor.

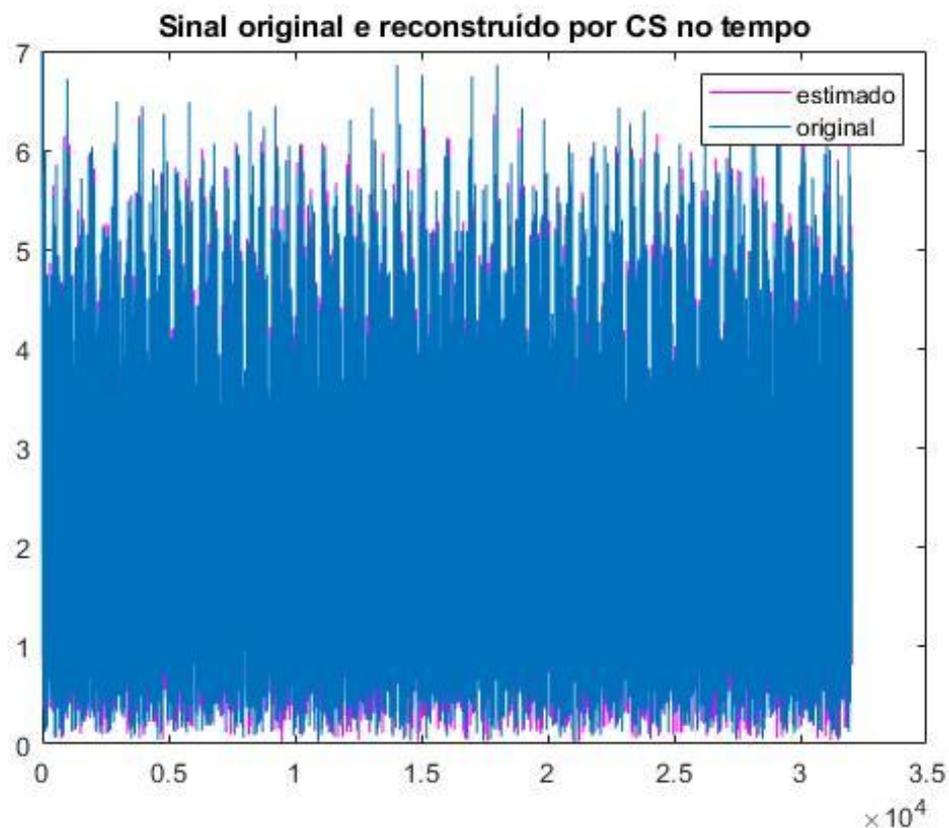
Figura 37 – Caso 11: Sinal original e sinal reconstruído no espectro da frequência



Fonte: Elaborado pelo autor.

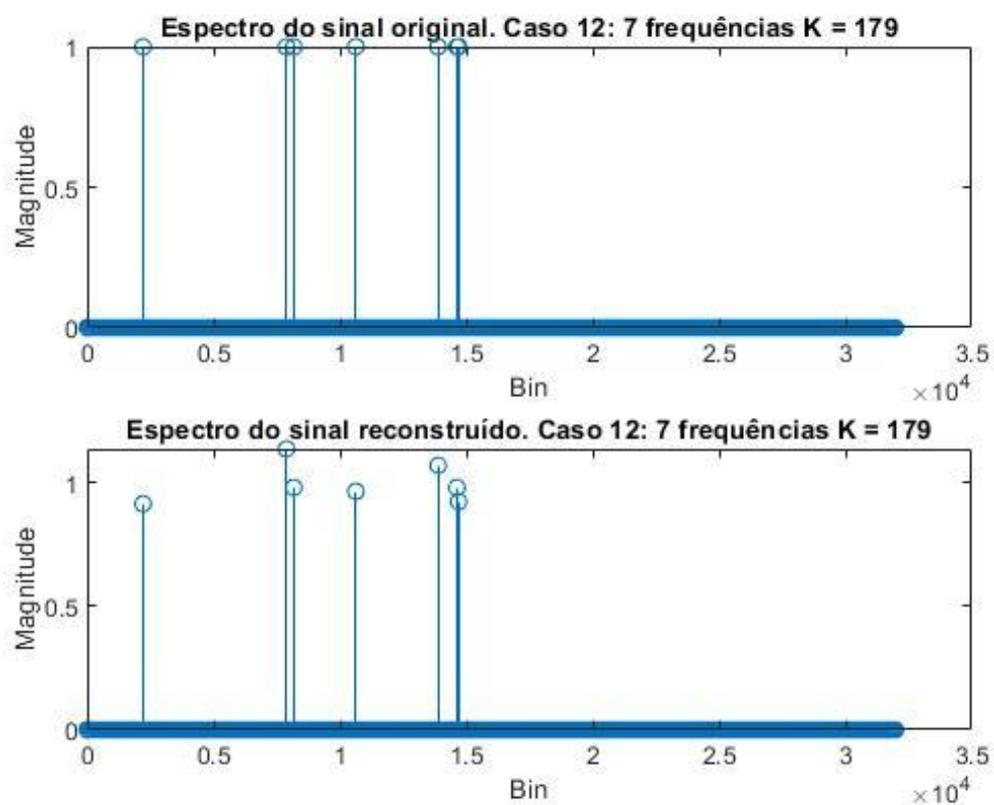
Para o caso 12, temos $K = 179$. Observa-se na Figura 38, o sinal original e reconstruído no tempo. E na Figura 39 tem-se o espectro na frequência do sinal original e o sinal reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9986 , e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 38 – Caso 12: Sinal original e sinal reconstruído no tempo



Fonte: Elaborado pelo autor.

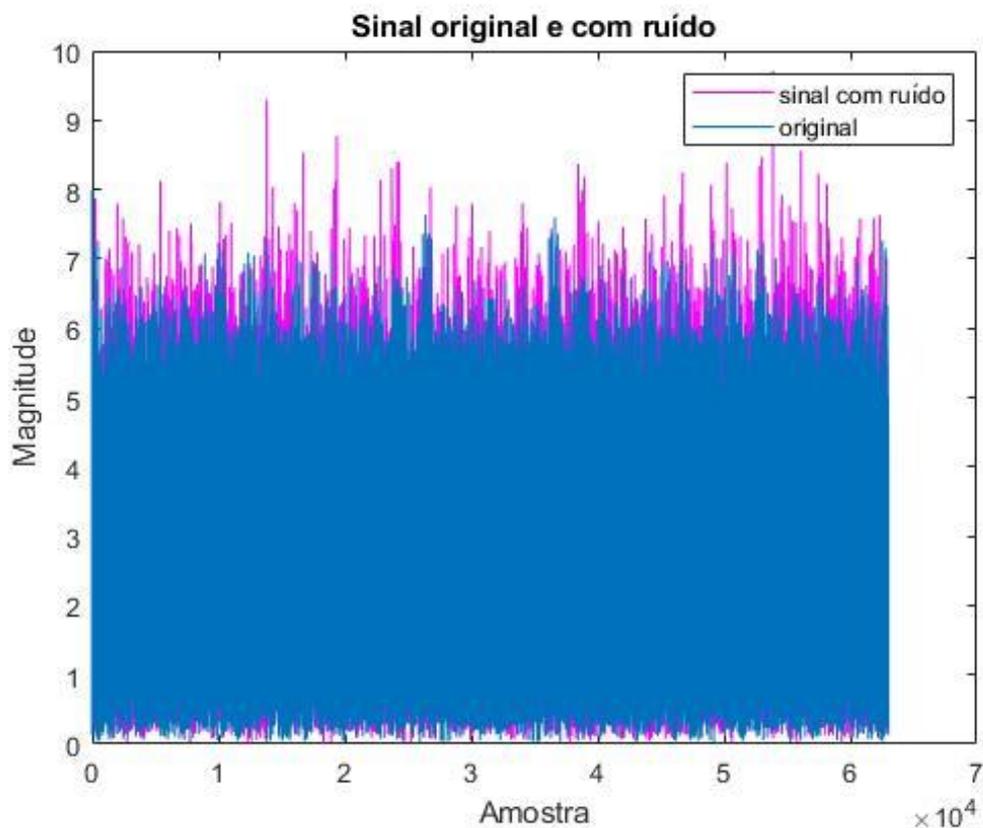
Figura 39 – Caso 12: Sinal original e sinal reconstruído no espectro da frequência



Fonte: Elaborado pelo autor.

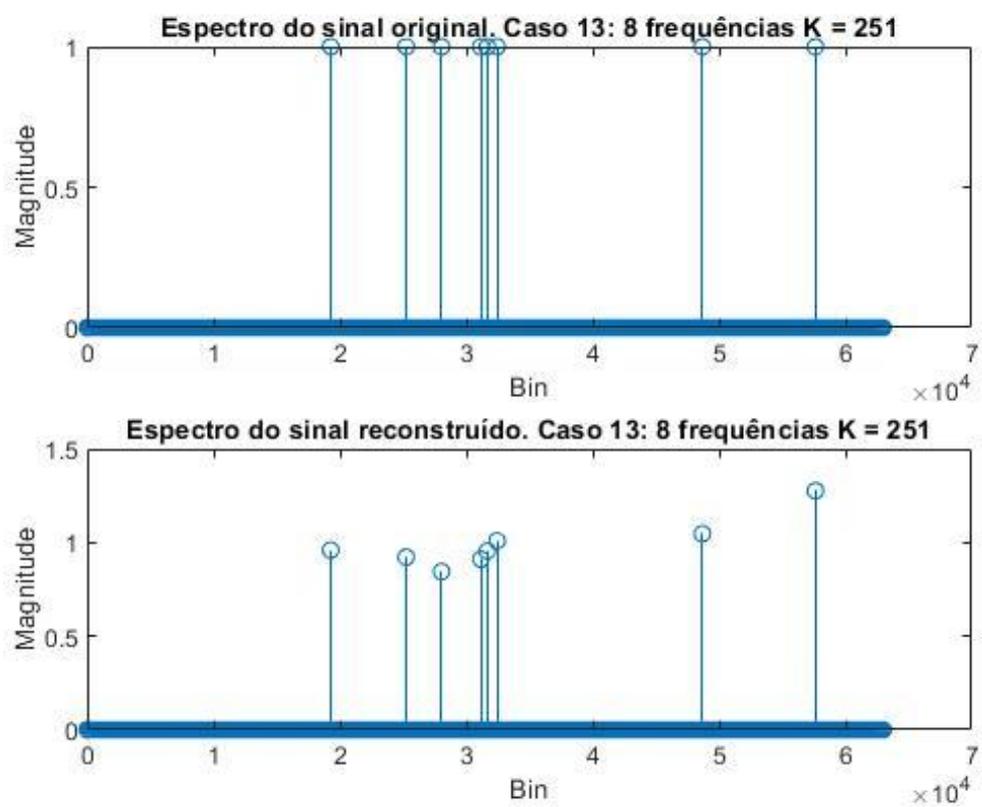
Para o caso 13, temos $K = 251$. Observa-se na Figura 40, o sinal original e reconstruído no tempo. E na Figura 41 tem-se o espectro na frequência do sinal original e reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9963 , e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 40 – Caso 13: Sinal original e sinal reconstruído no tempo



Fonte: Elaborado pelo autor.

Figura 41 – Caso 13: Sinal original e sinal reconstruído no espectro da frequência

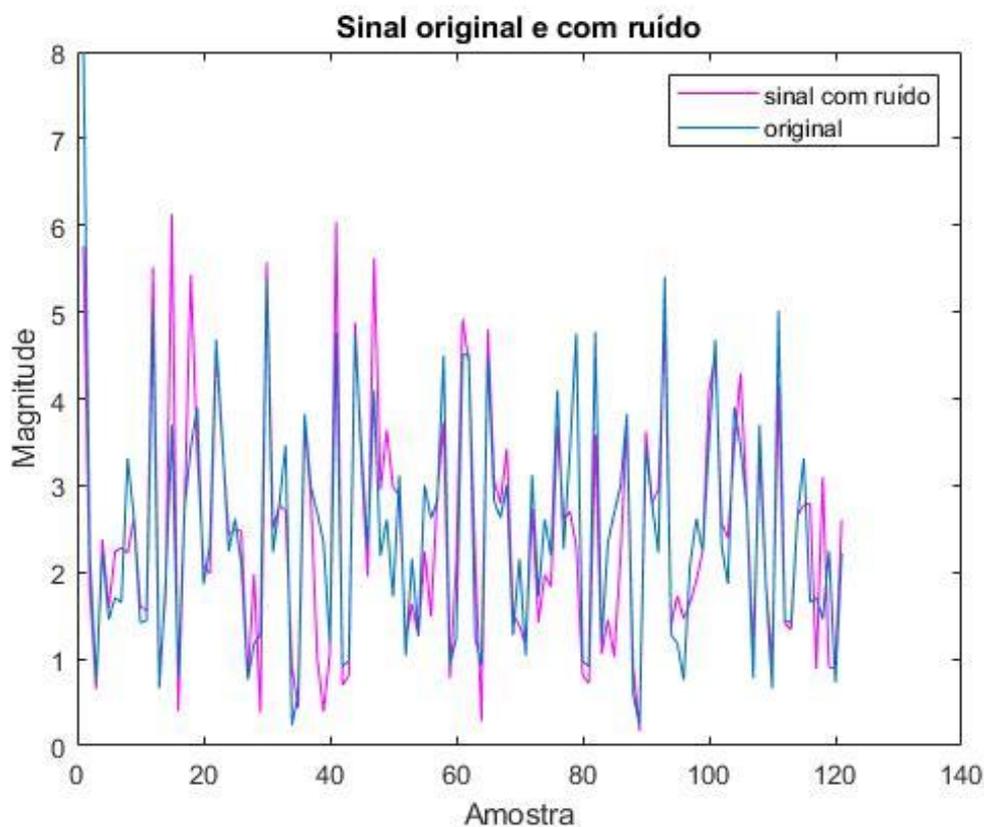


Fonte: Elaborado pelo autor.

4.8 Análises dos casos com 100 interações e o sinal original com ruído

Foram feitas simulações para $K = 11, 41, 103, 137, 179$ e 251 , com um sinal original com um ruído, as frequências foram geradas de forma aleatórias, respeitando o número máximo de componentes de frequências e foram feitas 100 interações para cada um desses casos. Pode ser observado que ao realizar 100 interações, pode-se obter simulações com uma boa reconstrução. Para os casos 1, 2 e 3 temos $K = 11$, e para os casos 4, 5 e 6 o K temos $K = 41$.

Figura 42 – Casos 1, 2 e 3: Sinal original e sinal original com ruído

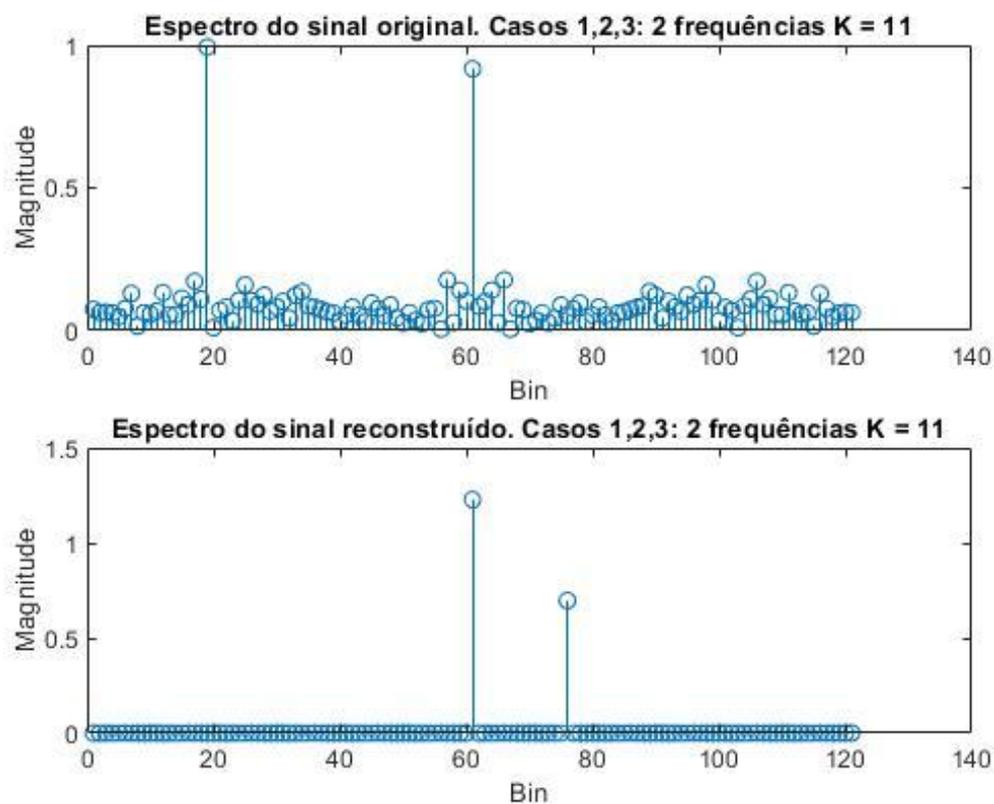


Fonte: Elaborado pelo autor.

Observa-se na Figura 42, $K = 11$ o sinal original no tempo, e o sinal original no tempo com a adição de ruído. E na Figura 43 tem-se o espectro na frequência

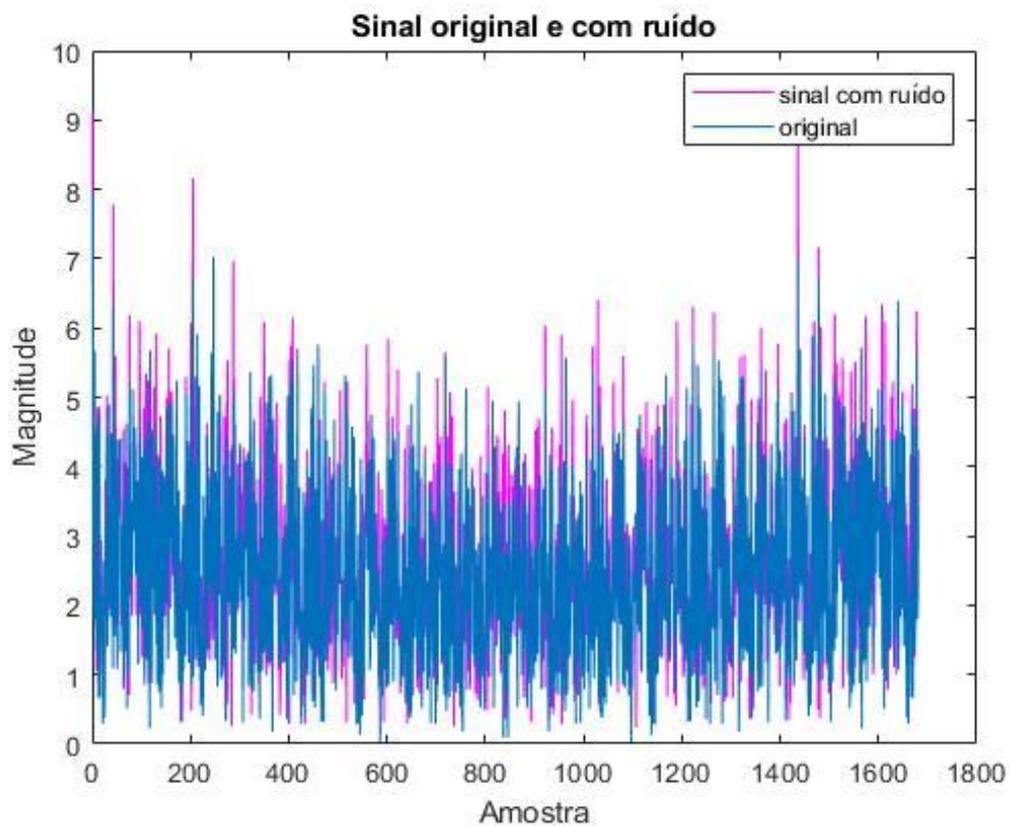
do sinal original com ruído e do sinal com ruído reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9909 , e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 43 – Casos 1,2 e 3: Sinal original e sinal com ruído no espectro da frequência



Fonte: Elaborado pelo autor.

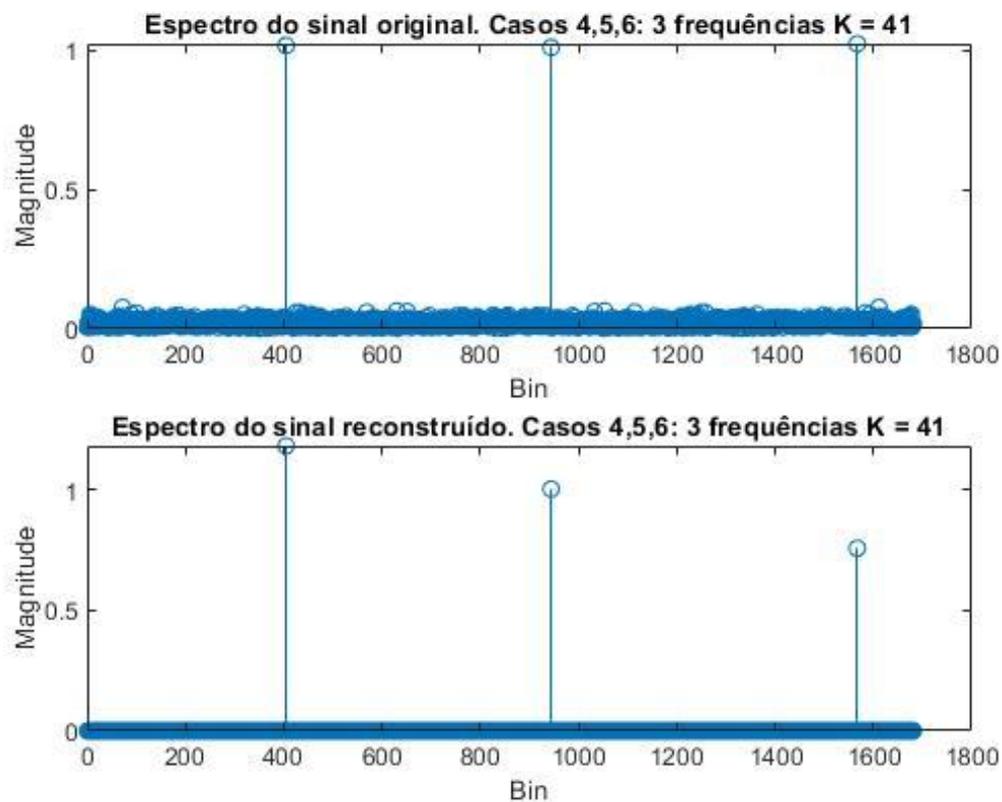
Figura 44 – Casos 4, 5 e 6: Sinal original e sinal original com ruído



Fonte: Elaborado pelo autor.

Observa-se na Figura 44, $K = 41$ o sinal original no tempo, e o sinal original no tempo com a adição de ruído. E na Figura 45 tem-se o espectro na frequência do sinal original com ruído e do sinal com ruído reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9941, e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

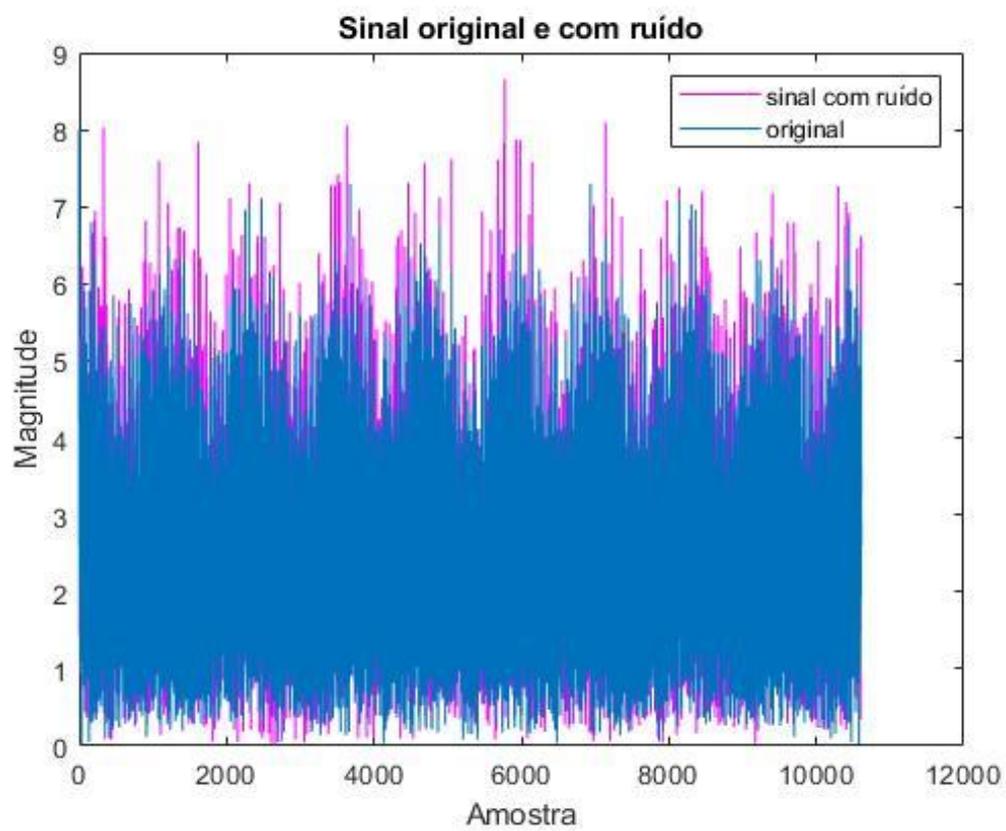
Figura 45 – Casos 4, 5 e 6: Sinal original e sinal com ruído no espectro da frequência



Fonte: Elaborado pelo autor.

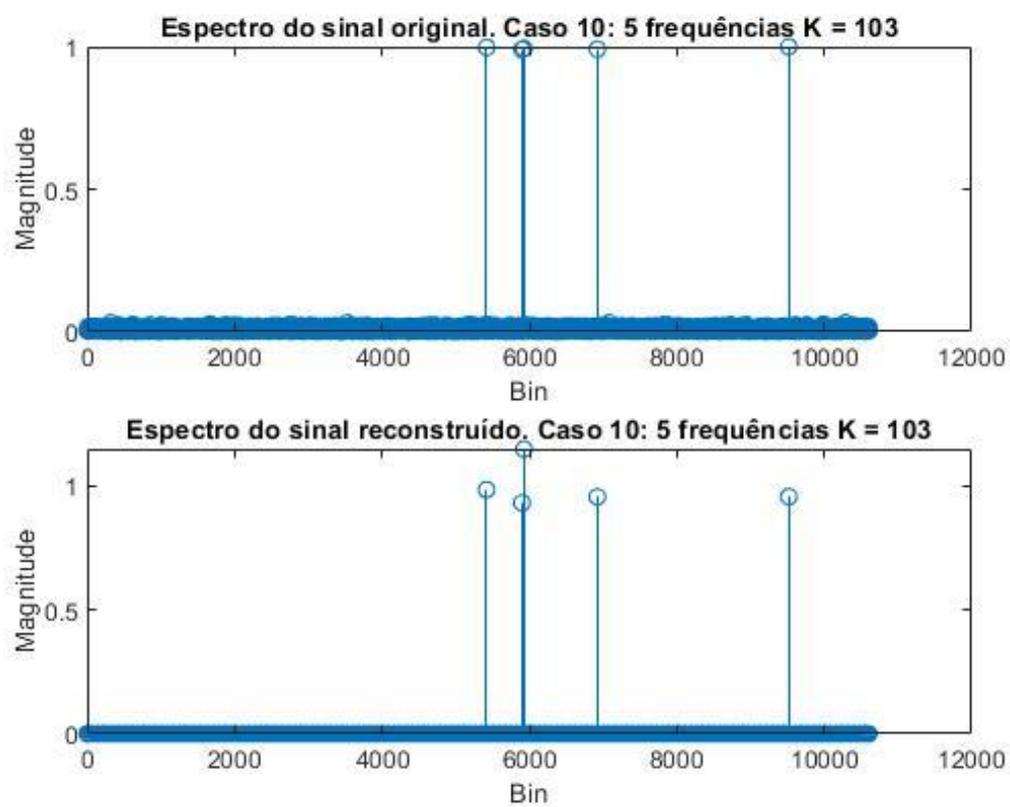
Para o caso 10, temos $K = 103$. Observa-se na Figura 46, o sinal original no tempo, e o sinal original no tempo com a adição de ruído. E na Figura 47 tem-se o espectro na frequência do sinal original com ruído e do sinal com ruído reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9926 , e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 46 – Caso 10: Sinal original e sinal original com ruído



Fonte: Elaborado pelo autor.

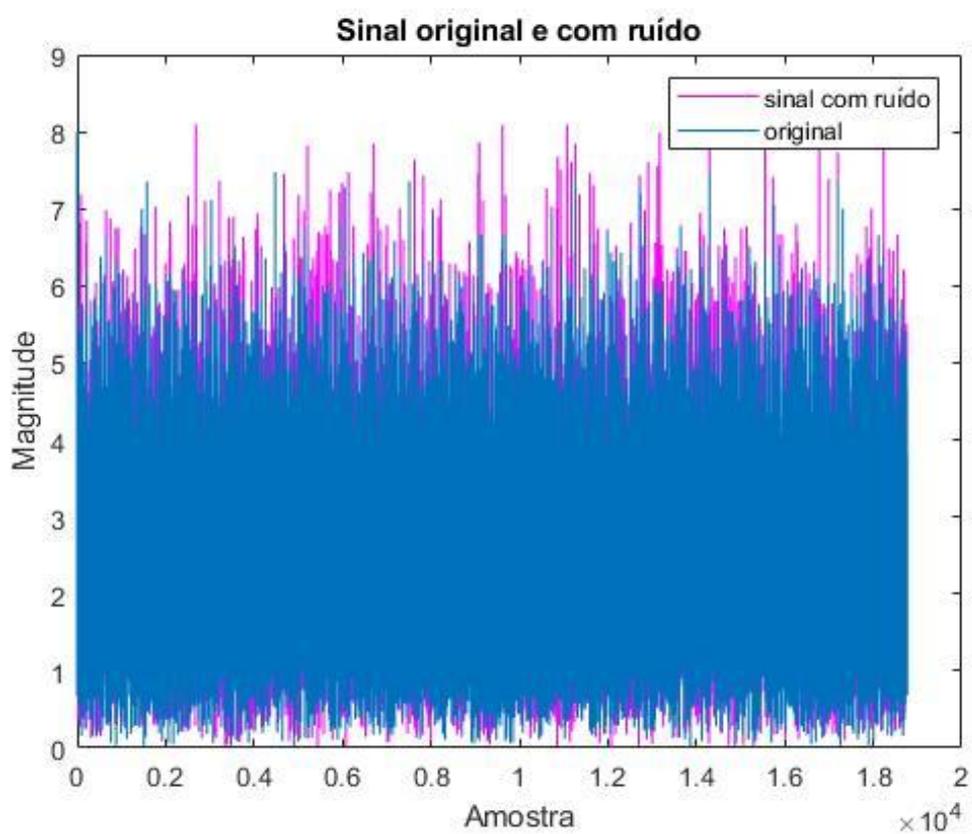
Figura 47 – Caso 10: Sinal original e sinal com ruído no espectro da frequência



Fonte: Elaborado pelo autor.

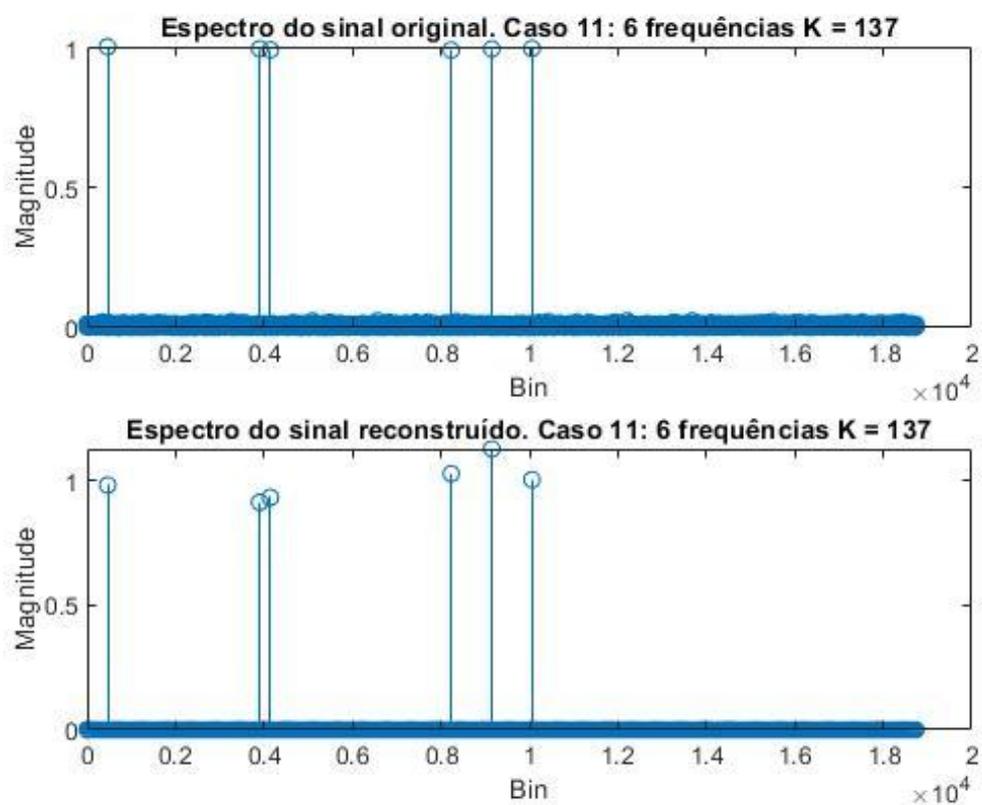
Para o caso 11, temos $K = 137$. Observa-se na Figura 48, o sinal original no tempo, e o sinal original no tempo com a adição de ruído. E na Figura 49 tem-se o espectro na frequência do sinal original com ruído e do sinal com ruído reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9987 , e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 48 – Caso 11: Sinal original e sinal original com ruído



Fonte: Elaborado pelo autor.

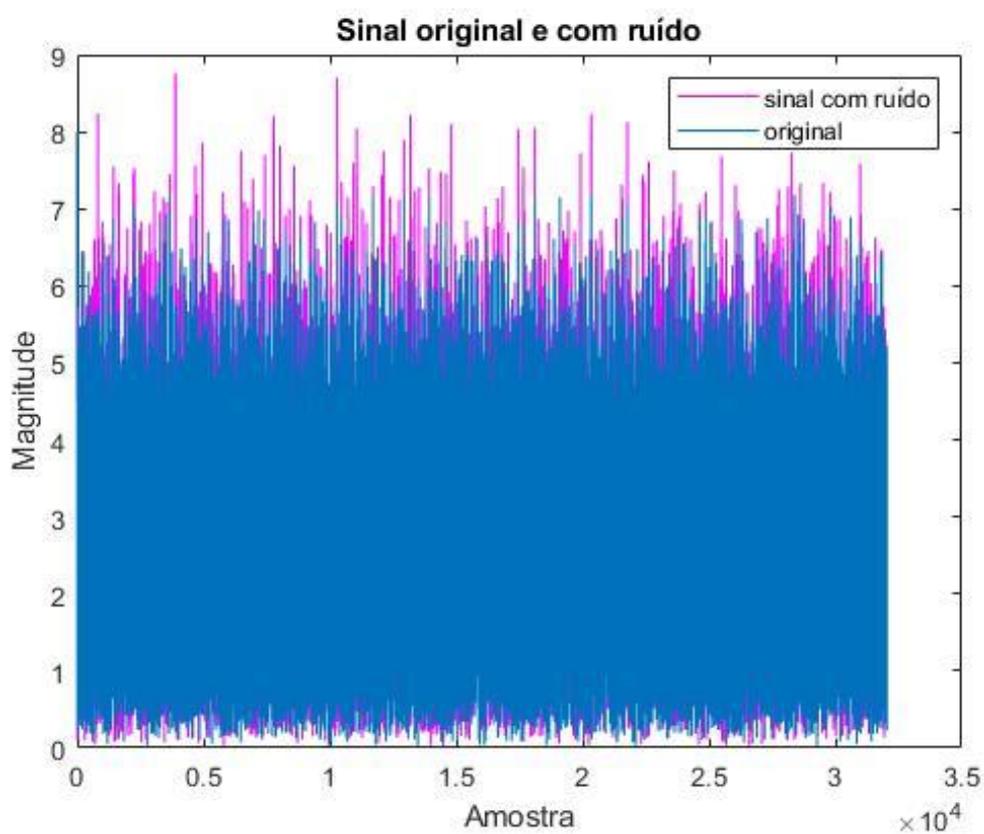
Figura 49 – Caso 11: Sinal original e sinal com ruído no espectro da frequência



Fonte: Elaborado pelo autor.

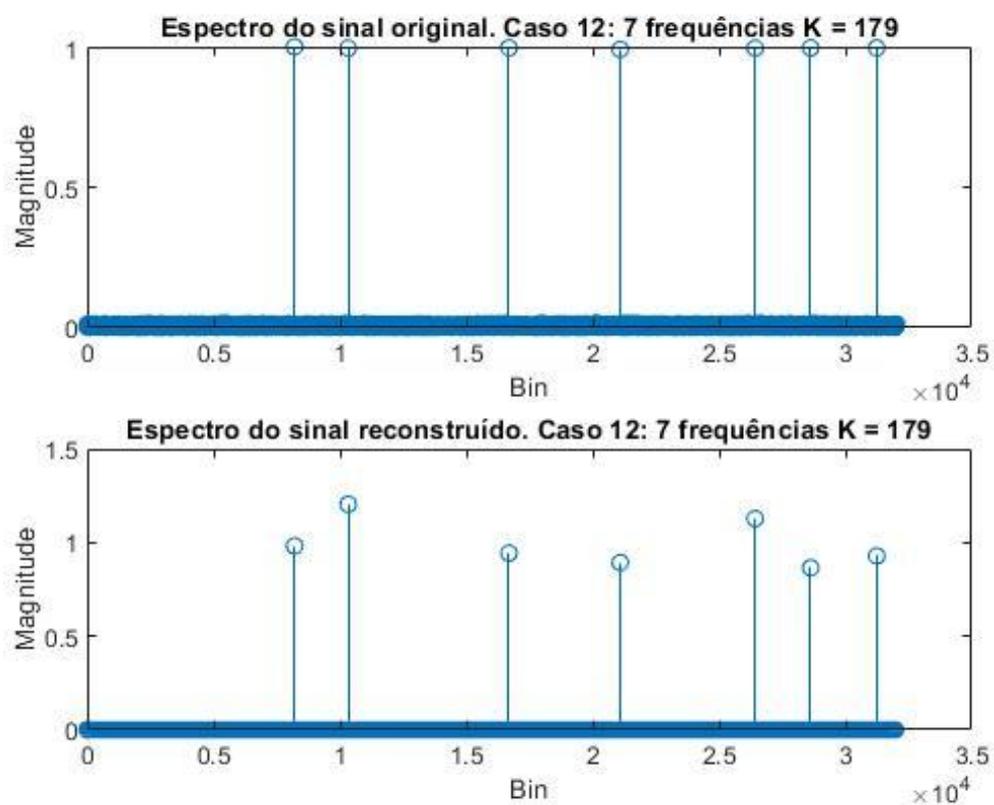
Para o caso 12, temos $K = 179$. Observa-se na Figura 50, o sinal original no tempo, e o sinal original no tempo com a adição de ruído. E na Figura 51 tem-se o espectro na frequência do sinal original com ruído e do sinal com ruído reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9996 , e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 50 – Caso 12: Sinal original e sinal original com ruído



Fonte: Elaborado pelo autor.

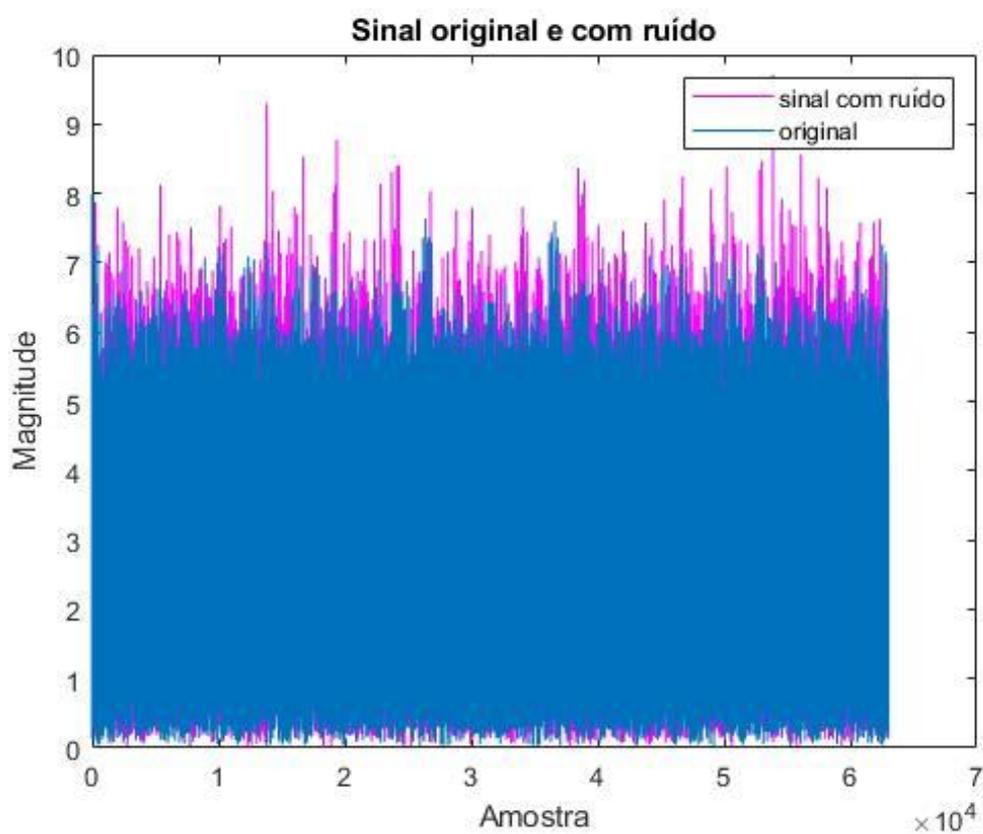
Figura 51 – Caso 12: Sinal original e sinal com ruído no espectro da frequência



Fonte: Elaborado pelo autor.

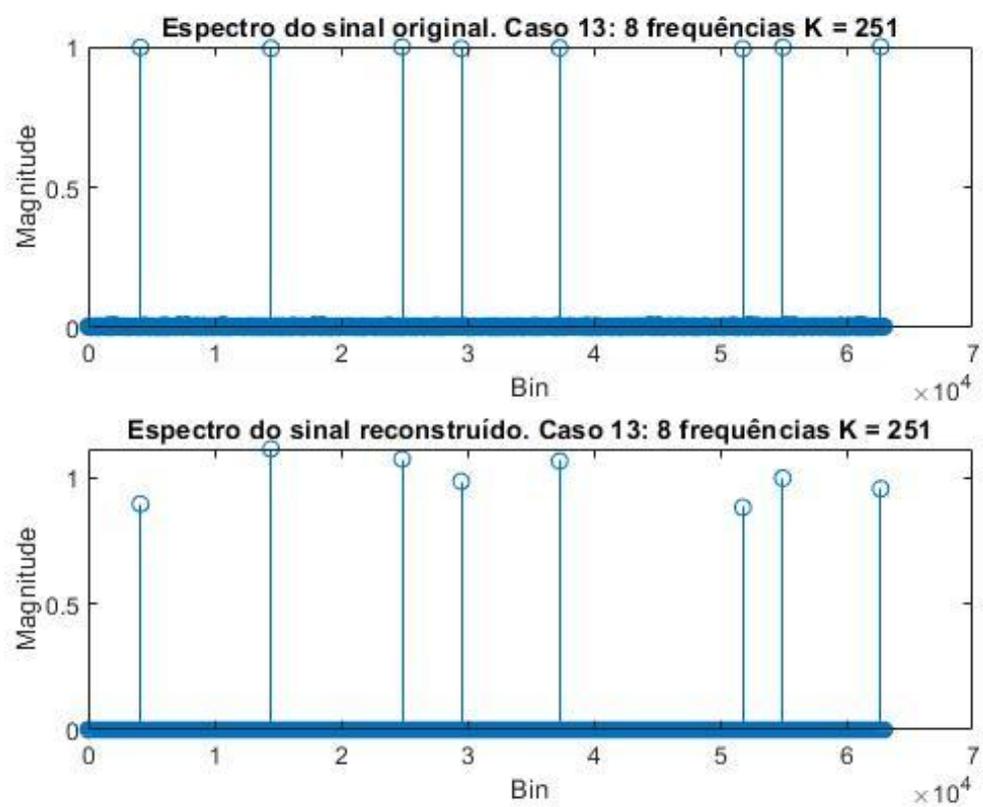
Para o caso 13, temos $K = 251$. Observa-se na Figura 52, o sinal original no tempo, e o sinal original no tempo com a adição de ruído. E na Figura 53 tem-se o espectro na frequência do sinal original com ruído e do sinal com ruído reconstruído. A Energia do sinal reconstruído sobre a Energia do sinal original E_r/E_o é de 0,9974 , e isso significa que o sinal foi reconstruído quase sem perdas, e pode-se observar que os bins das frequências foram localizados corretamente, com erro apenas em suas amplitudes.

Figura 52 – Caso 13: Sinal original e sinal original com ruído



Fonte: Elaborado pelo autor.

Figura 53 – Caso 13: Sinal original e sinal com ruído no espectro da frequência



Fonte: Elaborado pelo autor.

5 CONCLUSÕES

A metodologia proposta tem como objetivo a implementação de um algoritmo que visa diminuir a necessidade de memória de armazenamento em *hardwares* para sinais esparsos conhecido como *Compressive Sensing*. A aplicação do algoritmo proposto teve resultados em 13 casos distintos, onde percebe-se a reconstrução dos sinais originais. Destaca-se que quanto maior o comprimento do sinal de *chirp* (K) e menor o número de componentes de frequência do sinal (M), menor será o erro na reconstrução do sinal, e existe um máximo de componentes para cada K para que o sinal respeite a propriedade RIP e esse número máximo é definido pela equação 4.3.

Compressive Sensing tem como objetivo diminuir o armazenamento em memória do sinal processado, tornando o *hardware* mais barato. Para sinais elétricos, por exemplo, ao armazenarmos utilizando a técnica de processamento de sinais de Shannon-Nyquist teríamos 9409 amostras para um sinal esparsos com $K = 97$. Já para CS o mesmo sinal seria armazenado com apenas 97 amostras, o que demonstra um enorme ganho em memória e redução de custos, como mostrado no Caso 8.

Na análise do algoritmo de códigos de *chirp*, percebe-se a sua sensibilidade de performance em relação aos parâmetros. Apresentou-se como desafio a escolha dos parâmetros necessários para a sua melhor performance. Foram avaliados diversos K e suas análises para sinais com frequências fixas e variáveis. Neste trabalho foram realizados os mais diversos testes com sintonizações distintas, apresentando os melhores resultados nos casos apresentados, foram avaliados sinais com e sem ruído e sua comparação em tal método. Como proposta para resolução futura desse problema tem-se a aplicação de um método de otimização capaz de escolher esses parâmetros, tais como: otimizadores meta-heurísticos, não paramétricos e otimização inteira, adaptando em tempo real ao sinal, isto é, automatizar a busca pelos parâmetros das frequências e das amplitudes para atuar no maior número de casos possíveis.

Para o funcionamento da técnica é necessário os seguintes requisitos: um código de *chirp* cujo comprimento seja um número primo, um sinal esparsos e a propriedade RIP deve ser respeitada. Com essas exigências o trabalho proposto atendeu as expectativas, sendo publicado um artigo na área de análise de sinais de

potência, onde aplicou-se o algoritmo em sinais de potência [Melo et al. 2020].

Para trabalhos futuros sugere-se aplicar o método de CS utilizando a matriz determinística com códigos de *chirp* em sinais reais elétricos com harmônicos e definir quais as melhores taxas a serem aplicadas.

6 ANEXOS

6.1 Código MATLAB

Neste anexo seguem os códigos utilizados nessa dissertação para desenvolvimento do problema proposto. No código a seguir é onde se demonstra como é um sinal de chirp, como é sua parte real e imaginária.

```

%% Código de MATLAB: Código de chirp
K=17;
alpha = 1;
r=[1:K-1 0];
m=[1:K-1 0];
l=0:K-1;
%% componentes de chirp
chirpcode=alpha*exp((1i*2*pi*m(7)*l/K)+((1i*2*pi*r(5)*(l.^2)))/K);
figure(1)
plot3(1:K,real(chirpcode),imag(chirpcode))
title('Código de \textit{chirp} de comprimento K')
zlabel('Valor do código de chirp')
xlabel('Parte real')
ylabel('Parte imaginária')
figure(2)
subplot(211);plot(1:K, real(chirpcode), 'b*-');grid on
title('(a) Parte real')
subplot(212);plot(1:K, imag(chirpcode), 'm*-');grid on
title('(b) Parte imaginária')

```

Já nessa parte do código é feito um sinal com duas componentes e é demonstrado suas partes reais e imaginárias.

```

component1=alpha*exp((1i*2*pi*m(7)*l/K));
component2=alpha*exp((1i*2*pi*r(5)*(l.^ 2)/K));
figure(3);

```

```

subplot(221);plot(real(component1), 'b*-');
title('(a) Parte real da componente 1 exp((1i*2*pi*m(7)*l/K))')
axis([1 17 -1 1]);grid on
subplot(213);plot(real(component2), 'b*-');
title('(b) Parte real da componente 2 exp((1i*2*pi*r(5)*(l.^2)/K))')
axis([1 17 -1 1]);grid on
subplot(223);plot(imag(component1), 'm*-');
title('(c) Parte imaginária da componente 1 exp((1i*2*pi*m(7)*l/K))')
axis([1 17 -1 1]);grid on
subplot(214);plot(imag(component2), 'm*-');
title('(d) Parte imaginária da componente 2 exp((1i*2*pi*r(5)*(l.^2)/K))')
axis([1 17 -1 1]);grid on

```

No código a seguir é demonstrada a FFT do sinal de *chirp*.

```

figure(4);
subplot(311);stem(0:K-1, abs(fft(component1)));
title('(a) DFT da componente exp((1i*2*pi*m(7)*l/K))'); grid on
subplot(312);stem(0:K-1, abs(fft(component2)));
title('(b) DFT da componente exp((1i*2*pi*r(5)*(l.^ 2)/K))'); grid on
subplot(313);stem(0:K-1, angle(fft(chirpcode)));
title('(c) DFT da componente exp((1i*2*pi*m(7)*l/K)+((1i*2*pi*r(5)*(l.^ 2)))/K)');
grid on

```

A próxima parte demonstra como se extrai a frequência de *chirp* r .

```

K=17;
alpha = 1;
r=[1:K-1 0];
m=[1:K-1 0];
l=[0:K-1];
%% Extraindo r(5) de um único \textit{chirp} code
T=3;

```

```

chirpcode=alpha*exp((1i*2*pi*m(7)*l/K)+((1i*2*pi*r(5)*(l.^2)))/K);
f_l=conj(chirpcode).*circshift(chirpcode', -T)';
figure(5);
subplot(311);stem(abs(fft(chirpcode)));
title('DFT of the chirp'); grid on
r2TmodK=alpha*exp((1i*2*pi*mod(r(5)*T*2, K)*l/K));
subplot(312);stem(abs(fft(f_l)));
title('DFT of f(l)'); grid on
subplot(313);stem(abs(fft(r2TmodK)));
title('DFT of exp((1i*2*pi*mod(r(5)*TT*2, K)*l/K))'); grid on

```

Esse código na sequência demonstra os sinais e suas componentes.

```

clc; clear all; close all
alpha = 1;
K = 17;
r = [1:K-1 0];
m = [1:K-1 0];
l = [1:K-1 0];
r1 = r(5);
r2 = r(8);
TT = 3;
s1 = 2;
s2 = 2;
mod(r1*TT*2, K)
mod(r2*TT*2, K)
chirpcode1=alpha*exp((1i*2*pi*m(7)*l/K)+((1i*2*pi*r1*(l.^2)))/K);
chirpcode2=alpha*exp((1i*2*pi*m(9)*l/K)+((1i*2*pi*r2*(l.^2)))/K);
r2TmodK_1=alpha*exp((1i*2*pi*mod(r1*TT*2, K)*l/K));
r2TmodK_2=alpha*exp((1i*2*pi*mod(r2*TT*2, K)*l/K));
mixchirp=s1*chirpcode1+s2*chirpcode2;
fsig=conj(mixchirp).*circshift(mixchirp', -TT)';
figure(6); subplot(421);stem(abs(fft(chirpcode1)))

```

```

subplot(422);stem(abs(fft(chirpcode2)))
subplot(423);stem(abs(fft(r2TmodK_1)))
subplot(424);stem(abs(fft(r2TmodK_2)))
subplot(413);stem(abs(fft(mixchirp)))
subplot(414);stem(abs(fft(fsig)))

```

A seguir tem-se o código que demonstra as variações de K.

```

clc;clear all;close all
T=3;
for j=1:2
figure(6+j);
for K=6*j:6*j+5
clear fout
r=[0 1:K-1 ];
for i=1:K
fout(i)=mod(2*r(i)*T, K)
end
subplot(2,3,K-6*j+1);plot(sort(fout),'p');
title(K);xlabel('r(i)');ylabel('mod(2*r(i)*3, K)');
axis([0 K 0 K]); grid on
end
end

```

O código de *chirp* é misturado e decifrado de acordo com o código a seguir.

```

clc; clear all; close all
alpha = 1;
K = 17;
r = [1:K-1 0];
m = [1:K-1 0];
l = [1:K-1 0];
r1 = r(5);

```

```

r2 = r(8);
TT = 3;
s1 = 2;
s2 = 2;
mod(r1*TT*2, K)
mod(r2*TT*2, K)
chirpcode1=alpha*exp((1i*2*pi*m(7)*l/K)+((1i*2*pi*r1*(l.^2)))/K);
chirpcode2=alpha*exp((1i*2*pi*m(9)*l/K)+((1i*2*pi*r2*(l.^2)))/K);
r2TmodK_1=alpha*exp((1i*2*pi*mod(r1*TT*2, K)*l/K));
r2TmodK_2=alpha*exp((1i*2*pi*mod(r2*TT*2, K)*l/K));
mixchirp=s1*chirpcode1+s2*chirpcode2;
fsig=conj(mixchirp).*circshift(mixchirp', -TT)';
% Dechirping
dechirp1=exp(-1i*2*pi*r1*l.^2/K).*mixchirp/K;
figure(7);stem(abs(fft(dechirp1))); title('Mistura decifrada de r1')

```

Os códigos de *chirp* determinísticos com a implementação de um sinal harmônico é demonstrado no código a seguir.

```

%% Códigos de chirp determinísticos
%% Implementação com um sinal harmônico
clc; clear all; close all
K=11; % 7, 11, 17, 29, 37, 47
l=0:K-1;
r=0:K-1;
TT=3;
for i=1:K
    bijest(i)=mod(2*r(i)*TT, K);
end
%% Fazendo a matriz Phi CS
alpha = 1;
m=0:K-1;
r=0:K-1;

```

```

l=(0:K-1)';
for ii=1:K % índice do r (fator de chirp)
    for i = 1:K % índice do m (bin)
        Phi(:,i+K*(ii-1))= alpha*exp((1i*2*pi*m(i)*l/K)+((1i*2*pi*r(ii)*(l.^2)))/K);
    end
end
end
%% Sinal harmônico
N=K^2;
l2=0:N-1;
amts=[1 1]; % 20 10];
frequs=[2 30]; % 30 40];
x=zeros(N,1);
for j=1: size(amts,2)
    x=x+(amts(j)*exp(-1i*2*pi*frequs(j)*l2/N))'; % sinal de entrada
end
%% Fazendo a transformada de Fourier

s=fft(x)/N;s2=s;

figure(1)
stem(abs(s))
title('Espectro de Magnitude de s')

%% Compressive Sensing
mixchirp=Phi*s2;

Phi_TT=circshift(Phi, -TT); %circulando as linhas; %
mixchirp_TT=Phi_TT*s2;

% figure;
% subplot(211);plot(1:K, real(mixchirp), 'b*-');grid on
% title('(a) Parte Real do mixchirp')

```

```

% subplot(212);plot(1:K, imag(mixchirp), 'm*-');grid on
% title('(b) Parte Imaginária do mixchip')

% mixchirp=mixchirp';
% mixchirp_TT=mixchirp_TT';

%% Recovery

samples=[];

for jj=1:size(ams,2)
%% Atualizando f_l
    fsig=conj(mixchirp).*mixchirp_TT;
%% Obtendo r_i e m_i
    Fsig=abs(fft(fsig)/K);
    [aa, r_max]=max(Fsig);      % o valor máximo é mod(2*r*TT,K)
    rf=r(find(bijest==r_max-1)) % encontrando o r

    figure(1)
    stem(Fsig)
    title('espectro de f: identificando r')

%% Decifrando
    dechirp=exp(-1i*2*pi*rf*l.^2/K).*mixchirp;
%% Obtendo x_i e m_i
    temp=fft(dechirp)/K;
    [aa, m_max]=max(abs(temp)); % localizando o máxim

    figure(1)
    stem(abs(temp));
    title('FFT do dechirp')

```

```

sampval=temp(m_max)

m_max=mod(m_max,K); %mod((K-m_max+2),K); %
if m_max==0
    m_max=K; %onde ta K era 17
end
samploc=rf*K+m_max; %onde ta K era 17
samples=[samples [samploc; sampval]];
%% Atualizando y_1
s2(samploc)=s2(samploc)-sampval;
figure(1)
stem(abs(s2))
title('s2 reduzido')

mixchirp=Phi*s2;
mixchirp_TT=Phi_TT*s2;

%    mixchirp=mixchirp';
%    mixchirp_TT=mixchirp_TT';
    jj=jj+1;
end

s_est=zeros(size(s));
for i=1:size(samples,2)
    s_est(real(samples(1,i)),1)=samples(2,i);
end
x_est=N*(ifft(s_est));

figure
subplot(211);stem(abs(s));xlabel('Bin');ylabel('Magnitude');
title('Espectro do sinal original. Caso 1: 2 frequências K = 11');
subplot(212); stem(abs(s_est));xlabel('Bin');ylabel('Magnitude');

```

```
title('Espectro do sinal reconstruído. Caso 1: 2 frequências K = 11')
```

```
figure
```

```
plot(1:K^2, abs(x_est(1:K^2)), 'm', 1:K^2, abs(x(1:K^2))); ;xlabel('Amostra');ylabe
```

```
title('Sinal original e reconstruído por CS para o caso 1')
```

```
legend('estimado', 'original')
```

```
err=(abs(x)-abs(x_est)).^2;
```

```
MSE = abs(mean(err))
```

```
figure
```

```
plot(abs(err))
```

```
title('abs(MSE)')
```

REFERÊNCIAS

- [Amaripadath et al. 2018]AMARIPADATH, D. et al. Measurement of supraharmomic emissions (2-150 khz) in real grid scenarios. In: IEEE. *2018 Conference on Precision Electromagnetic Measurements (CPEM 2018)*. [S.l.], 2018. p. 1–2.
- [Amini e Marvasti 2011]AMINI, A.; MARVASTI, F. Deterministic construction of binary, bipolar, and ternary compressed sensing matrices. *IEEE Transactions on Information Theory*, IEEE, v. 57, n. 4, p. 2360–2370, 2011.
- [Applebaum et al. 2009]APPLEBAUM, L. et al. Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery. *Applied and Computational Harmonic Analysis*, Elsevier, v. 26, n. 2, p. 283–290, 2009.
- [Baraniuk 2007]BARANIUK, R. A lecture on compressive sensing. *IEEE Signal processing magazine*, Citeseer, v. 24, n. 4, 2007.
- [Baraniuk et al. 2011]BARANIUK, R. et al. An introduction to compressive sensing. *Connexions e-textbook*, p. 24–76, 2011.
- [Baraniuk 2007]BARANIUK, R. G. Compressive sensing [lecture notes]. *IEEE signal processing magazine*, IEEE, v. 24, n. 4, p. 118–121, 2007.
- [Baron et al. 2005]BARON, D. et al. An information-theoretic approach to distributed compressed sensing. In: *Proc. 45rd Conference on Communication, Control, and Computing*. [S.l.: s.n.], 2005.
- [Basta e Morsi 2021]BASTA, B.; MORSI, W. Low and high order harmonic distortion in the presence of fast charging stations. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 126, p. 106557, 2021.
- [Boche et al. 2015]BOCHE, H. et al. Compressed sensing and its applications. *Compressed Sensing and Its Applications*, edited by H. Boche, G. Caire, R. Calderbank, M. Marz, G. Kutynick, and R. Mathar (Springer, 2017), Springer, p. 1–54, 2015.
- [Bodmann e Casazza 2011]BODMANN, B. G.; CASAZZA, P. G. A quantitative notion of redundancy for finite frames. *Applied and Computational Harmonic Analysis*, Elsevier, v. 30, n. 3, p. 348–362, 2011.

- [Bose e Ray-Chaudhuri 1960]BOSE, R. C.; RAY-CHAUDHURI, D. K. On a class of error correcting binary group codes. *Information and control*, Elsevier, v. 3, n. 1, p. 68–79, 1960.
- [Bourgain et al. 2011]BOURGAIN, J. et al. Explicit constructions of RIP matrices and related problems. *Duke Mathematical Journal*, Duke University Press, v. 159, n. 1, p. 145–185, 2011.
- [Brown e Erdős 1984]BROWN, W. G.; ERDÖS, P. Inverse extremal digraph problems. In: *Finite and infinite sets*. [S.l.]: Elsevier, 1984. p. 119–156.
- [Bryan e Leise 2013]BRYAN, K.; LEISE, T. Making do with less: an introduction to compressed sensing. *Siam Review*, SIAM, v. 55, n. 3, p. 547–566, 2013.
- [Calderbank e Howard 2010]CALDERBANK, R.; HOWARD, S. Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property. *IEEE journal of selected topics in signal processing*, IEEE, v. 4, n. 2, p. 358–374, 2010.
- [Candes et al. 2011]CANDES, E. J. et al. Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis*, Elsevier, v. 31, n. 1, p. 59–73, 2011.
- [Candes e Tao 2005]CANDES, E. J.; TAO, T. Decoding by linear programming. *IEEE transactions on information theory*, IEEE, v. 51, n. 12, p. 4203–4215, 2005.
- [Candes e Tao 2006]CANDES, E. J.; TAO, T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, IEEE, v. 52, n. 12, p. 5406–5425, 2006.
- [Christensen 2008]CHRISTENSEN, O. *Frames and bases: An introductory course*. [S.l.]: Springer Science & Business Media, 2008.
- [Christensen et al. 2003]CHRISTENSEN, O. et al. *An introduction to frames and Riesz bases*. [S.l.]: Springer, 2003. v. 7.

- [Cohen e Dahmen 2009]COHEN, A.; DAHMEN, W. Compressed sensing and best k-term approximation. *Journal of the American mathematical society*, v. 22, n. 1, p. 211–231, 2009.
- [Cormode e Muthukrishnan 2006]CORMODE, G.; MUTHUKRISHNAN, S. Combinatorial algorithms for compressed sensing. In: SPRINGER. *International colloquium on structural information and communication complexity*. [S.l.], 2006. p. 280–294.
- [Davenport et al. 2012]DAVENPORT, M. A. et al. *Introduction to compressed sensing*. [S.l.]: Citeseer, 2012.
- [Deedell e CoSaMP 2008]DEEDEL, D.; COSAMP, J. T. “iterative signal recovery from incomplete and inaccurate samples. In: APPL. [S.l.], 2008.
- [DeVore 2007]DEVORE, R. A. Deterministic constructions of compressed sensing matrices. *Journal of complexity*, Elsevier, v. 23, n. 4-6, p. 918–925, 2007.
- [Ding e Xing 2003]DING, C.; XING, C. Several classes of $(2m-1, w, 2)$ optical orthogonal codes. *Discrete Applied Mathematics*, Elsevier, v. 128, n. 1, p. 103–120, 2003.
- [Donoho 2006]DONOHO, D. L. High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension. *Discrete & Computational Geometry*, Springer, v. 35, n. 4, p. 617–652, 2006.
- [Donoho e Elad 2003]DONOHO, D. L.; ELAD, M. Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 100, n. 5, p. 2197–2202, 2003.
- [Donoho e Tanner 2005]DONOHO, D. L.; TANNER, J. Neighborliness of randomly projected simplices in high dimensions. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 102, n. 27, p. 9452–9457, 2005.
- [Eldar e Kutyniok 2012]ELDAR, Y. C.; KUTYNIOK, G. *Compressed sensing: theory and applications*. [S.l.]: Cambridge university press, 2012.

- [Espín-Delgado et al. 2021]ESPÍN-DELGADO, Á. et al. Diagnosis of supraharmonics-related problems based on the effects on electrical equipment. *Electric Power Systems Research*, Elsevier, v. 195, p. 107179, 2021.
- [Falco e Varilone 2021]FALCO, P. D.; VARILONE, P. Statistical characterization of supraharmonics in low-voltage distribution networks. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 11, n. 8, p. 3574, 2021.
- [Gilbert et al. 2006]GILBERT, A. C. et al. Algorithmic linear dimension reduction in the l_1 norm for sparse vectors. *arXiv preprint cs/0608079*, 2006.
- [Guevara et al.]GUEVARA, D. P. et al. Assessment of harmonic distortion associated with pv penetration in a low voltage distribution network. In: IEEE. *2020 IEEE PES Transmission & Distribution Conference and Exhibition-Latin America (T&D LA)*. [S.l.]. p. 1–6.
- [Hendrickx e Olshevsky 2010]HENDRICKX, J. M.; OLSHEVSKY, A. Matrix p -norms are np -hard to approximate if $p \neq 1, 2$. *SIAM Journal on Matrix Analysis and Applications*, SIAM, v. 31, n. 5, p. 2802–2812, 2010.
- [Hoory e Linial 2006]HOORY, S.; LINIAL, N. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, v. 43, n. 4, p. 439–561, 2006.
- [Howard e Calderbank 2006]HOWARD, S. D.; CALDERBANK, A. R. The finite heisenberg-weyl groups in radar and communications. *EURASIP Journal on Advances in Signal Processing*, Springer, v. 2006, p. 1–12, 2006.
- [Howard e Calderbank 2008]HOWARD, S. D.; CALDERBANK, A. R. A fast reconstruction algorithm for deterministic compressive sensing using second order reed-muller codes. In: IEEE. *2008 42nd Annual Conference on Information Sciences and Systems*. [S.l.], 2008. p. 11–15.
- [Indyk 2008]INDYK, P. Explicit constructions for compressed sensing of sparse signals. In: *SODA*. [S.l.: s.n.], 2008. v. 8, p. 30–33.
- [Keevash 2014]KEEVASH, P. The existence of designs. *arXiv preprint arXiv:1401.3665*, 2014.

- [Khosravy et al. 2020]KHOSRAVY, M. et al. *Frontier applications of nature inspired computation*. [S.l.]: Springer Nature, 2020.
- [Khosravy et al. 2020]KHOSRAVY, M. et al. Particle swarm optimization of morphological filters for electrocardiogram baseline drift estimation. In: *Applied nature-inspired computing: algorithms and case studies*. [S.l.]: Springer, 2020. p. 1–21.
- [Khosravy et al. 2020]KHOSRAVY, M. et al. Recovery in compressive sensing: a review. *Compressive Sensing in Healthcare*, Elsevier, p. 25–42, 2020.
- [Kovacevic e Chebira 2007]KOVACEVIC, J.; CHEBIRA, A. Life beyond bases: The advent of frames (part i). *IEEE Signal Processing Magazine*, IEEE, v. 24, n. 4, p. 86–104, 2007.
- [Li et al. 2012]LI, S. et al. Deterministic construction of compressed sensing matrices via algebraic curves. *IEEE Transactions on Information Theory*, IEEE, v. 58, n. 8, p. 5035–5041, 2012.
- [Lin 2004]LIN, S. *Costello. DJ Error control coding*. [S.l.]: Prentice-Hall, Inc, 2004.
- [Mallat 1999]MALLAT, S. *A wavelet tour of signal processing*. [S.l.]: Elsevier, 1999.
- [Melo et al. 2020]MELO, K. et al. Chirp code deterministic compressive sensing: analysis on power signal. In: *Information Technology and Intelligent Transportation Systems*. [S.l.]: IOS Press, 2020. p. 125–134.
- [Mendes et al. 2018]MENDES, T. M. et al. Supraharmonic analysis using subsampling. In: IEEE. *2018 18th International Conference on Harmonics and Quality of Power (ICHQP)*. [S.l.], 2018. p. 1–5.
- [Mendes et al. 2020]MENDES, T. M. et al. Comparative analysis of the measurement methods for the supraharmonic range. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 118, p. 105801, 2020.
- [Naidu e Murthy 2016]NAIDU, R. R.; MURTHY, C. R. Construction of binary sensing matrices using extremal set theory. *IEEE Signal Processing Letters*, IEEE, v. 24, n. 2, p. 211–215, 2016.

- [Pope 2009]POPE, G. *Compressive sensing: A summary of reconstruction algorithms*. Dissertação (Mestrado) — ETH, Swiss Federal Institute of Technology Zurich, Department of Computer . . . , 2009.
- [Qaisar et al. 2013]QAISAR, S. et al. Compressive sensing: From theory to applications, a survey. *Journal of Communications and networks*, KICS, v. 15, n. 5, p. 443–456, 2013.
- [Rani e Dhok 2018]RANI, M.; DHOK, S. B. A systematic review of compressive sensing: Concepts, implementations and applications. *IEEE Access*, IEEE, v. 6, p. 4875–4894, 2018.
- [Rödl 1985]RÖDL, V. On a packing and covering problem. *European Journal of Combinatorics*, Elsevier, v. 6, n. 1, p. 69–78, 1985.
- [Shaltiel 2004]SHALTIEL, R. Recent developments in explicit constructions of extractors. In: *Current Trends in Theoretical Computer Science: The Challenge of the New Century Vol 1: Algorithms and Complexity Vol 2: Formal Models and Semantics*. [S.l.]: World Scientific, 2004. p. 189–228.
- [Sharma et al. 2016]SHARMA, S. K. et al. Application of compressive sensing in cognitive radio communications: A survey. *IEEE communications surveys & tutorials*, IEEE, v. 18, n. 3, p. 1838–1860, 2016.
- [Vidyasagar 2019]VIDYASAGAR, M. *An introduction to compressed sensing*. [S.l.]: SIAM, 2019.
- [Zhang et al. 2010]ZHANG, H.-x. et al. Implementation of compressive sensing in ecg and eeg signal processing. *The Journal of China Universities of Posts and Telecommunications*, Elsevier, v. 17, n. 6, p. 122–126, 2010.
- [Zhang e Zhao 2012]ZHANG, S.; ZHAO, X. Robust facial expression recognition via compressive sensing. *Sensors*, Molecular Diversity Preservation International, v. 12, n. 3, p. 3747–3761, 2012.
- [Zhang et al. 1995]ZHANG, W. et al. Dead-time modifications to fast fourier transform power spectra. *The Astrophysical Journal*, v. 449, p. 930, 1995.