



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS - CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA

**Aprimorando o desempenho computacional
do MUSIC para localização em ambientes
internos em tempo real**

Rafael Pintar Alevato

Orientador: Prof. Richard Demo Souza, D.Sc.

Coorientador: Pedro Lemos, B.Sc.

FLORIANÓPOLIS
2022

RAFAEL PINTAR ALEVATO

**Aprimorando o desempenho computacional do
MUSIC para localização em ambientes internos em
tempo real**

Trabalho de Conclusão do Curso sub-
metido à Universidade Federal de
Santa Catarina, como requisito neces-
sário para obtenção do grau de Bacha-
rel em Engenharia Elétrica

Orientador:

Prof. Richard Demo Souza, D.Sc.

Coorientador:

Pedro Lemos, B.Sc.

FLORIANÓPOLIS
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Alevato, Rafael Pintar

Aprimorando o desempenho computacional do MUSIC para
localização em ambientes internos em tempo real / Rafael
Pintar Alevato ; orientador, Richard Demo Souza,
coorientador, Pedro Lemos, 2022.

65 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Elétrica, Florianópolis, 2022.

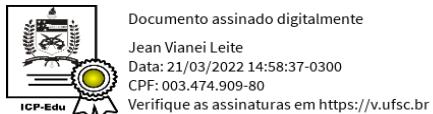
Inclui referências.

1. Engenharia Elétrica. 2. direção de chegada. 3. MUSIC.
4. desempenho computacional. 5. gradiente. I. Souza,
Richard Demo. II. Lemos, Pedro. III. Universidade Federal
de Santa Catarina. Graduação em Engenharia Elétrica. IV.
Título.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

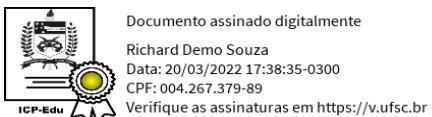
RAFAEL PINTAR ALEVATO

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Engenharia Elétrica, e aprovada, em sua forma final pelo Curso de Graduação em Engenharia Elétrica.

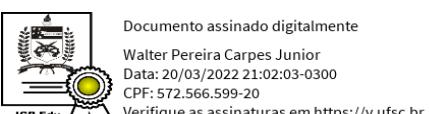


Prof. Jean Vianei Leite, D.Sc.
Coordenador do Curso de Graduação em Engenharia Elétrica
Universidade Federal de Santa Catarina - UFSC

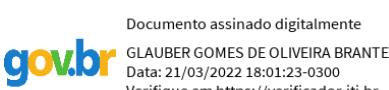
Banca Examinadora:



Prof. Richard Demo Souza, D.Sc.
Orientador
Universidade Federal de Santa Catarina - UFSC



Prof. Walter Pereira Carpes Júnior, D.Sc.
Universidade Federal de Santa Catarina - UFSC



Prof. Glauber Brante, D.Sc.
Universidade Tecnológica Federal do Paraná - UTFPR

Florianópolis, 18 de março de 2022

"If I have seen further it is by standing on the shoulders of Giants."
(Isaac Newton)

Agradecimentos

Agradeço a todos os familiares e amigos que me ajudaram nesta jornada. Agradeço em especial a minha mãe Valdeneia Pintar Elevato que sempre me apoiou e nunca perdeu a confiança em mim mesmo nos momentos mais difíceis. Agradeço também a minha namorada Tamilis Frois que me apoiou durante todo este trabalho.

Agradeço a todos os professores da graduação e em especial ao meu orientador Prof. Richard Demo Souza que me guiou e acreditou em minha capacidade de realizar este trabalho. Agradeço também aos meus companheiros da Khomp, em especial Willian Zanatta por ter confiado em mim e ao Elder Tramontin e Pedro Lemos que me apoiaram durante todo o trabalho.

Por fim, agradeço a todas as pessoas que passaram pela minha vida, pelos momentos bons e ruins. Somente com todas estas experiências que eu pude crescer como ser humano e ser quem sou hoje.

Resumo

A técnica de estimação de direção de chegada MUSIC oferece alta precisão e versatilidade, porém tem alto custo computacional. Este trabalho propõe implementar aprimoramentos para o método a fim de tornar viável sua utilização em tempo real para a localização de pessoas e ativos em ambientes internos. Também propõe comparar o tempo de execução dos aprimoramentos com outra técnica mais rápida: o ESPRIT. Foram implementados cinco diferentes aprimoramentos para a técnica MUSIC e testados em 10 mil amostras coletadas em um ambiente de escritório. Os algoritmos implementados reduziram o tempo de execução em até 220 vezes em comparação com o MUSIC de busca exaustiva. Ainda, descobriu-se uma possível maneira de identificar e descartar amostras deterioradas utilizando os algoritmos de aprimoramento do MUSIC. Por fim, apesar do algoritmo ESPRIT ainda ser 24 vezes mais rápido que o melhor MUSIC implementado neste trabalho, foi possível utilizar o MUSIC em um aplicativo de estimação de direção em tempo real.

Palavras-chave: direção de chegada; MUSIC; desempenho computacional; gradiente

Abstract

MUSIC is a direction of arrival estimation technique which offers high precision and versatility, although it has high computational cost. This work proposes to implement performance improvements to MUSIC algorithm, to locate assets indoor in real time. It also proposes to compare the runtime of the implemented improvements with another faster direction of arrival technique: ESPRIT. Five different MUSIC improvements were implemented and tested with 10 thousand samples collected in an office environment. The improvements reduced runtime up to 220 times comparing to plain MUSIC. Also, a possible way of identifying and disposing of deteriorated samples was discovered, utilizing one of the MUSIC improvements. Finally, although ESPRIT algorithm is 24 times faster than the best MUSIC implemented in this work, MUSIC technique was able to be used in a real time direction of arrival estimation application.

Keywords: direction of arrival; MUSIC; computational performance, gradient

Listas de ilustrações

Figura 1 – Relação entre direção de chegada e defasagem de um ULA [1]	27
Figura 2 – Relação entre direção de chegada e defasagem de um URA [2]	28
Figura 3 – Pacote do protocolo Bluetooth 5.1 para estimativa de DoA [3]	37
Figura 4 – Transmissor EFR32BG22 Thunderboard Kit [4].	38
Figura 5 – Módulo de recepção WSTK [5].	38
Figura 6 – Arranjo de antenas BRD4185A [6].	39
Figura 7 – Módulo de processamento Raspberry Pi 4 Model B [7].	39
Figura 8 – Pseudo espectro em 3D do MUSIC de uma amostra comum	42
Figura 9 – Curvas de nível do pseudo espectro do MUSIC de uma amostra comum	42
Figura 10 – Pseudo espectro em 3D do MUSIC de uma amostra comum para elevação de 0° a 360°	43
Figura 11 – Pseudo espectro em 3D do MUSIC de uma amostra com máximos locais claros	44
Figura 12 – Curvas de nível do pseudo espectro do MUSIC de uma amostra com máximos locais claros	45
Figura 13 – Pseudo espectro em 3D do MUSIC de uma amostra deteriorada	45
Figura 14 – Curvas de nível do pseudo espectro do MUSIC de uma amostra deteriorada	46
Figura 15 – Exemplo de busca grossa-fina para $p_g = 15^\circ$ e $p_f = 1^\circ$	47
Figura 16 – Variação de p_g para o método de busca grossa-fina.	52
Figura 17 – Variação de α para $p_g = 5^\circ$ do método Gradiente simples com α estático e adaptável.	53
Figura 18 – Variação de p_g com $\alpha = 0,1$ para o método Gradiente simples estático e $\alpha = 0,7$ para o método adaptável.	54
Figura 19 – Variação de β com $p_g = 5^\circ$ e $\alpha = 0,07$ para os métodos <i>Momentum</i> e <i>Nesterov Momentum</i>	55
Figura 20 – Variação de α com $p_g = 5^\circ$ e $\beta = 0,9$ para os métodos <i>Momentum</i> e <i>Nesterov Momentum</i>	55
Figura 21 – Variação de p_g com $\alpha = 0,07$ e $\beta = 0,9$ para os métodos <i>Momentum</i> e <i>Nesterov Momentum</i>	56

Listas de tabelas

Tabela 1 – Resultados experimentais de precisão do MUSIC, ESPRIT e API da Silabs para amostras de ponto fixo.	51
Tabela 2 – Resultados experimentais para escolha do parâmetro p_g para a busca grossa-fina.	56
Tabela 3 – As 5 execuções mais rápidas com diferentes parâmetros e com MAE < $0,3^\circ$ para o método do Gradiente simples com α estático.	57
Tabela 4 – As 5 execuções mais rápidas com diferentes parâmetros e com MAE < $0,3^\circ$ para o método do Gradiente simples com α adaptável.	57
Tabela 5 – As 5 execuções mais rápidas com diferentes parâmetros e com MAE < $0,3^\circ$ para o método <i>Momentum</i>	57
Tabela 6 – As 5 execuções mais rápidas com diferentes parâmetros e com MAE < $0,3^\circ$ para o método <i>Nesterov Momentum</i>	57
Tabela 7 – Melhores parâmetros de cada método usados nos testes completos de acurácia e de tempo de execução.	57
Tabela 8 – Execuções mais rápidas dos métodos baseados no Gradiente RMSE 1% < $0,3^\circ$	58
Tabela 9 – Resultado experimentais de acurácia e de tempo de execução dos métodos de aprimoramento do MUSIC.	58

Lista de abreviaturas e siglas

API - *Application Programming Interface* (interface de programação de aplicações)

CTE - *Constant Tone Extension* (extensão de tom constante)

DoA - *Direction of Arrival* (direção de chegada)

IQ - *In-phase and Quadrature* (em fase e quadratura)

MAD - *Mean Absolute Deviation* (desvio absoluto médio)

MAE - *Mean Absolute Error* (erro absoluto médio)

RMSE - *Root Mean Square Error* (raiz do erro quadrático médio)

SoC - *System on a Chip* (sistema em um chip)

ULA - *Uniform Linear Array* (arranjo uniforme linear)

URA - *Uniform Rectangular Array* (arranjo uniforme retangular)

Lista de símbolos

α	<i>learning rate</i>
β	<i>momentum</i>
p_g	Passo grosso
p_f	Passo fino
σ	<i>Standard deviation</i>
θ	Azimute
ϕ	Elevação
ψ	Defasagem
λ	Comprimento de onda

Sumário

1	INTRODUÇÃO	23
1.1	Objetivos	24
1.1.1	Objetivos Gerais	24
1.1.2	Objetivos Específicos	24
1.2	Organização	25
2	FUNDAMENTOS TEÓRICOS	27
2.1	Conceitos básicos sobre direção de chegada	27
2.2	Métodos de estimação DoA	29
2.2.1	MUSIC	30
2.2.2	ESPRIT	30
2.3	O método do Gradiente	34
2.3.1	<i>Momentum</i>	34
2.3.2	<i>Nesterov Momentum</i>	35
3	MATERIAIS E MÉTODOS	37
3.1	Materiais	37
3.1.1	O protocolo Bluetooth 5.1	37
3.1.2	Transmissor	37
3.1.3	Receptor	37
3.1.4	Módulo de processamento	38
3.1.5	Linguagens de programação	40
3.2	Amostragem	40
3.3	Implementação dos métodos DoA	41
3.4	Análise do pseudo espectro do MUSIC	41
3.5	Implementação dos aprimoramentos do MUSIC	44
3.5.1	Busca em dois estágios	44
3.5.2	Gradiente simples	46
3.5.3	<i>Momentum</i>	48
3.5.4	<i>Nesterov Momentum</i>	48
4	EXPERIMENTOS E RESULTADOS	49
4.1	Metodologia	49
4.1.1	Precisão do MUSIC e ESPRIT	49
4.1.2	Parâmetros dos aprimoramentos	49
4.1.3	Acurácia dos aprimoramentos e tempo de execução	50

4.2	Resultados	51
4.2.1	Precisão do MUSIC e ESPRIT	51
4.2.2	Análise dos parâmetros dos aprimoramentos	51
4.2.2.1	Busca grossa-fina	52
4.2.2.2	Gradiente simples	52
4.2.2.3	Momentum e Nesterov	53
4.2.3	Escolha dos melhores parâmetros	54
4.2.4	Acurácia dos aprimoramentos e tempo de execução	58
4.3	Validação em tempo real	59
5	CONCLUSÃO	61
	REFERÊNCIAS	63

1 INTRODUÇÃO

A localização de pessoas e ativos em ambientes internos está cada vez mais comum com o avanço das tecnologias sem fio. Entre as possíveis aplicações desta tecnologia estão: segurança dos trabalhadores, otimização de processos, rastreamento de produtos, rastreamento de equipamentos médicos, controle de acesso, verificação do cumprimento de regulações, entre outros. [8]. Técnicas tradicionais de localização em ambientes externos (e.g. GPS) não apresentam boa acurácia em ambientes internos [9], além de consumir muita energia [10]. Para melhorar estes pontos foram desenvolvidas tecnologias específicas para este propósito.

Uma das técnicas de localização com alta acurácia em ambientes internos [11] é a estimativa de direção de chegada (DoA - *Direction of Arrival*). Métodos de DoA se baseiam na detecção da direção de chegada por arranjos de antenas, denominados âncoras. Uma âncora recebe um sinal de referência de um transmissor, também chamado de *beacon*, e estima sua direção utilizando a defasagem do sinal recebido em cada antena do seu arranjo. Para obter a localização completa do *beacon*, são posicionadas várias âncoras em pontos fixos do ambiente e utilizando a informação de 2 ou mais âncoras, pode-se chegar a um valor estimado de posição por triangulação.

Um método de estimativa de DoA que oferece uma boa relação entre precisão e desempenho computacional é o ESPRIT (*Estimation of Signal Parameters via Rotational Invariance Technique*) [12], porém dependendo da geometria do arranjo de antenas pode não ser possível utilizá-lo [13]. Uma outra técnica mais versátil e com maior precisão [14] é o MUSIC (*Multiple Signal Classification*) [15]. Ela pode ser utilizada em arranjos pequenos e simples como o URA (*Uniform Rectangular Array*) de 8 antenas. O problema desta técnica em comparação com o ESPRIT é seu custo computacional. O MUSIC é um método baseada em decomposição em autovalores e envolve manipulação da matriz de autocorrelação do sinal recebido para obter subespaços correlacionados ao sinal e ao ruído [14]. Ao final deste processo, o MUSIC busca o pico em um pseudo espectro de resultados para cada passo de azimute ou elevação. Este processo pode ser lento em grandes arranjos de antenas.

Para solucionar este problema, algumas otimizações para o MUSIC foram propostas. O Root-MUSIC se baseia em achar raízes de polinômios ao invés de realizar a custosa busca pelo pico do pseudo espectro [16], porém ele é aplicável em arranjos unidimensionais (e.g. ULA - *Uniform Linear Array*). Este método pode ser estendido para atender arranjos com geometrias arbitrárias [17], mas foge do escopo deste trabalho devido complexidade de implementação.

A utilização de arranjos com antenas em duas dimensões (em comparação com uma

dimensão do ULA) permite a identificação de dois ângulos (azimute e elevação) e assim uma localização completa em um ambiente de três dimensões. Os URA são populares na localização em ambientes internos, no entanto a complexidade computacional do MUSIC fica ainda maior. Como é necessário buscar um par azimute-elevação ao invés de um único ângulo θ , a complexidade fica quadrática. Para uma resolução de 1° , o algoritmo precisa procurar o maior valor em 32,4 milhares de pares de azimute-elevação (360° de azimute e 90° de elevação). Para uma resolução de $0,1^\circ$, é necessário procurar em 3,24 milhões de pares. Ambas as resoluções não são aplicáveis em um sistema de tempo real. Um possível aprimoramento de dividir a busca do pico em duas fases foi proposta em [13]. Na primeira fase é realizada uma busca com uma resolução menor e na segunda é feita uma busca localizada com maior resolução em torno do máximo achado pela primeira busca. Com isso, o MUSIC se torna viável para aplicações em tempo real.

Este trabalho apresenta análises de possíveis melhorias computacionais para o MUSIC e a comparação de seus desempenhos computacionais com o algoritmo ESPRIT e com o MUSIC simples. Além da busca em duas etapas sugerida acima, este trabalho propõe alternativas mais rápidas para a busca da segunda etapa, agilizando ainda mais o algoritmo. Esse ganho de velocidade é importante, pois quanto maior a velocidade de processamento, maior o número de *beacons* que podem ser processados por segundo. Por fim, será verificado se estes aprimoramentos permitirão utilizar o MUSIC em tempo real em um aplicativo implementado em [1].

Agradeço a Khomp pela oportunidade e por ter me concedido o apoio e materiais necessários para realização deste projeto.

1.1 Objetivos

1.1.1 Objetivos Gerais

O objetivo deste trabalho é implementar e comparar o desempenho computacional de aprimoramentos para a técnica de DoA MUSIC em ambientes internos. Além disso, a precisão das técnicas MUSIC e ESPRIT foram comparadas em amostras feitas em um ambiente de escritório. Por fim, testar as técnicas em uma aplicação de estimativa de direção de chegada em tempo real.

1.1.2 Objetivos Específicos

- Implementar as técnicas de DoA MUSIC e ESPRIT.
- Comparar a precisão do MUSIC e do ESPRIT em um ambiente de escritório baseado em amostras de um transmissor fixo em diferentes pontos no espaço.

- Implementar diferentes tipos de aprimoramento para o método MUSIC a fim de melhorar seu desempenho computacional.
- Obter o pseudo espectro dos resultados do MUSIC para analisar os possíveis impactos sobre os algoritmos implementados.
- Analisar as melhores escolhas de parâmetros para os diferentes aprimoramentos desenvolvidos para o MUSIC utilizando amostras gerais coletadas em um ambiente de escritório.
- Comparar as acurárias e tempos de execução dos diferentes aprimoramentos do MUSIC implementados para amostras gerais coletadas em um ambiente de escritório.
- Comparar os tempos de execução de todos os métodos para todas as amostras coletadas.
- Validar todas as técnicas implementadas em um programa de estimação de DoA em tempo real.

1.2 Organização

O capítulo 2 discute os fundamentos teóricos dos métodos de estimação de DoA e também a base por trás dos algoritmos de otimização utilizados. Capítulo 3 apresenta os detalhes das implementações e os materiais utilizados. Capítulo 4 analisa os resultados obtidos nos experimentos. Por fim, o Capítulo 5 faz as conclusões gerais sobre o desempenho dos algoritmos implementados e discute trabalhos futuros.

2 FUNDAMENTOS TEÓRICOS

Este capítulo apresenta os conceitos fundamentais sobre as técnicas de estimação de direção de chegada e sobre os métodos de otimização Gradiente utilizados neste trabalho. Primeiramente, serão introduzidos detalhes básicos sobre como utilizar arranjos de antenas para estimar a DoA de um sinal. Em seguida, serão apresentados os métodos de estimação de DoA MUSIC e ESPRIT. Ao final, haverá uma breve descrição sobre o método do Gradiente.

2.1 Conceitos básicos sobre direção de chegada

Utilizando um arranjo de antenas é possível estimar a direção de chegada de um sinal com frente de onda plana local utilizando a frequência do equivalente banda base do sinal e as defasagens nas diferentes antenas do arranjo. Para isso, é necessário conhecer a geometria do arranjo e as distâncias entre as antenas. Também pode-se considerar o sinal enviado como plano nas antenas pois o transmissor está suficientemente distante (*far field*) [18]. Essa hipótese é aceitável no contexto deste trabalho devido ao tamanho do comprimento de onda do sinal transmitido. A Figura 1 ilustra o caso do ULA e a Figura 2 o caso do URA.

No caso geral, a diferença de fase entre as antenas depende de valores de azimute (θ) e elevação (ϕ) e pode ser representada por ψ_i , onde i é o i -ésimo elemento do arranjo.

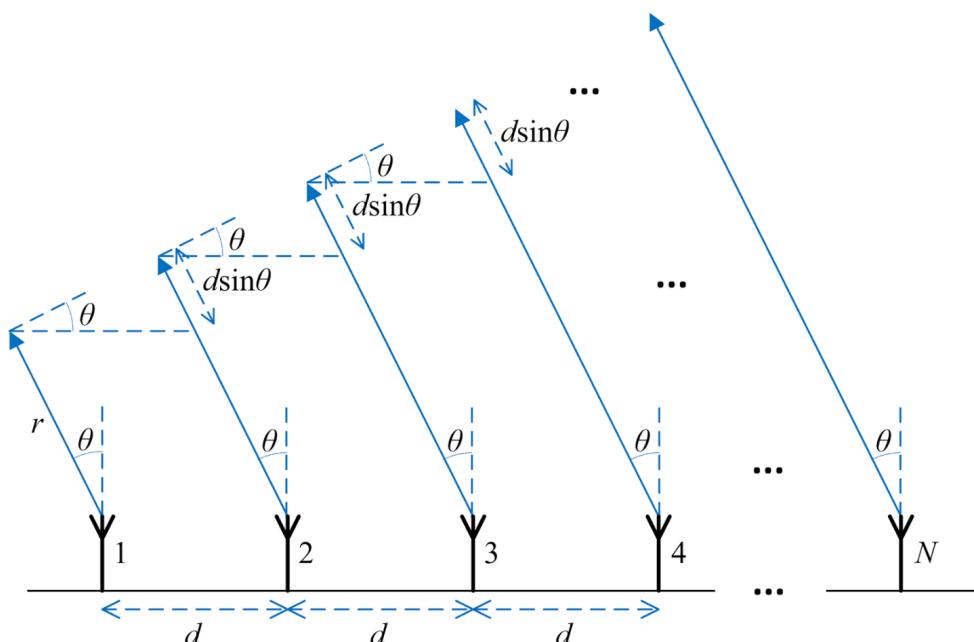


Figura 1 – Relação entre direção de chegada e defasagem de um ULA [1]

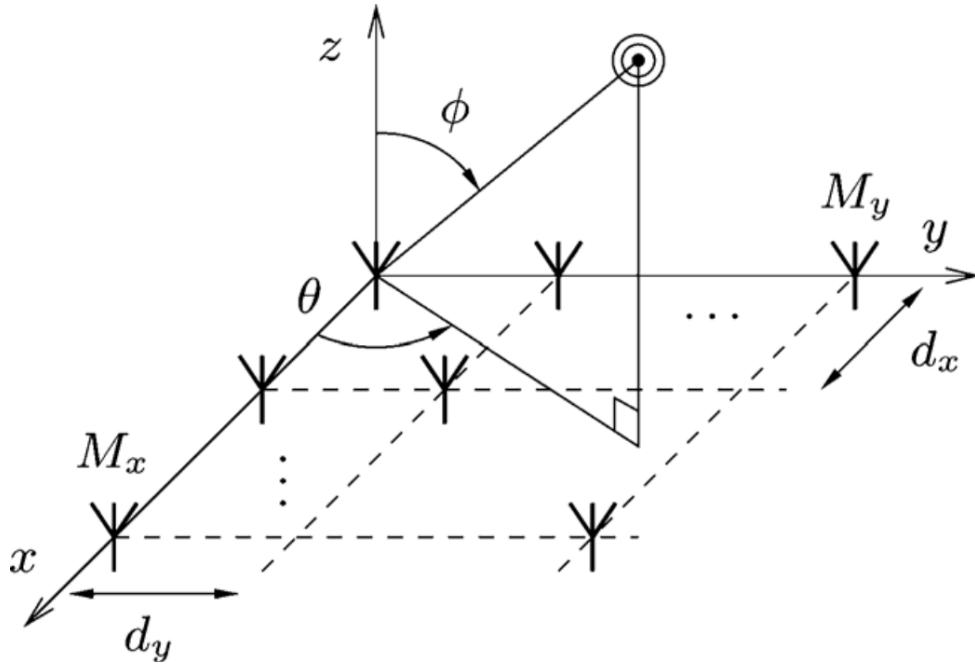


Figura 2 – Relação entre direção de chegada e defasagem de um URA [2]

Agrupando todas as defasagens em um vetor, pode-se obter o *steering vector* (dimensão $N \times 1$) associado ao arranjo de antenas, como visto na Equação (2.1) [1].

$$\mathbf{a}(\theta, \phi) = [1, e^{j\psi_2}, e^{j\psi_3}, \dots, e^{j\psi_N}]^T . \quad (2.1)$$

No caso ULA com mesma distância entre antenas (d), a defasagem é uma função apenas do ângulo θ representado na Figura 1. Neste caso, dado a constante

$$k = \frac{2\pi d}{\lambda} \quad (2.2)$$

a defasagem fica:

$$\psi = k \sin \theta . \quad (2.3)$$

Observe que, neste caso, a defasagem não depende do ângulo de elevação, reduzindo assim a dimensão do problema. A equação do *steering vector* é:

$$\mathbf{a}(\theta) = [1, e^{j\psi}, e^{j2\psi}, \dots, e^{j(N-1)\psi}]^T . \quad (2.4)$$

Para o caso URA a defasagem é função de θ e ϕ , ilustrados na Figura 2. O problema então pode ser dividido em duas equações para as defasagens, uma para cada eixo (x e y). Dado a constante k e distância igual entre elementos ($d = d_x = d_y$), as defasagens ficam:

$$\psi_x = k \cos \theta \sin \phi \quad (2.5)$$

$$\psi_y = k \sin \theta \sin \phi . \quad (2.6)$$

A partir destas equações de diferença de fase pode-se obter dois *steering vectors*:

$$\mathbf{a}_x = [1, e^{j\psi_x}, e^{j2\psi_x}, \dots, e^{j(M_x-1)\psi_x}]^T \quad (2.7)$$

$$\mathbf{a}_y = [1, e^{j\psi_y}, e^{j2\psi_y}, \dots, e^{j(M_y-1)\psi_y}]^T . \quad (2.8)$$

Por fim, o *steering vector* geral do caso URA é dado pelo produto de Kronecker [19] dos dois vetores de cada eixo como:

$$\mathbf{a}(\theta, \phi) = \mathbf{a}_x \otimes \mathbf{a}_y . \quad (2.9)$$

2.2 Métodos de estimação DoA

Esta seção mostra um pouco sobre os dois métodos de estimação deste trabalho: MUSIC e ESPRIT. Ambos os métodos utilizam o *steering vector* visto anteriormente. Além disso, ambos os métodos utilizam a matriz de autocorrelação do vetor do sinal recebido [14]. Se o sinal recebido no primeiro elemento é $s(t)$, então o vetor de sinal em todos os elementos é

$$\mathbf{x} = \mathbf{a}(\theta, \phi) s(t) . \quad (2.10)$$

Com estas premissas, pode ser definida uma matriz de autocorrelação deste vetor de sinal dada por

$$\mathbf{P}_{\mathbf{xx}} = \mathbb{E}\{\mathbf{x} \mathbf{x}^H\} \quad (2.11)$$

onde \mathbf{x}^H é o conjugado transposto de \mathbf{x} . Na prática, no entanto, não conhecemos a matriz $\mathbf{P}_{\mathbf{xx}}$, já que isso envolveria conhecer previamente sua direção de chegada.

Para obter o sinal, é necessário realizar amostragens em cada antena. Supondo que foram capturadas L amostras do sinal em cada antenas, pode então ser definido uma matriz \mathbf{X} (dimensão $N \times L$) na qual em uma linha estão as amostras da uma antena. A matriz de autocorrelação pode ser estimada por:

$$\mathbf{P}_{\mathbf{xx}} \approx \mathbf{R}_{\mathbf{xx}} = \frac{1}{L} (\mathbf{X} \mathbf{X}^H) \quad (2.12)$$

onde \mathbf{X}^H é o conjugado transposto de \mathbf{X} . Por fim, esta matriz pode ser decomposta em autovalores. Seja \mathbf{D} uma matriz diagonal de dimensão $N \times N$ cujos elementos na diagonal são os autovalores da matriz $\mathbf{R}_{\mathbf{xx}}$, \mathbf{E} uma matriz de dimensão $N \times N$ cujas colunas são os autovetores de $\mathbf{R}_{\mathbf{xx}}$, então

$$\mathbf{R}_{\mathbf{xx}} = \mathbf{E} \mathbf{D} \mathbf{E}^{-1} \quad (2.13)$$

onde \mathbf{E}^{-1} é a inversa da matriz \mathbf{E} . Uma interpretação que pode ser tirada dessa decomposição é que há $N - 1$ autovetores associados a ruído e apenas 1 associado ao sinal (para quando é recebido apenas 1 sinal por vez). O autovetor associado ao sinal é o autovetor que

tem o maior autovalor. Ordenando a matriz \mathbf{E} do autovetor associado ao menor autovalor para o associado ao maior autovalor pode-se separar \mathbf{E} em uma matriz \mathbf{E}_N ($N \times (N - 1)$) associada ao ruído e um vetor \mathbf{e}_S ($N \times 1$) associado ao sinal [14].

2.2.1 MUSIC

MUSIC é uma técnica com alta resolução e alta precisão [14]. É possível utilizar a ortogonalidade dos autovetores associados ao ruído com o *steering vector* para detectar a direção de chegada do sinal recebido. A equação

$$R(\theta, \phi) = \frac{1}{|\mathbf{a}(\theta, \phi)^H \mathbf{E}_N \mathbf{E}_N^H \mathbf{a}(\theta, \phi)|} \quad (2.14)$$

deve ser máxima quando o par de ângulos azimute-elevação for igual a direção de chegada real do sinal. A função $R(\theta, \phi)$ pode ser chamada de pseudo espectro do MUSIC.

2.2.2 ESPRIT

O algoritmo ESPRIT se baseia em obter de forma inteligente dois subarranjos de antenas a partir do arranjo original que obedecam a igualdade

$$\mathbf{e}_{S2} = e^{j\psi} \mathbf{e}_{S1} \quad (2.15)$$

onde \mathbf{e}_{S1} e \mathbf{e}_{S2} são os autovetores associados ao sinal de ambos subarranjos e podem ser obtidos a partir das equações

$$\mathbf{e}_{S1} = \mathbf{J}_{S1} \mathbf{e}_S \quad (2.16)$$

$$\mathbf{e}_{S2} = \mathbf{J}_{S2} \mathbf{e}_S , \quad (2.17)$$

nas quais \mathbf{J}_{S1} e \mathbf{J}_{S2} são chamadas matrizes de seleção e tem o objetivo de selecionar os valores do autovetor associado ao sinal correspondente às antenas dos subarranjos. Então, para obter $e^{j\psi}$ é só utilizar

$$e^{j\psi} = [\mathbf{e}_{S1}^H \mathbf{e}_{S1}]^{-1} \mathbf{e}_{S1}^H \mathbf{e}_{S2} . \quad (2.18)$$

Por fim, ψ pode ser obtido a partir das equações de defasagem entre os elementos do arranjo que dependem da geometria. Para um ULA, por exemplo, ψ está definido na Equação (2.3). Para um URA, no entanto, há dois valores para ψ e podem ser obtidos a partir das Equações (2.5) e (2.6).

Uma parte essencial deste método é a escolha das matrizes de seleção que irão permitir utilizar a Equação (2.15). No caso de um ULA, apenas duas matrizes de seleção são necessárias. No entanto no caso do URA, são necessárias duas matrizes de seleção para

cada ψ , pois a Equação (2.15) tem que ser resolvida tanto para ψ_x quanto para ψ_y . O arranjo URA-16 (arranjo uniforme retangular com 16 antenas) será usado como exemplo.

O *steering vector* para o URA-16 é

$$\mathbf{a} = \begin{bmatrix} 1 \\ e^{j\psi_x} \\ e^{j2\psi_x} \\ e^{j3\psi_x} \\ e^{j\psi_y} \\ e^{j\psi_y} e^{j\psi_x} \\ e^{j\psi_y} e^{j2\psi_x} \\ e^{j\psi_y} e^{j3\psi_x} \\ e^{j2\psi_y} \\ e^{j2\psi_y} e^{j\psi_x} \\ e^{j2\psi_y} e^{j2\psi_x} \\ e^{j2\psi_y} e^{j3\psi_x} \\ e^{j3\psi_y} \\ e^{j3\psi_y} e^{j\psi_x} \\ e^{j3\psi_y} e^{j2\psi_x} \\ e^{j3\psi_y} e^{j3\psi_x} \end{bmatrix}. \quad (2.19)$$

Para encontrar ψ_y é necessário então escolher matrizes de seleção que permitam chegar à

$$\mathbf{a}_{2y} = e^{j\psi_y} \mathbf{a}_{1y} \quad (2.20)$$

e isso é feito escolhendo os subarranjos contendo os doze primeiros elementos como \mathbf{a}_{1y} e os doze últimos elementos como \mathbf{a}_{2y} . Os *steering vectors* de cada subarranjo então são:

$$\mathbf{a}_{1y} = \begin{bmatrix} 1 \\ e^{j\psi_x} \\ e^{j2\psi_x} \\ e^{j3\psi_x} \\ e^{j\psi_y} \\ e^{j\psi_y} e^{j\psi_x} \\ e^{j\psi_y} e^{j2\psi_x} \\ e^{j\psi_y} e^{j3\psi_x} \\ e^{j2\psi_y} \\ e^{j2\psi_y} e^{j\psi_x} \\ e^{j2\psi_y} e^{j2\psi_x} \\ e^{j2\psi_y} e^{j3\psi_x} \\ e^{j3\psi_y} \\ e^{j3\psi_y} e^{j\psi_x} \\ e^{j3\psi_y} e^{j2\psi_x} \\ e^{j3\psi_y} e^{j3\psi_x} \end{bmatrix}, \quad \mathbf{a}_{2y} = \begin{bmatrix} e^{j\psi_y} \\ e^{j\psi_y} e^{j\psi_x} \\ e^{j\psi_y} e^{j2\psi_x} \\ e^{j\psi_y} e^{j3\psi_x} \\ e^{j2\psi_y} \\ e^{j2\psi_y} e^{j\psi_x} \\ e^{j2\psi_y} e^{j2\psi_x} \\ e^{j2\psi_y} e^{j3\psi_x} \\ e^{j3\psi_y} \\ e^{j3\psi_y} e^{j\psi_x} \\ e^{j3\psi_y} e^{j2\psi_x} \\ e^{j3\psi_y} e^{j3\psi_x} \end{bmatrix}$$

e as matrizes de seleção $\mathbf{J}_{\mathbf{S1y}}$ e $\mathbf{J}_{\mathbf{S2y}}$ são:

$$\mathbf{J}_{\mathbf{S1y}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{J}_{\mathbf{S2y}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} .$$

Com as matrizes de seleção, as Equações (2.16) e (2.17) podem ser utilizadas para obter os subarranjos e (2.15) para obter ψ_y . É necessário ainda obter matrizes de seleção para conseguir obter ψ_x . Assim como anteriormente, a escolha tem que obedecer a igualdade

$$\mathbf{a}_{2x} = e^{j\psi_x} \mathbf{a}_{1x} . \quad (2.21)$$

Para isso, foram escolhidos os subarranjos que formam os seguintes *steering vectors*:

$$\mathbf{a}_{1x} = \begin{bmatrix} 1 \\ e^{j\psi_x} \\ e^{j2\psi_x} \\ e^{j\psi_y} \\ e^{j\psi_y} e^{j\psi_x} \\ e^{j\psi_y} e^{j2\psi_x} \\ e^{j2\psi_y} \\ e^{j2\psi_y} e^{j2\psi_x} \\ e^{j3\psi_y} \\ e^{j3\psi_y} e^{j\psi_x} \\ e^{j3\psi_y} e^{j2\psi_x} \end{bmatrix}, \quad \mathbf{a}_{2x} = \begin{bmatrix} e^{j\psi_x} \\ e^{j2\psi_x} \\ e^{j3\psi_x} \\ e^{j\psi_y} e^{j\psi_x} \\ e^{j\psi_y} e^{j2\psi_x} \\ e^{j\psi_y} e^{j3\psi_x} \\ e^{j2\psi_y} e^{j\psi_x} \\ e^{j2\psi_y} e^{j2\psi_x} \\ e^{j2\psi_y} e^{j3\psi_x} \\ e^{j3\psi_y} e^{j\psi_x} \\ e^{j3\psi_y} e^{j2\psi_x} \\ e^{j3\psi_y} e^{j3\psi_x} \end{bmatrix}.$$

As matrizes de seleção $\mathbf{J}_{\mathbf{S1x}}$ e $\mathbf{J}_{\mathbf{S2x}}$ neste caso ficam:

Utilizando novamente as Equações (2.16), (2.17) e (2.15) pode-se obter ψ_x . Com ψ_x e ψ_y é possível usar as Equações (2.5) e (2.6) para obter os ângulos azimute (θ) e elevação (ϕ) da seguinte maneira:

$$\theta = \arctan\left(\frac{\psi_y}{\psi_x}\right) \quad (2.22)$$

$$\phi = \arcsin\left(\frac{\psi_y}{k \sin \theta}\right). \quad (2.23)$$

2.3 O método do Gradiente

O método do Gradiente é um algoritmo de primeira ordem que otimiza o problema de localizar o mínimo ou o máximo de uma função diferenciável [20]. Ele se baseia em dar passos onde o gradiente da função está apontando, pois, localmente, este é o ponto onde a curva está crescendo ou diminuindo com maior intensidade. Dependendo do sentido para onde o passo é dado define-se se a otimização é de ascensão (encontrar o máximo da função) ou de declive (encontrar o mínimo). As equações do método Gradiente em sua forma mais simples para encontrar o máximo da função (ascendente) são

$$\begin{aligned} \mathbf{v}^{(k+1)} &= \alpha \mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mathbf{v}^{(k+1)}. \end{aligned} \quad (2.24)$$

Nestas equações \mathbf{x} é o vetor dos pontos da função, $\mathbf{g} = \nabla f(\mathbf{x}^{(k)})$ é o gradiente e α é o *learning rate* do método (geralmente um número entre 0 e 1). Escolher o *learning rate* é uma tarefa difícil, pois a escolha de um *learning rate* alto pode fazer o método não convergir e um número muito baixo pode fazer com que a convergência demore muito. Além disso, a convergência para o máximo ou mínimo global só é garantida caso não haja outros mínimos ou máximos locais. Geralmente este método tem as seguintes condições de parada:

- Número máximo de iterações: útil para os casos em que o método diverge.
- Progresso mínimo: útil para quando o método começa a avançar muito devagar e não chegaria no seu objetivo em um tempo hábil.
- Magnitude do gradiente: condição principal do método. Quando $|\mathbf{g}| < \varepsilon$, o gradiente está suficientemente próximo de zero e o objetivo da função foi encontrado.

2.3.1 Momentum

O método do Gradiente tem dificuldades com subidas lentas. Para superar este problema, o *Momentum* foi proposto [20]. A ideia é acumular uma parte dos passos

anteriores no passo atual, permitindo acelerar a convergência. As equações do *Momentum* são:

$$\begin{aligned}\mathbf{v}^{(k+1)} &= \beta\mathbf{v}^{(k)} + \alpha\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mathbf{v}^{(k+1)}\end{aligned}\tag{2.25}$$

onde β é um coeficiente de 0 a 1 que amortece o passo anterior.

2.3.2 Nesterov Momentum

O algoritmo *Nesterov momentum* tenta olhar para o futuro e prever como será o gradiente do próximo passo, para tomar melhores decisões sobre o passo atual [20]. Para realizar isso com baixo custo computacional, o método faz o gradiente de \mathbf{x} mais o valor do *momentum* o que aproxima bem o que seria o passo seguinte. As equações que definem este método são:

$$\begin{aligned}\mathbf{v}^{(k+1)} &= \beta\mathbf{v}^{(k)} + \alpha\nabla f(\mathbf{x}^{(k)} + \beta\mathbf{v}^{(k)}) \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mathbf{v}^{(k+1)}.\end{aligned}\tag{2.26}$$

3 MATERIAIS E MÉTODOS

Este capítulo apresenta as ferramentas e materiais utilizados além dos detalhes de implementação dos algoritmos.

3.1 Materiais

3.1.1 O protocolo Bluetooth 5.1

Bluetooth 5.1 é um protocolo de comunicação com alta eficiência energética [11] e pronto para a utilização em detecção de direção de chegada [3]. Há 40 canais com espaçamento de 2 MHz cada e frequência de transmissão que vai de 2,4 GHz a 2,48 GHz [21]. Neste protocolo há um pacote específico para estimativa de DoA, ilustrado na Figura 3. A parte que será utilizada para estimar a DoA é o CTE (*Constant Tone Extension*), um sinal senoidal com frequência 250 kHz em banda base. Este sinal, após ser recebido pelo arranjo de antenas, é amostrado em cada antena para calcular a defasagem em cada antena e assim, conseguir fazer a estimativa de DoA.

3.1.2 Transmissor

Utilizou-se como transmissor o kit de desenvolvimento **EFR32BG22 Thunderboard Kit** da Silabs [4] (Figura 4). Este kit é pequeno (45 mm x 30 mm) e equipado com um SoC (*System on a Chip*) Bluetooth. Ele deve ser carregado junto ao ativo a ser rastreado e a partir dele, é que será identificado a direção. A taxa de transmissão utilizada foi de 1 envio de sinal a cada 100 ms (10 Hz).

3.1.3 Receptor

A placa principal do receptor utilizada foi **EFR32xG22 Wireless Starter Kit** (WSTK) [5] (Figura 5) em conjunto com o arranjo de 16 antenas **BRD4185A** [6] (Figura 6) ambos da Silabs. O WSTK foi responsável por controlar as antenas e realizar a amostragem do sinal. O arranjo de antenas é um URA com 16 antenas com mesma distância de 40mm



Figura 3 – Pacote do protocolo Bluetooth 5.1 para estimativa de DoA [3]



Figura 4 – Transmissor EFR32BG22 Thunderboard Kit [4].

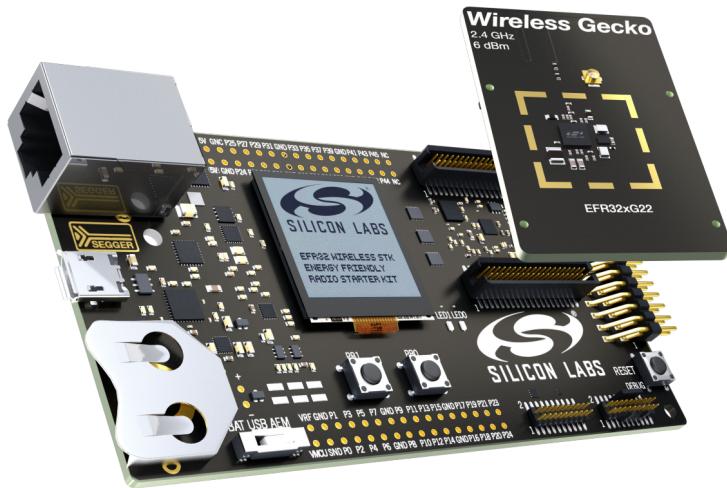


Figura 5 – Módulo de recepção WSTK [5].

entre cada antena nos dois eixos. Um aplicativo rodando em um kit de desenvolvimento **Raspberry Pi 4 Model B** foi responsável por processar as amostras e realizar as estimativas das direções de chegada.

3.1.4 Módulo de processamento

Para processar as amostras, utilizou-se o kit **Raspberry Pi 4 Model B** (Figura 7). Este kit conta com um processador **Quad core 64-bit ARM-Cortex A72** com frequência de 1,5 GHz e 4 GB de memória RAM [22]. Este processador possui quatro núcleos e suporte para SIMD (*Advanced Single Instruction Multiple Data*) [23]. Esse dado tem relevância pois, muitas operações com matriz e vetores podem ser paralelizadas utilizando instruções vetoriais e múltiplos núcleos. A escolha desse kit se deu, além do baixo custo, por ser um processador versátil e amplamente utilizado.

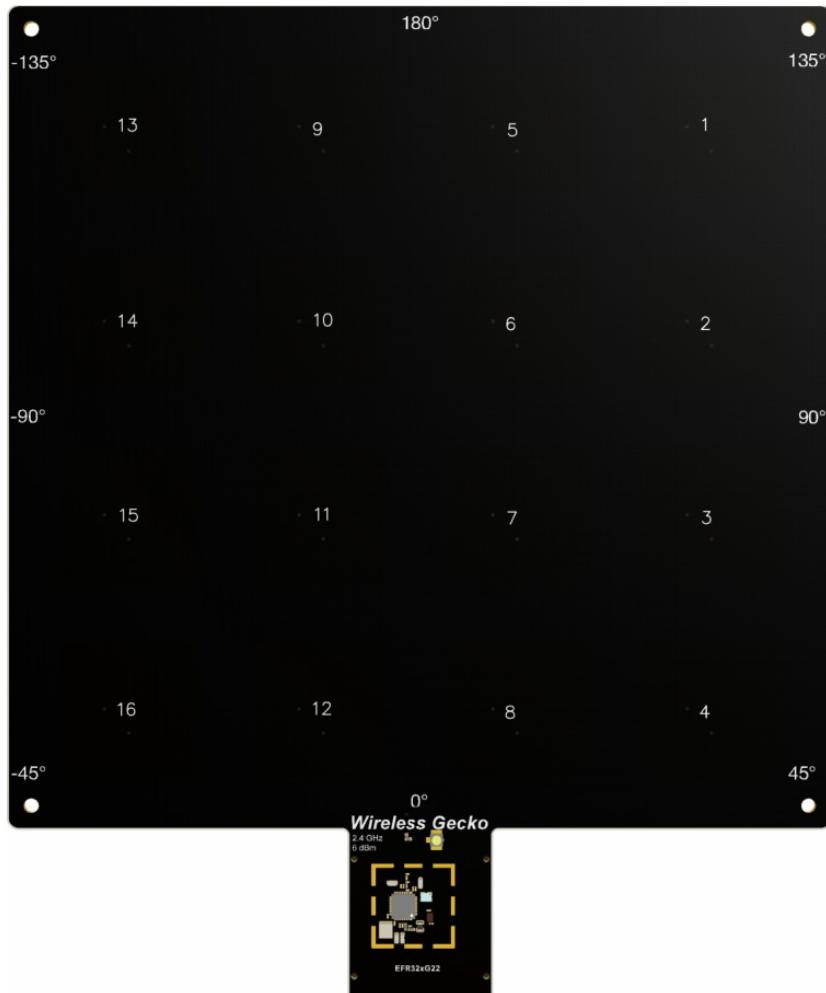


Figura 6 – Arranjo de antenas BRD4185A [6].



Figura 7 – Módulo de processamento Raspberry Pi 4 Model B [7].

3.1.5 Linguagens de programação

A linguagem de programação escolhida para o processamento de amostras e estimação de DoA foi C++ devido ao seu bom equilíbrio entre abstração e velocidade de processamento. Dentro da linguagem C++ foi escolhido a biblioteca de álgebra linear **Eigen** devido a facilidade de uso e boa relação entre abstração e performance [24]. Além da implementação dos algoritmos de DoA, também foi utilizado C++ para realizar os testes e salvar os resultados.

Para analisar os resultados estatísticos dos testes e também para gerar as imagens do pseudo espectro do MUSIC utilizou-se a linguagem Python em conjunto com as bibliotecas Numpy, Pandas e Matplotlib.

3.2 Amostragem

Para realizar o processo de amostragem recorreu-se à um aplicativo desenvolvido em [1]. Este aplicativo utiliza a API (*Application Programming Interface*) desenvolvida pela Silabs e faz amostras em tempo real. O código foi alterado para salvar as amostras em um arquivo. Como visto anteriormente, o transmissor emitia um sinal a cada 100 ms (10 sinais por segundo). Além disso, utilizou-se uma opção da API da Silabs que elimina amostras ruins baseado nos dados IQ (*In-phase and Quadrature*) das amostras.

Para ter um número relevante de amostras para os testes foram utilizados conjuntos com 10 mil amostras cada. Baseando-se nos objetivos vistos no Capítulo 1, foram escolhidas dois tipos de amostras: pontos fixos e andando em um ambiente de escritório.

As amostras com o transmissor em pontos fixos serão usadas para poder medir o desvio padrão dos métodos, porém sem ter os ângulos exatos naquele ponto pois, seria inviável medir ângulos precisos com as ferramentas disponíveis. Como o *beacon* está em um ponto fixo, o desvio padrão das medidas deve ser o menor possível, demonstrando a precisão da técnica. Além de comparar o MUSIC com o ESPRIT, também será comparado com o método de estimativa da API da Silabs. Foram capturados três conjuntos de amostras em pontos fixos.

Para analisar a acurácia e tempo de execução dos algoritmos implementados, foram feitas amostras durante um uso comum em escritório. Estas amostras foram obtidas carregando o transmissor pelo escritório e dando voltas nas várias baías chegando a distâncias de até 10 m.

3.3 Implementação dos métodos DoA

A implementação dos métodos DoA em C++ foi feita utilizando as equações vistas no Capítulo 2. Primeiramente implementou-se o método ESPRIT, com o seu funcionamento verificado em tempo real pelo aplicativo mencionado acima. Este aplicativo, além de realizar a amostragem, fazia a estimativa da direção de chegada tanto pelo método ESPRIT implementado neste trabalho quanto pela API da Silabs. Ambos os valores eram coerentes com a posição do *beacon*.

Posteriormente, o método MUSIC foi implementado com busca exaustiva. Devido ao alto gasto computacional, este método não pode ser validado em tempo real, mostrando a necessidade das otimizações desenvolvidas neste trabalho.

3.4 Análise do pseudo espectro do MUSIC

Após a implementação das técnicas DoA, analisou-se o pseudo espectro do MUSIC para averiguar quais aprimoramentos seriam interessantes de serem aplicados. Para fazer essas análises, contou-se com a linguagem Python para gerar as imagens do pseudo espectro com os resultados produzidos pela técnica MUSIC implementada em C++.

Conforme visto em [3], o pseudo espectro do MUSIC esperado é parecido com o que foi obtido nesta amostra das Figuras 8 e 9. Normalmente, em um sistema de coordenadas de três dimensões, dois ângulos definem uma direção: azimute (0° a 360°) e elevação (0° a 180°). No entanto, no caso da estimativa de DoA com um URA, o ângulo de elevação acima e abaixo da antena não consegue ser diferenciado, assim a elevação vai de 0° a 90° . Isso não é um problema, pois neste tipo de aplicação as antenas geralmente são colocadas no teto do edifício, assim apenas o ângulo de 0° a 90° é relevante.

Deveríamos então, ver essa repetição do pseudo espectro do MUSIC no caso de utilizarmos ângulos de elevação superiores a 90° . Apenas para servir de ilustração, foi gerado a imagem do pseudo espectro 3D do MUSIC para elevação de 0° a 360° da mesma amostra mostrada na Figura 8. Esse pseudo espectro completo está ilustrado na Figura 10.

Observe que para elevação de 90° a 180° é como se o pseudo espectro estivesse espelhado. Já para elevação de 180° a 360° acontece algo interessante: o pseudo espectro é igual ao de 0° a 180° porém com o azimute somado de 180° . Como no nosso algoritmo do Gradiente o resultado pode estar fora da extensão comum (elevação pode dar menor que 0° ou maior que 90°), essas observações são relevantes para poder trazer o resultado para valores de azimute e elevação reais. Daqui em diante, no entanto, todas as figuras do pseudo espectro serão com elevação de 0° a 90° .

Como é possível observar neste pseudo espectro comum, a curva produzida pelo resultado do MUSIC é uma curva suave e com um máximo global claro. Notando essas

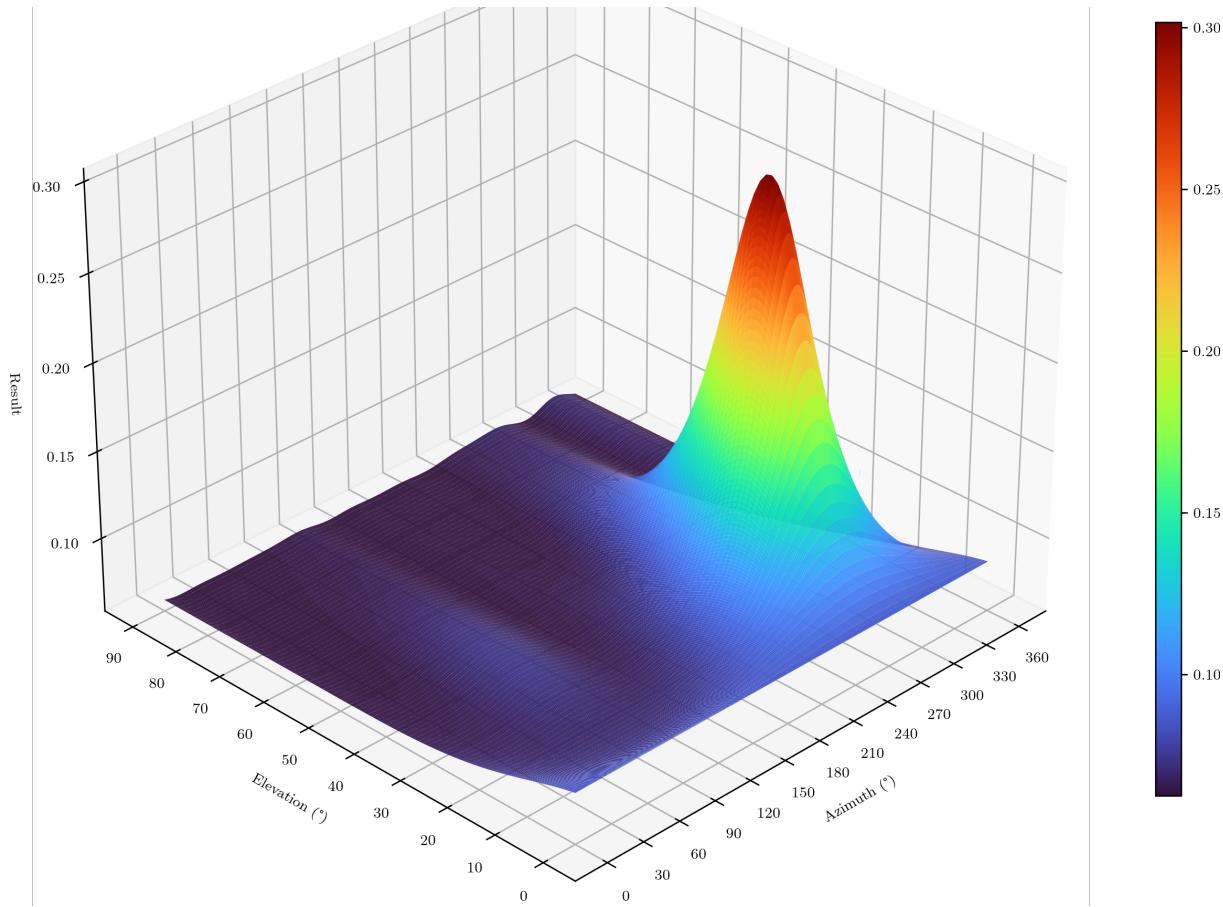


Figura 8 – Pseudo espectro em 3D do MUSIC de uma amostra comum

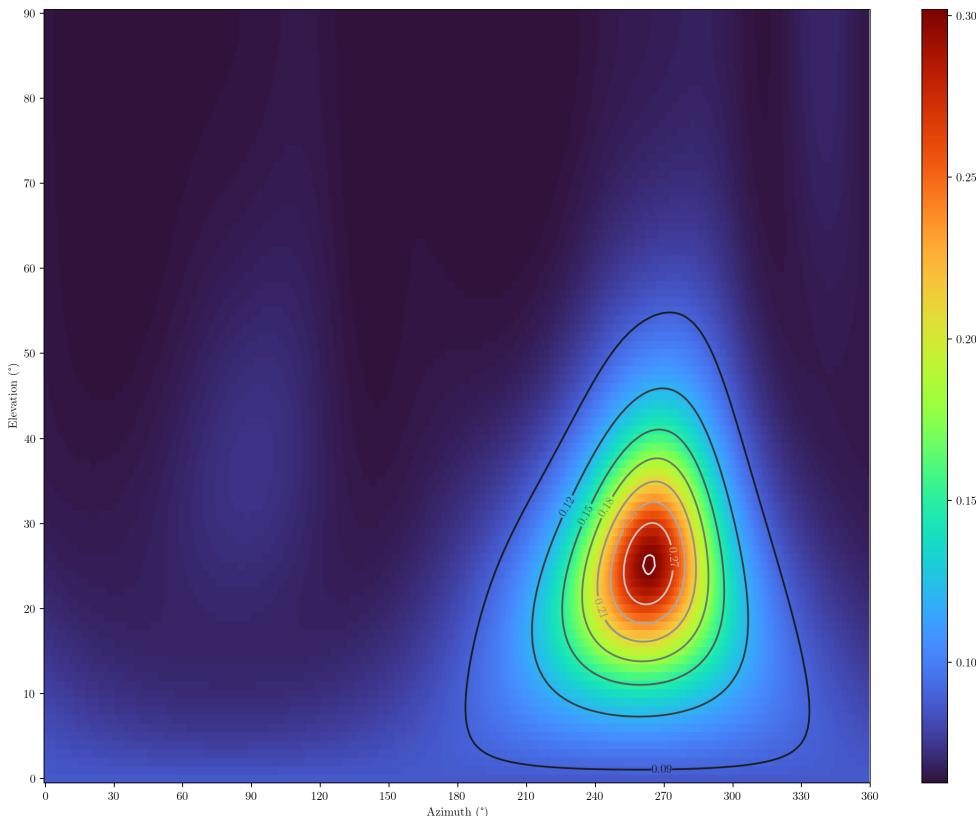


Figura 9 – Curvas de nível do pseudo espectro do MUSIC de uma amostra comum

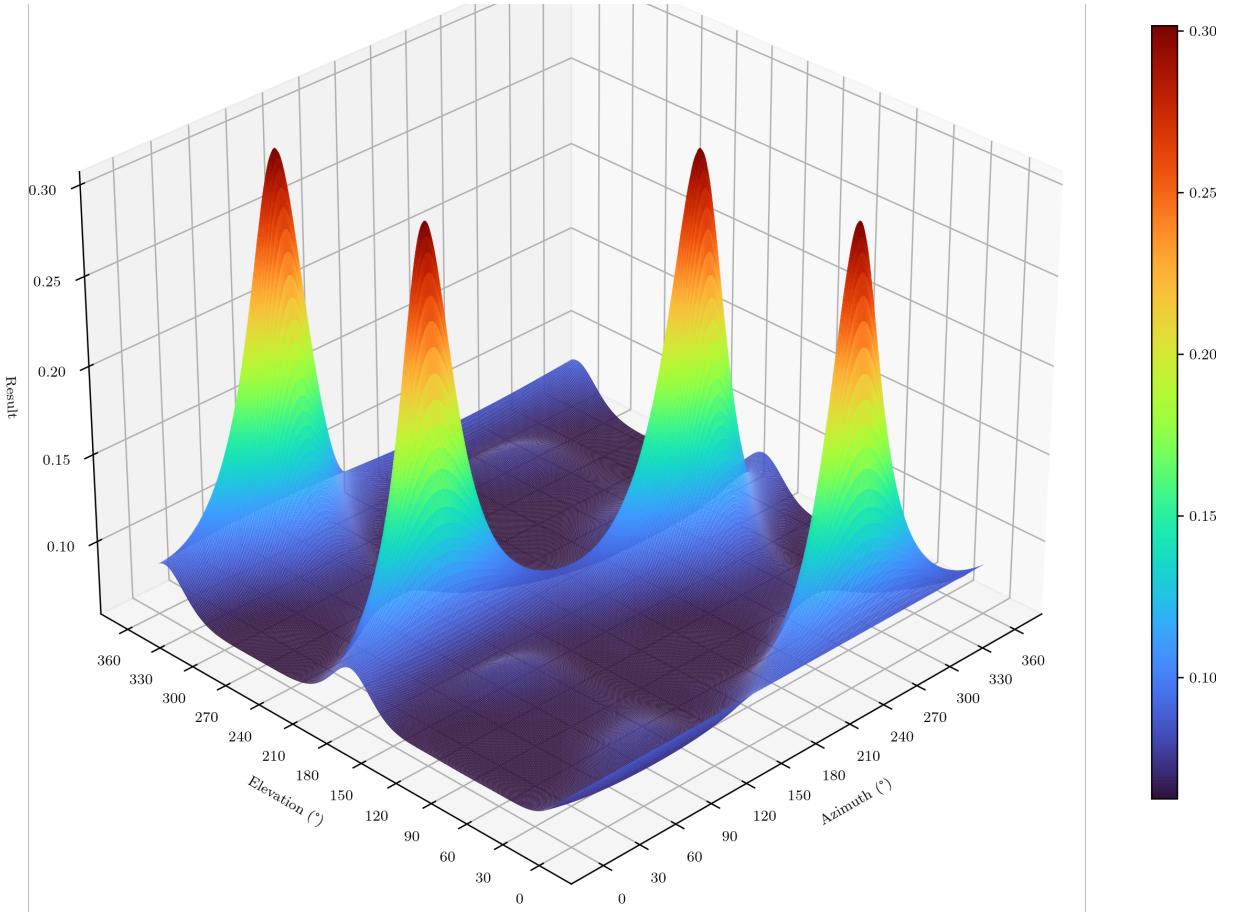


Figura 10 – Pseudo espectro em 3D do MUSIC de uma amostra comum para elevação de 0° a 360°

características, ponderou-se utilizar o método de otimização Gradiente. Porém além do máximo global, há máximos locais também como fica evidente nas Figuras 11 e 12, com isso o algoritmo pode convergir em um máximo local. Todavia, se for possível chegar na rampa que vai para o máximo global o Gradiente provavelmente convergiria para o ponto desejado.

Com isso em mente, surgiu a ideia de realizar uma busca grossa com uma resolução menor e utilizar o maior valor dessa busca como ponto inicial do Gradiente. Neste caso, a chance de começar na rampa que vai para o máximo global, aumenta. Este tipo de busca em dois estágios pode ser visto em [13].

No entanto, por volta de 1% das amostras coletadas apresentaram características que dificultam a detecção de um claro máximo global. Como pode ser observado nas Figuras 13 e 14, os resultados do MUSIC em máximos locais são muito próximos do global. Inclusive, há um *range* de elevação no qual o azimute fica praticamente indefinido, podendo assumir qualquer valor de 0° a 360° . Como as amostras foram tomadas andando pelo escritório, não dá para confirmar qual foi o fenômeno que causou esse problema.

Nestas amostras o método do Gradiente tem bastante dificuldade para atravessar

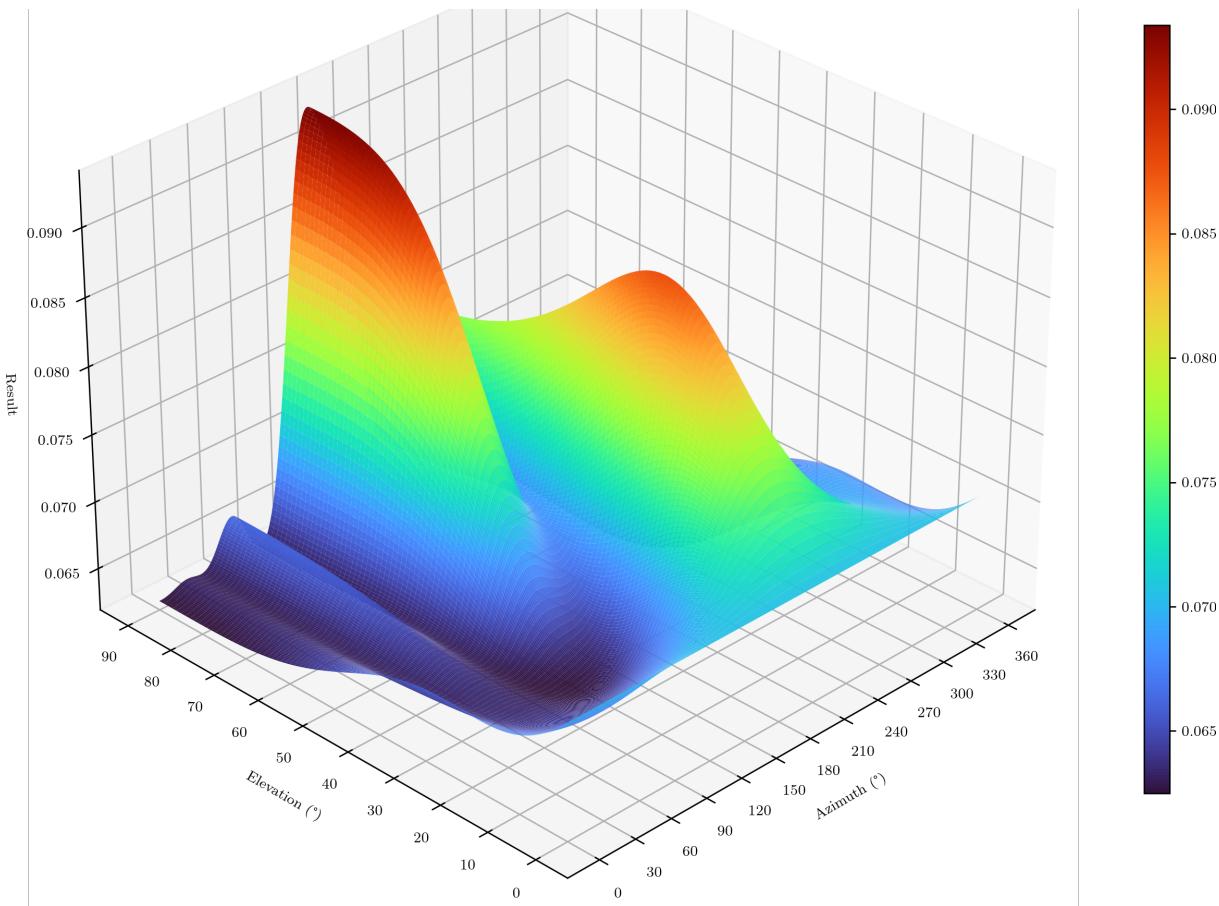


Figura 11 – Pseudo espectro em 3D do MUSIC de uma amostra com máximos locais claros

essas pequenas inclinações. Uma possível solução para o Gradiente convergir mais rápido, é utilizar o método de otimização *Momentum*. Uma outra solução para o problema como um todo, é, na etapa de busca grossa, tentar identificar estas amostras e excluí-las, pois o resultado é dúvida demais para se tirar alguma conclusão. Isso poderia ser feito observando se há uma diferença relevante entre o máximo global a média de todos os resultados percorridos, por exemplo.

3.5 Implementação dos aprimoramentos do MUSIC

3.5.1 Busca em dois estágios

Como visto na Seção 3.4, surgiu a ideia de otimizar o MUSIC fazendo uma busca em duas etapas: uma grossa (menor resolução) e uma fina (maior resolução). A busca fina pode ser uma busca simples em torno do ponto máximo da busca grossa ou com algum outro método de otimização como o Gradiente.

Para o caso da busca grossa seguida de uma fina (grossa-fina), pode-se supor que o máximo global esteja até um passo grosso (p_g) de distância do máximo encontrado na busca grossa. Se isso for verdade, então é necessário realizar a busca fina apenas nesta região de

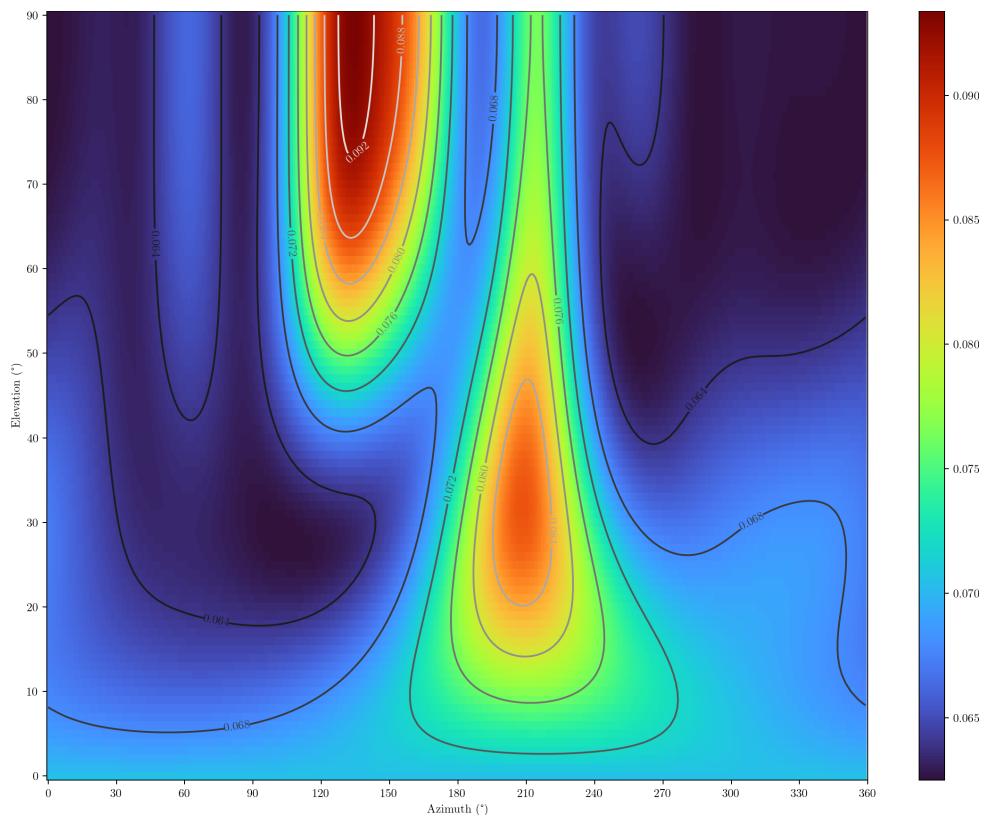


Figura 12 – Curvas de nível do pseudo espectro do MUSIC de uma amostra com máximos locais claros

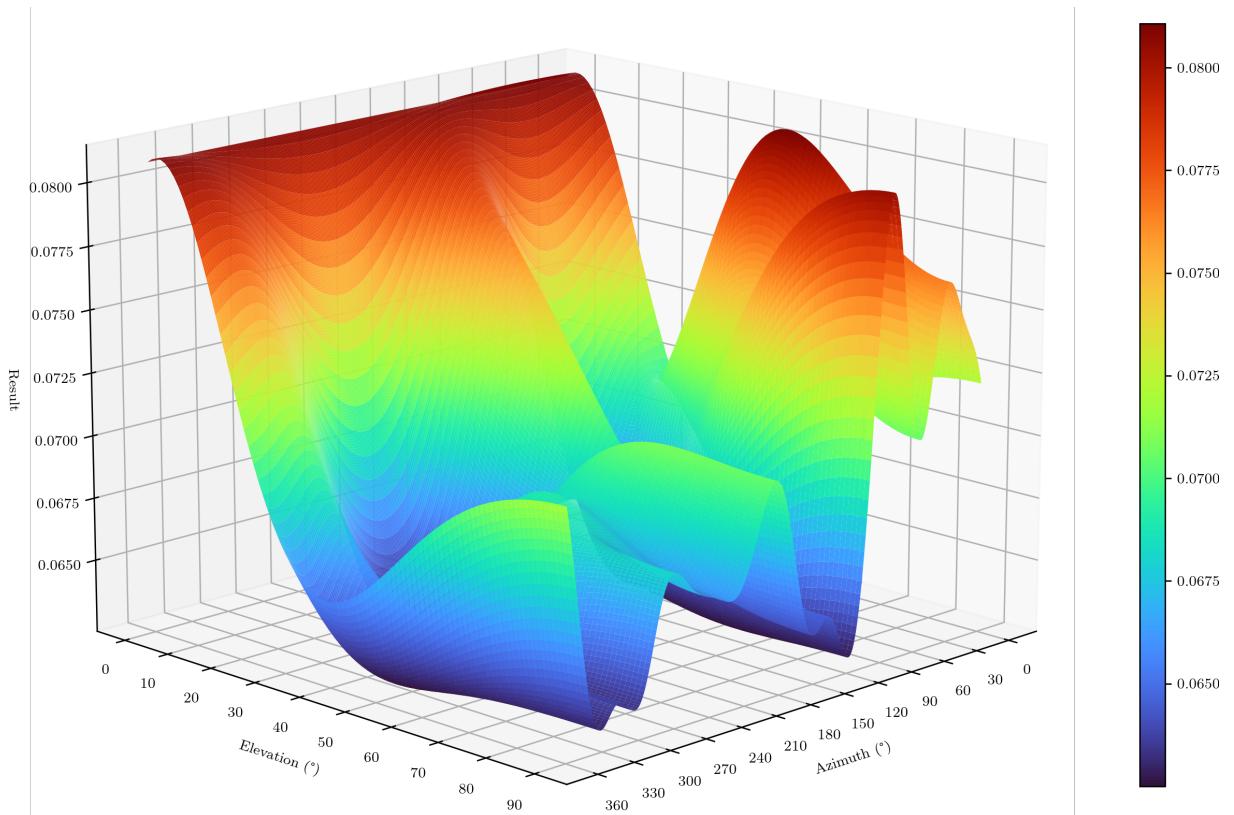


Figura 13 – Pseudo espectro em 3D do MUSIC de uma amostra deteriorada

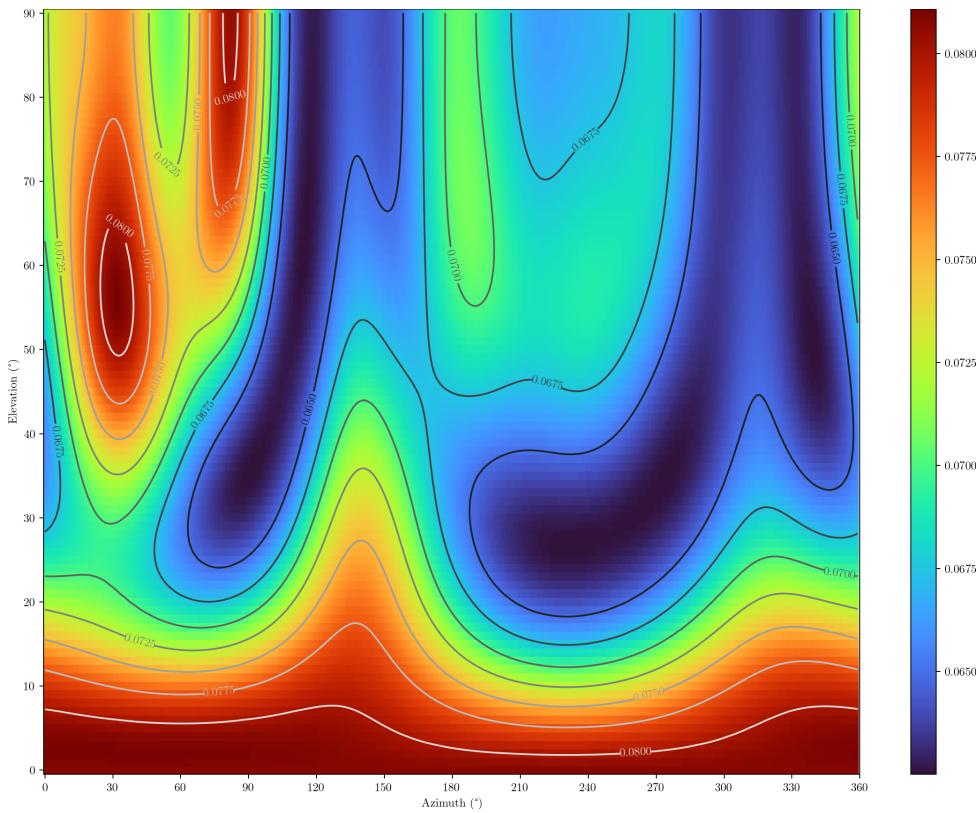


Figura 14 – Curvas de nível do pseudo espectro do MUSIC de uma amostra deteriorada

um passo grosso para cada lado do máximo achado na busca grossa. Este suposição será verificada nos experimentos. Caso ela seja verdade, o erro médio com relação ao MUSIC de busca exaustiva de alta resolução será parecido com o tamanho do passo fino (p_f) dado na busca grossa-fina. Esse método está melhor exemplificado na Figura 15.

Para obter o passo grosso que minimiza o número de iterações, dado um passo fino desejado, pode-se utilizar:

$$i = \min \left(\frac{(\frac{360}{p_g})^2}{4} + \left(\frac{2p_g}{p_f} \right)^2 \right) \quad (3.1)$$

onde i é o número de iterações, p_g é o tamanho do passo grosso e p_f é o tamanho do passo fino. Para achar o mínimo, podemos derivar com relação a p_g e igualar a 0 obtendo

$$p_g = \left(\frac{(360 p_f)^2}{16} \right)^{\frac{1}{4}}. \quad (3.2)$$

Assim para um passo fino de $0,1^\circ$ o passo grosso que otimiza o número de iterações é aproximadamente 3° para este tipo de busca grossa-fina.

3.5.2 Gradiente simples

Utilizando a mesma ideia da busca em dois estágios, pode-se realizar a etapa de busca grossa e utilizar o valor do resultado como ponto inicial para o método do Gradiente

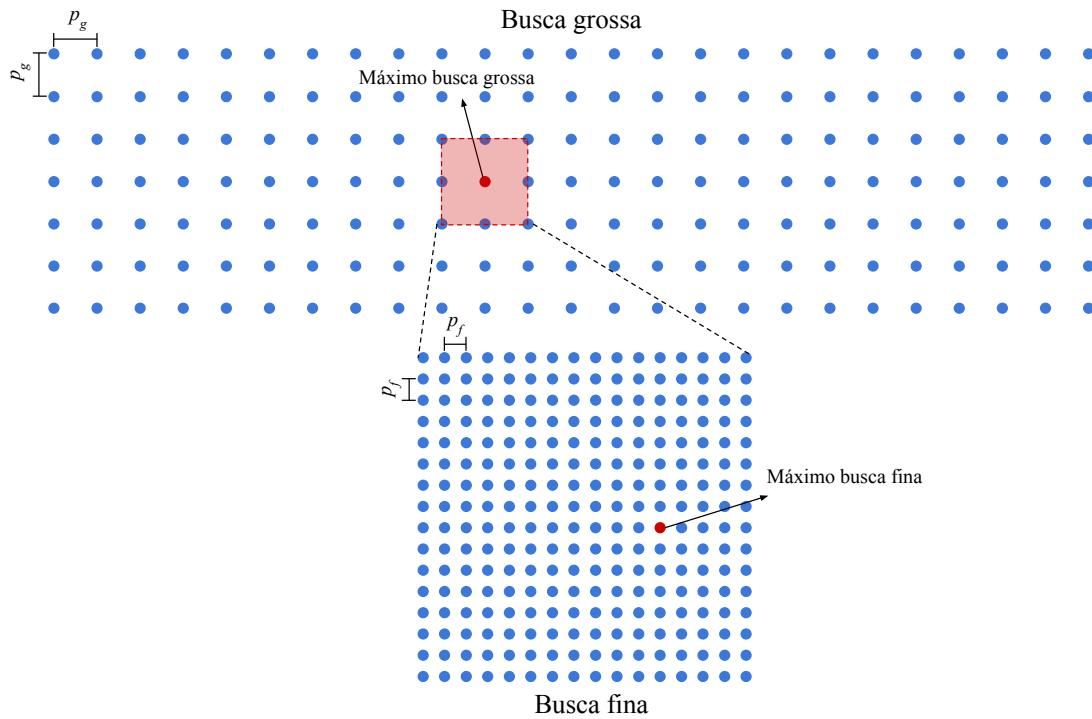


Figura 15 – Exemplo de busca grossa-fina para $p_g = 15^\circ$ e $p_f = 1^\circ$

descrito na Seção 2.3. Primeiramente, pensou-se em utilizar um *learning rate* com valor estático. Isso se demonstrou problemático pois, para atingir o pico com a precisão desejada, o *learning rate* tinha que ser muito pequeno o que fazia com que a convergência demorasse muito.

Para corrigir esses problemas, pensou-se então em utilizar um *learning rate* adaptável. Porém, como os métodos convencionais para se calcular o *learning rate* ideal possuem um alto custo operacional [20], um método próprio foi elaborado para alterar esse valor durante a execução.

Ao cruzar o máximo global, o valor do gradiente do azimute ou da elevação muda o sentido, assim, quando o sinal do gradiente inverte, significa que estamos mais próximos do máximo e podemos diminuir o *learning rate*. Então, implementou-se um algoritmo no qual, toda vez que o gradiente troca de sinal, o *learning rate* é dividido por dois. Esta técnica está demonstrada no Algoritmo 1.

Este aprimoramento pode ter problemas em casos que o gradiente do azimute ou elevação trocam de sinal, porém ainda não chegaram próximo ao máximo. Os experimentos mostrarão se isso ocorre com frequência. Se o tempo de execução ficar maior do que o do Gradiente simples, quer dizer que essa troca de sinal prematura está ocorrendo com frequência. A vantagem dessa técnica é que, como o *learning rate* vai decrescendo, podemos começar com um valor maior e chegar ao valor máximo pela primeira vez mais rapidamente.

Em todos os métodos baseados no Gradiente são feitas derivadas parciais para

Algoritmo 1: Método Gradiente com *learning rate* adaptável utilizado

```

enquanto  $|g^{(k)}| > \varepsilon$  faça
     $g^{(k)} \leftarrow \nabla f(x^{(k)})$ 
    se  $\text{sinal}(g^{(k)}) \neq \text{sinal}(g^{(k-1)})$  então
         $\alpha \leftarrow \frac{1}{2} \alpha$ 
    fim
     $x^{(k+1)} \leftarrow x^{(k)} + \alpha g^{(k)}$ 
     $g^{(k-1)} \leftarrow g^{(k)}$ 
     $k \leftarrow k + 1$ 
fim

```

calcular o gradiente no ponto. Para realizar essas derivadas, utilizou-se o método numérico das diferenças finitas [20] olhando para frente. A derivada pode então ser aproximada por

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad (3.3)$$

Neste método, é necessário a escolha de um tamanho de passo para a diferenciação (h) e, quanto menor, mais próximo da derivada analítica. Para h , empregou-se a aproximação raiz quadrada da precisão do tipo de *float* utilizado. Como neste trabalho utilizou-se *double*, o valor de h é aproximadamente $1,5 \times 10^{-8}$.

3.5.3 Momentum

Além dos métodos do Gradiente vistos anteriormente, implementou-se a técnica *Momentum* como visto na Seção 2.3.1. Para esta técnica, utilizou-se o *learning rate* estático, pois o α adaptável proposto neste trabalho não foi vantajoso. Isso ocorreu porque quando o método atravessava o máximo, ele ainda tinha um *momentum* acumulado e ao reduzir o *learning rate* ele demorava mais para desacelerar e voltar para o pico. Foi também tentado zerar o *momentum* no ponto em que trocava de *learning rate*, porém isso também não gerou melhoria no tempo de execução e o método ficou muito próximo ao desempenho com α estático. Utilizar o α estático implica em começar com valores menores para poder convergir. Isto é contrabalanceado pelo fato de o algoritmo ir ganhando velocidade conforme vai subindo até o topo.

3.5.4 Nesterov Momentum

Foi implementado também o método *Nesterov Momentum* que é um aprimoramento da técnica *Momentum* como descrito na Seção 2.3.2. Utilizou-se *learning rate* estático assim como no *Momentum* pelos mesmos motivos apresentados anteriormente.

4 EXPERIMENTOS E RESULTADOS

4.1 Metodologia

Dados os objetivos vistos na Seção 1.1.2, foram pensados quatro tipos de experimentos: precisão do MUSIC e ESPRIT, escolha dos melhores parâmetros, acurácia entre os métodos MUSIC e tempo de execução de todos os métodos.

4.1.1 Precisão do MUSIC e ESPRIT

Para os testes de precisão foram coletadas amostras em pontos fixos como discutido na Seção 3.2. Então, utilizou-se o MUSIC com busca exaustiva de resolução de $0,05^\circ$ para obter a direção de chegada de cada amostra. Este valor foi escolhido pois o propósito deste teste é comparar a precisão da melhor versão do ESPRIT com a melhor versão do MUSIC. No entanto, se a resolução fosse muito alta, ficaria inviável computar os resultados com a busca exaustiva para todas as amostras. Após a execução do MUSIC, foi executado o método ESPRIT para cada amostra e obtido também as direções de chegada.

Para fazer a análise do desvio padrão, como há dois ângulos no resultado, foi utilizada a distância da origem $(0,0)$ até o ponto definido pelo azimute e elevação da direção (θ, ϕ) . Também analisou-se o desvio padrão (σ) para azimute e elevação separadamente. Além do desvio padrão, foi calculado o MAD (*Mean Absolute Deviation*), para se ter uma ideia da contribuição de *outliers* no desvio padrão.

4.1.2 Parâmetros dos aprimoramentos

Para escolher os parâmetros dos métodos implementados, uma busca entre combinações de vários parâmetros foi feita. A melhor escolha deve se basear em uma boa relação entre tempo de execução e acurácia com relação ao MUSIC de busca exaustiva com resolução de $0,05^\circ$. A acurácia mínima escolhida foi baseada em um erro máximo de 15 cm dado uma distância de 30 m entre o transmissor e o receptor. Assim, o erro angular médio máximo dos aprimoramentos deve ser aproximadamente $0,3^\circ$.

Para evitar sobreajustes, utilizou-se como conjunto de treinamento um subconjunto de 2 mil amostras das 10 mil do conjunto de uso comum em escritório conforme discutido na Seção 3.2. Foram então executados experimentos variando os parâmetros relevantes de cada método neste conjunto de treinamento para encontrar os melhores parâmetros. Nestes experimentos foram calculados o RMSE (*Root Mean Squared Error*) e o MAE (*Mean Absolute Error*) da distância do ponto formado pelo azimute e elevação obtido em

cada experimento com a referência (azimute e elevação do MUSIC de busca exaustiva com resolução de $0,05^\circ$).

O passo grosso (p_g) é um parâmetro relevante em todos os métodos e para a busca grossa-fina é o único relevante. Segundo as equações discutidas na Seção 3.5.1 é esperado que um p_g próximo a 5° seja o mais rápido utilizando o passo fino de $0,3^\circ$ como exigido nos critérios de erro máximo.

Para os métodos baseados no Gradiente, quanto maior o p_g , mais rápida a busca grossa é feita, porém maior a chance de cair próximo a um máximo local. Além disso, mais longe fica do máximo global e assim pode demorar mais para convergir.

Também, um parâmetro importante para todos os métodos baseados no Gradiente é o valor de convergência: ε . Como base em alguns testes preliminares, se observou que com $\varepsilon = 1 \times 10^{-4}$, a acurácia desejada era alcançada sem muito ganho ao diminuir este valor. Portanto este parâmetro não foi alterado durante os experimentos.

Para o método do Gradiente simples com ou sem *learning rate* (α) adaptável há dois parâmetros relevantes: p_g , como mencionado anteriormente, e α . No caso do *learning rate* fixo, quanto menor o α , mais lento, porém mais fácil de convergir. No método com α adaptável, pode-se começar com um *learning rate* maior sem comprometer muito a convergência.

Para o método *Momentum* além do p_g e α , o fator *momentum* (β) também é importante, pois pode ajudar a subir mais rapidamente as rampas quando o seu *learning rate* for pequeno. O método do Gradiente *Nesterov Momentum* segue a mesma lógica do *Momentum* no quesito dos parâmetros.

4.1.3 Acurácia dos aprimoramentos e tempo de execução

Para os testes de acurácia calculou-se o erro dos resultados dos algoritmos com os melhores parâmetros em relação a referência (MUSIC busca exaustiva) como comentado na Seção 4.1.2. Foi então calculado o RMSE e o MAE de todos os métodos para todo o conjunto de amostras de uso comum em escritório (10 mil amostras). Foram comparados tanto os erros de azimute e elevação separadamente quanto calculando a distância deste ponto até o ponto referência. O MAE foi útil para verificar a contribuição dos *outliers*.

Como foi observado na Seção 3.4, aproximadamente 1% das amostras apresentaram aquelas características que dificultam a determinação da direção de chegada. Então, analisou-se o impacto das piores amostras, descartando 1% dos piores valores de erro do cálculo. Além do critério de acurácia mínima, o ponto principal deste trabalho é a redução do tempo computacional. Então o tempo de execução também foi medido para verificar qual método é o mais rápido.

4.2 Resultados

4.2.1 Precisão do MUSIC e ESPRIT

Os resultados dos experimentos de precisão com as amostras de pontos fixos podem ser vistos na Tabela 1. Além do MUSIC e ESPRIT implementados neste trabalho, também foi medido a variância da API da Silabs. Como pode ser observado, nos 3 conjuntos de amostras de ponto fixo a Silabs teve o menor desvio absoluto médio (MAD) e o menor desvio padrão (σ) tanto nos desvios de distância (dst.) quanto nos de azimute (az.) e elevação (el.). Isso pode estar ocorrendo devido a presença de um filtro que é incluído no modo de funcionamento básico da API [25]. Um filtro em amostras de um objeto parado ameniza muito os efeitos de *outliers*, o que não está presente nos métodos implementados neste trabalho.

Outro ponto interessante é que o desvio padrão do MUSIC está pior do que o do ESPRIT, porém o MAD está menor. Uma possível interpretação para este fenômeno é que o MUSIC está tendo mais dificuldades com amostras anômalas, seja por interferências, reflexões ou qualquer outro fenômeno.

		MAD dst.	σ dst.	MAD az.	σ az.	MAD el.	σ el.
Conj. 1	MUSIC	4,1975°	8,6128°	2,0365°	8,5105°	4,7399°	6,5641°
	ESPRIT	4,6589°	6,4586°	1,8465°	6,0749°	5,3191°	7,1224°
	Silabs	0,9300°	1,1788°	0,9057°	1,1539°	1,7972°	2,2666°
Conj. 2	MUSIC	3,4977°	10,9712°	2,8472°	11,0574°	5,8038°	7,2491°
	ESPRIT	3,7891°	5,9085°	2,3586°	6,6148°	4,3858°	6,0278°
	Silabs	1,3358°	1,6246°	1,3659°	1,6542°	1,9405°	2,4688°
Conj. 3	MUSIC	2,0604°	6,0833°	1,8215°	6,5930°	1,7973°	2,9670°
	ESPRIT	2,2989°	6,3690°	2,1553°	11,6199°	3,0095°	4,0780°
	Silabs	1,0005°	1,2667°	0,7524°	0,9653°	1,0893°	1,3761°

Tabela 1 – Resultados experimentais de precisão do MUSIC, ESPRIT e API da Silabs para amostras de ponto fixo.

4.2.2 Análise dos parâmetros dos aprimoramentos

Em todos os experimentos de análise de parâmetros vistos a seguir, o RMSE é bem maior que o MAE. Isto significa que há *outliers* que contribuem muito para o erro. Assim, utilizou-se o MAE ao invés do RMSE para as análises, pois isso diminui a distorção dos resultados pelos *outliers*.

Sobre estes *outliers*, na Seção 3.4 foi comentado que aproximadamente 1% das amostras tem máximos globais indefinidos e que uma possível solução é tentar identificar essas amostras e descartá-las. Estes *outliers* observados podem estar vindo dessas amostras. Para se ter uma ideia de quanto esses *outliers* estão contribuindo com o alto RMSE,

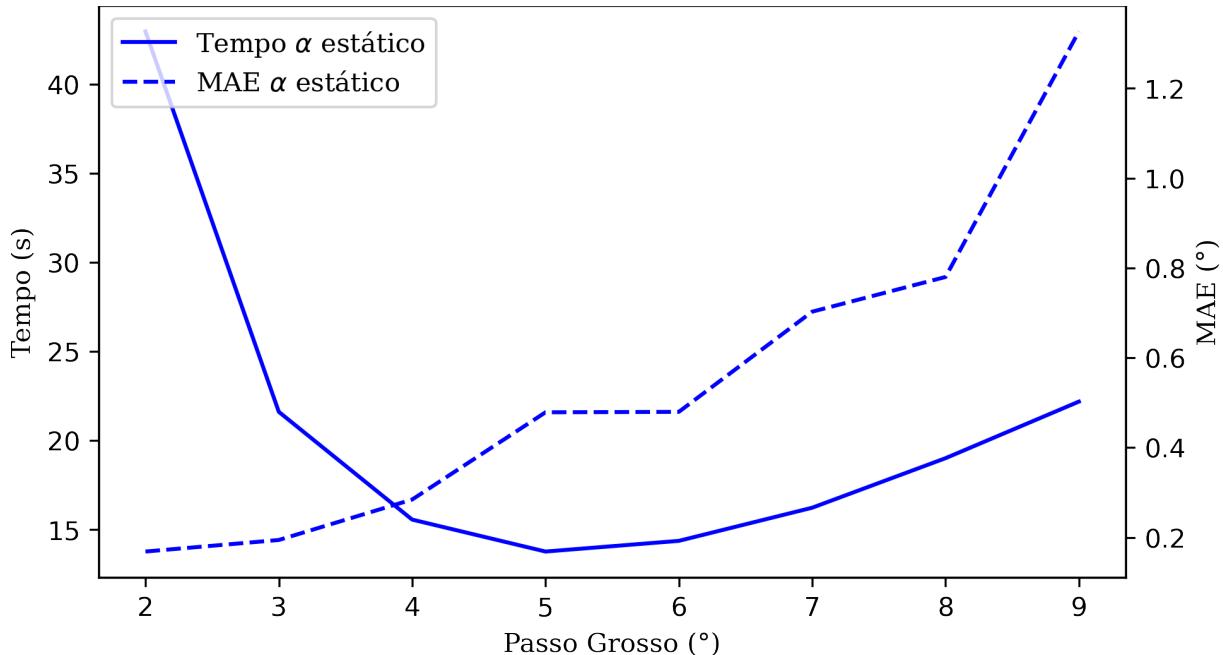


Figura 16 – Variação de p_g para o método de busca grossa-fina.

descartou-se 1% das amostras com maiores erros e calculado o RMSE desse novo conjunto. Os resultados encontrados foram RMSE muito menores que os anteriores como pode ser visto nesta seção. Esse RMSE foi denominado "RMSE 1%".

4.2.2.1 Busca grossa-fina

Foram executados testes para a busca grossa-fina com diferentes passos grossos (p_g) e os resultados estão ilustrados na Figura 16. Como esperado, a execução mais rápida é com passo grosso próximo a 5° e conforme se afasta deste ponto aumenta o tempo de execução. Isso ocorre devido ao número de iterações realizadas conforme comentado na Seção 3.5.1. Além disso, quanto maior o p_g também maior o MAE.

4.2.2.2 Gradiente simples

Nesta seção iremos abordar os métodos de Gradiente simples com *learning rate* (α) estático e adaptável. Para poder fazer uma análise do comportamento do tempo e do MAE com a variação de α , foi fixado o passo grosso em 5° (um valor mediano bom para ambos os métodos).

No caso do *learning rate* estático, como pode ser observado na Figura 17, a tendência é que com o aumento do α , pior o MAE. Isto provavelmente se deve devido a maior dificuldade de convergir ao chegar próximo ao pico.

Já no caso do *learning rate* adaptável o MAE se mantém baixo até que o α passe de 0,8. Como o α vai diminuindo conforme o algoritmo cruza o pico, a tendência é que ele consiga convergir mesmo começando com *learning rates* mais altos.

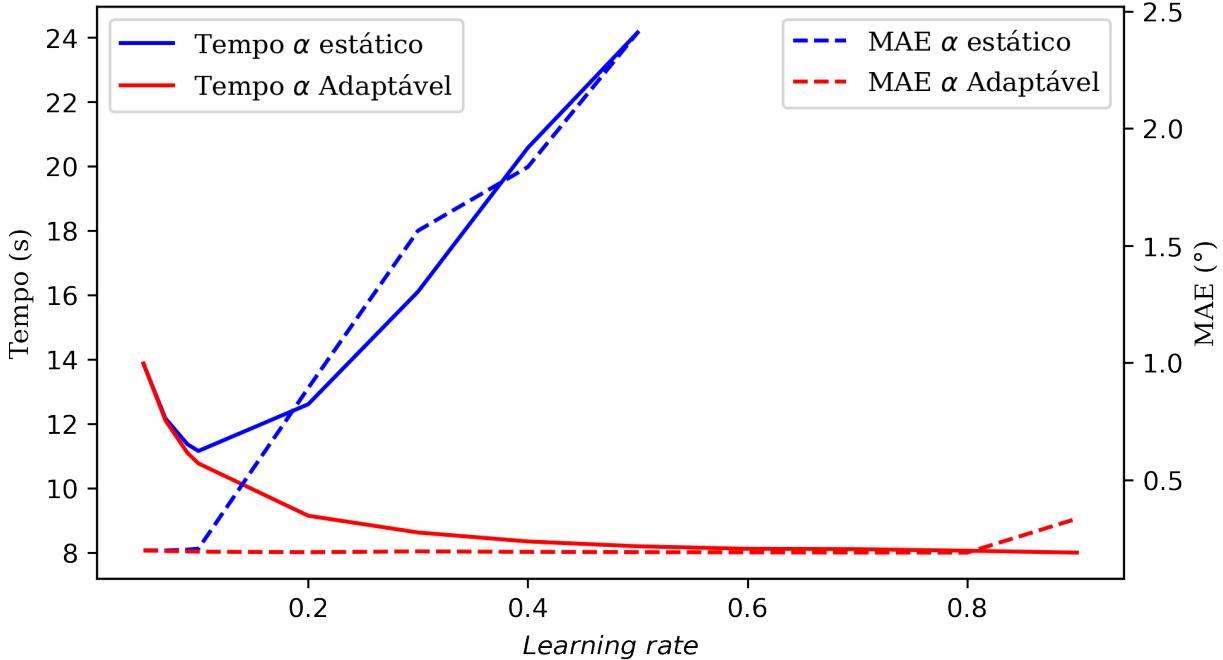


Figura 17 – Variação de α para $p_g = 5^\circ$ do método Gradiente simples com α estático e adaptável.

Para a análise de p_g , fixou-se o α em $0,1^\circ$ para o método com *learning rate* estático e $\alpha = 0,7^\circ$ para o método com *learning rate* adaptável. Isto foi feito para mostrar a vantagem do método com α adaptável: poder ser utilizado com α maior sem prejudicar o MAE como fica ilustrado na Figura 18.

Quanto maior o p_g , mais rápida é a busca grossa. No algoritmo de busca grossa-fina no entanto, mais buscas com um passo menor teriam que ser realizadas, aumentando o tempo de execução total. Já no caso do Gradiente, ele caminha na direção do pico e assim com menos iterações consegue chegar no máximo, portanto o gargalo do método acaba sendo a busca grossa. Assim, com um p_g maior, a busca grossa é mais rápida e o tempo de execução total é menor.

Esta aumento de velocidade de computação no entanto vem ao custo do aumento do erro. O MAE piora com o aumento do passo grosso, pois aumenta a probabilidade de, na busca grossa, cair perto de algum pico diferente do máximo global. Então há uma troca entre tempo de execução e acurácia, e os parâmetros devem ser escolhidos conforme a necessidade da aplicação.

4.2.2.3 Momentum e Nesterov

Por último falta investigar os dois algoritmos baseados em *momentum*. Para esses algoritmos foram analisados três parâmetros: *momentum* (β), *learning rate* (α) e o passo grosso (p_g). Para analisar o impacto de cada parâmetro isoladamente fixou-se os valores dos outros parâmetros, assim como feito anteriormente.

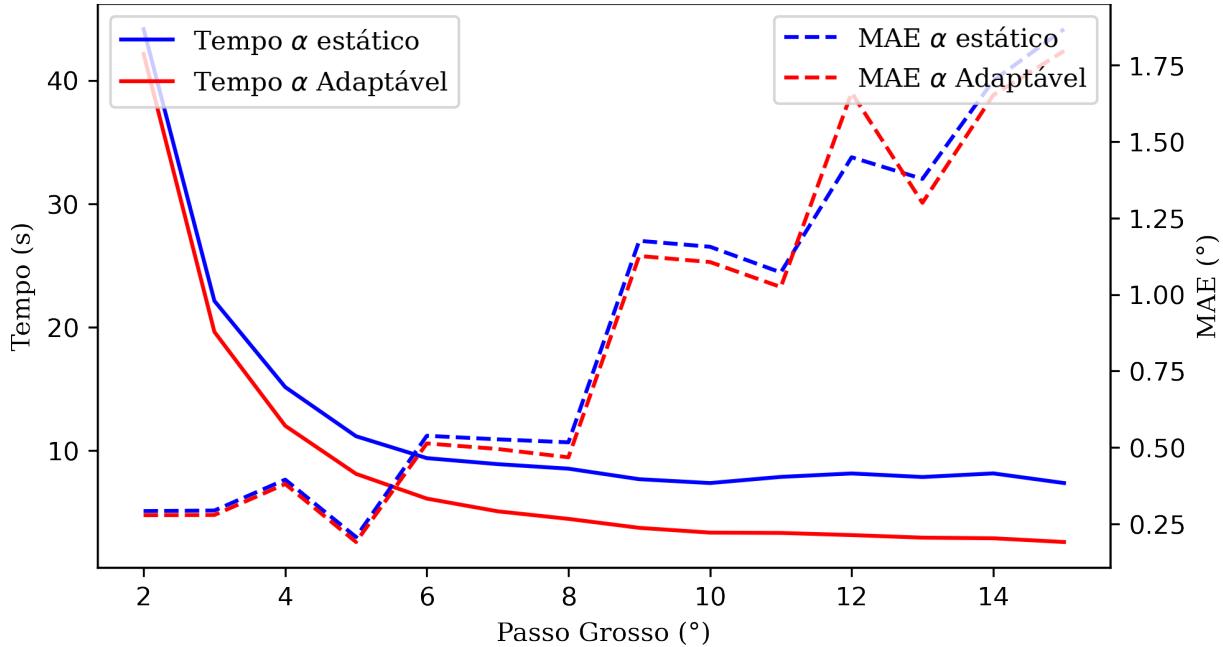


Figura 18 – Variação de p_g com $\alpha = 0,1$ para o método Gradiente simples estático e $\alpha = 0,7$ para o método adaptável.

Primeiramente, analisou-se o impacto do β no tempo de execução e no MAE. Para isso foi definido $p_g = 5$ e $\alpha = 0,07^\circ$ pois estes métodos são melhores com um *learning rate* menor. Ambos os métodos se comportaram de forma parecida para este valor de α , pois como ele é bem pequeno, o passo futuro (como utilizado no *Nesterov*) não diferencia muito do atual. Em ambos os casos a melhor relação entre tempo de execução e erro médio parece estar entre $\beta = 0,85$ e $\beta = 0,9$.

Para se aprofundar na análise do α , fixou-se o p_g em 5° e o β em $0,9$, pois pelos dados analisados anteriormente, ambos os métodos melhoram o MAE e o tempo de execução perto deste valor. O resultado destes testes podem ser visualizados na Figura 20. Como se pode observar, valores próximos de $0,07^\circ$ são melhores tanto em tempo de execução quanto em erro médio. Isso ocorre pois o algoritmo converge mais rápido com um *learning rate* menor e, como ele acelera o passo conforme avança, ele consegue chegar rapidamente no máximo.

Por último, para o passo grosso, foi definido $\alpha = 0,07^\circ$ e $\beta = 0,9^\circ$ por ser os melhores valores analisados até agora. Os valores de tempo de execução e MAE estão mostrados na Figura 21. Estes resultados são parecidos com o que ocorre para o Gradiente sem *momentum* e as mesmas observações feitas na Seção 4.2.2.2 se aplicam aqui.

4.2.3 Escolha dos melhores parâmetros

Como comentado na Seção 4.1.2, foi executada uma busca exaustiva com várias combinações de parâmetros para todos os métodos. No caso da busca grossa-fina, só há

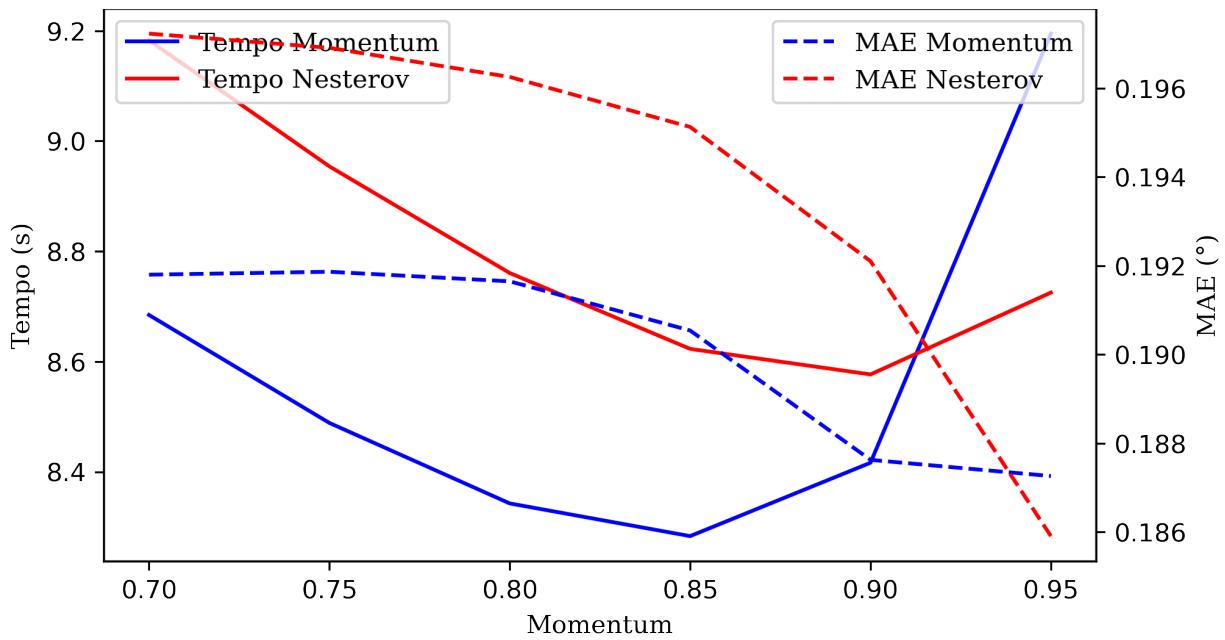


Figura 19 – Variação de β com $p_g = 5^\circ$ e $\alpha = 0,07$ para os métodos *Momentum* e *Nesterov Momentum*.

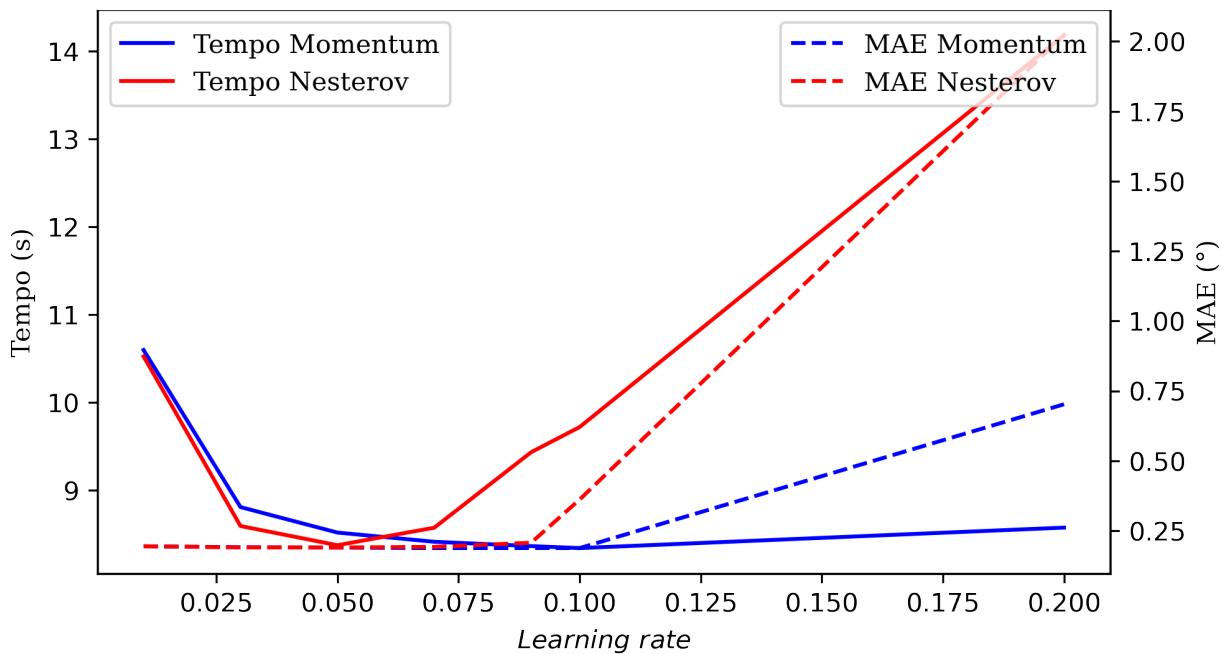


Figura 20 – Variação de α com $p_g = 5^\circ$ e $\beta = 0,9$ para os métodos *Momentum* e *Nesterov Momentum*.

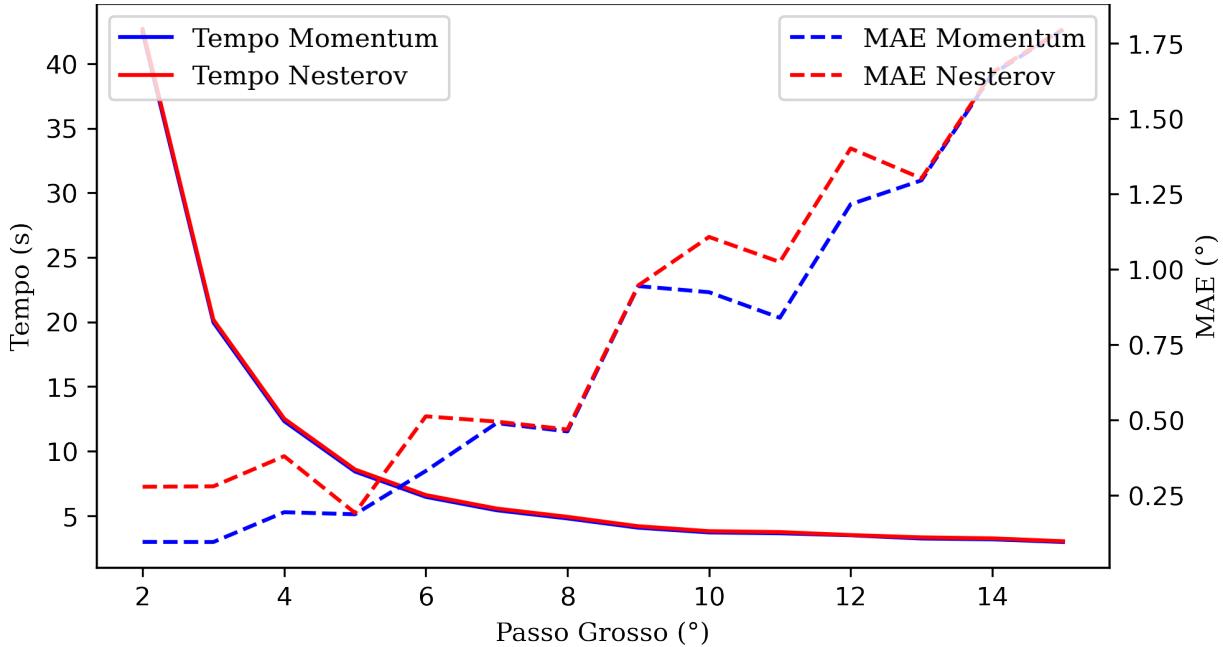


Figura 21 – Variação de p_g com $\alpha = 0,07$ e $\beta = 0,9$ para os métodos *Momentum* e *Nesterov Momentum*.

p_g	Tempo (s)	MAE	RMSE	RMSE 1%
2°	42,9763	0,1687°	1,6013°	0,1330°
3°	21,6049	0,1944°	1,8114°	0,1350°
4°	15,5510	0,2845°	4,1453°	0,1384°
5°	13,7561	0,4787°	9,0651°	0,1381°
6°	14,3563	0,4797°	7,3349°	0,1365°
7°	16,2182	0,7028°	9,2024°	0,1990°
8°	19,0011	0,7801°	8,8870°	0,2136°
9°	22,1796	1,3267°	14,5058°	0,9829°

Tabela 2 – Resultados experimentais para escolha do parâmetro p_g para a busca grossa-fina.

um parâmetro: passo grosso. Neste caso foram executados testes para a busca grossa-fina com (p_g) de 2° a 9° (Tabela 2). Conforme o esperado, o passo com o menor tempo de execução foi o de 5° , porém nesta resolução o critério de acurácia mínima não é atingido. Utilizou-se $p_g = 3^\circ$ nos testes finais de acurácia, para ser mais conservador.

Para escolher os melhores parâmetros dos outros métodos, foi passado um filtro em todos os experimentos executados selecionando pelo critério de acurácia mínima ($MAE < 0,3^\circ$). Após esse filtro, os experimentos foram ordenadas em ordem crescente de tempo de execução. As 5 execuções mais rápidas de cada método estão mostradas nas tabelas 3, 4, 5 e 6.

Então foi escolhida a mais rápida configuração de cada método para serem executados nos testes de acurácia e tempo de execução. Estes valores podem ser vistos na Tabela 7.

p_g	α	Tempo (s)	MAE	RMSE	RMSE 1%
5°	0,10	11,1536	0,2071°	4,1054°	0,1155°
5°	0,09	11,3634	0,2031°	4,1046°	0,1105°
5°	0,07	12,1642	0,1995°	4,1069°	0,1037°
5°	0,05	13,8613	0,1994°	4,1134°	0,1002°
3°	0,10	22,1204	0,2937°	8,2159°	0,1109°

Tabela 3 – As 5 execuções mais rápidas com diferentes parâmetros e com $MAE < 0,3^\circ$ para o método do Gradiente simples com α estático.

p_g	α	Tempo (s)	MAE	RMSE	RMSE 1%
5°	0,80	8,0548	0,1903°	4,0938°	0,0982°
5°	0,70	8,1033	0,1907°	4,0940°	0,0989°
5°	0,60	8,1147	0,1913°	4,0941°	0,0991°
5°	0,50	8,1927	0,1923°	4,0946°	0,0994°
5°	0,40	8,3430	0,1936°	4,0957°	0,0997°

Tabela 4 – As 5 execuções mais rápidas com diferentes parâmetros e com $MAE < 0,3^\circ$ para o método do Gradiente simples com α adaptável.

p_g	α	β	Tempo (s)	MAE	RMSE	RMSE 1%
5°	0,10	0,80	8,1216	0,1909°	4,0932°	0,0986°
5°	0,10	0,85	8,1477	0,1894°	4,0940°	0,0974°
5°	0,09	0,80	8,1788	0,1912°	4,0933°	0,0989°
5°	0,09	0,85	8,1819	0,1897°	4,0936°	0,0983°
5°	0,10	0,75	8,1836	0,1912°	4,0935°	0,0989°

Tabela 5 – As 5 execuções mais rápidas com diferentes parâmetros e com $MAE < 0,3^\circ$ para o método *Momentum*.

p_g	α	β	Tempo (s)	MAE	RMSE	RMSE 1%
5°	0,05	0,90	8,3772	0,1899°	4,0934°	0,1010°
5°	0,05	0,85	8,4950	0,1919°	4,0933°	0,1025°
5°	0,05	0,95	8,5126	0,1840°	4,0931°	0,0929°
5°	0,07	0,90	8,5766	0,1921°	4,0948°	0,1103°
5°	0,03	0,90	8,5957	0,1903°	4,0932°	0,0989°

Tabela 6 – As 5 execuções mais rápidas com diferentes parâmetros e com $MAE < 0,3^\circ$ para o método *Nesterov Momentum*.

Método	p_g	α	β
Busca grossa-fina	4°		
Gradiente simples α estático	5°	0,09	
Gradiente simples α adaptável	5°	0,80	
Gradiente <i>Momentum</i>	5°	0,10	0,85
Gradiente <i>Nesterov Momentum</i>	5°	0,05	0,95

Tabela 7 – Melhores parâmetros de cada método usados nos testes completos de acurácia e de tempo de execução.

Por fim, se considerar que seja possível retirar as amostras deterioradas como comentado na Seção 3.4, pode-se ter uma ideia do novo erro médio utilizando RMSE 1%. Então, considerando como critério de corte $\text{RMSE } 1\% < 0,3^\circ$, as execuções mais rápidas dos métodos baseados no Gradiente se encontram na Tabela 8.

Como pode-se observar, alguns destes métodos ficaram aproximadamente três vezes mais rápidos. Isto mostra um grande potencial caso seja possível descartar as amostras deterioradas. Estes resultados no entanto não serão utilizados nos testes de acurácia e tempo de execução, pois são tópicos de trabalhos futuros.

Aprimoramento do MUSIC	p_g	α	β	Tempo (s)	RMSE 1%
Gradiente simples α estático	11°	0,10		7,8568	0,2689°
Gradiente simples α adaptável	13°	0,90		2,8062	0,1965°
Gradiente <i>Momentum</i>	13°	0,10	0,85	2,9089	0,1948°
Gradiente <i>Nesterov Momentum</i>	13°	0,05	0,95	3,2635	0,1920°

Tabela 8 – Execuções mais rápidas dos métodos baseados no Gradiente $\text{RMSE } 1\% < 0,3^\circ$.

4.2.4 Acurácia dos aprimoramentos e tempo de execução

Com os melhores parâmetros para cada método obtidos a partir do conjunto de treinamento, foram executados todos os métodos no conjunto completo de amostras de uso comum em escritório. O objetivo é executar o mais rápido possível sem perder a acurácia mínima de $0,3^\circ$. Os resultados obtidos estão mostrados na Tabela 9. Dois métodos ficaram com MAE acima do máximo exigido ($0,3^\circ$), porém ficaram próximos do limite. O método mais rápido nesta comparação foi o Gradiente com *learning rate* adaptável. Como comparação, o MUSIC de busca exaustiva com resolução $0,3^\circ$ levou 8917 segundos, ou seja, aproximadamente 220 vezes mais tempo que o aprimoramento do MUSIC mais rápido.

Por último, foi executado o método ESPRIT no mesmo conjunto de amostras que os outros métodos e seu tempo de execução foi 1,6537 s. Isso significa que o ESPRIT foi aproximadamente 24 vezes mais rápido que o melhor método do MUSIC desenvolvido neste trabalho. Além disso, como visto nos testes de precisão, o MAD é bem próximo dos

Aprimoramento do MUSIC	Tempo		MAE	RMSE
	Total (s)	Amostra (ms)		
Busca grossa-fina	108,1367	10,8137	0,3367°	7,2494°
Gradiente simples α estático	56,8658	5,6866	0,2845°	6,2201°
Gradiente simples α adaptável	39,9917	3,9992	0,3205°	7,2172°
Gradiente <i>Momentum</i>	41,2383	4,1238	0,2796°	6,5136°
Gradiente <i>Nesterov Momentum</i>	43,1562	4,3156	0,2588°	6,2032°

Tabela 9 – Resultado experimentais de acurácia e de tempo de execução dos métodos de aprimoramento do MUSIC.

dois métodos, fazendo com que o ESPRIT pareça ser a escolha mais óbvia dado que ele seja aplicável no arranjo de antenas utilizado.

No entanto, algumas coisas pesam a favor do MUSIC. A primeira é que durante a busca grossa realizada, é possível tentar identificar e descartar amostras deterioradas. Isso traria além de maior acurácia, maior velocidade de execução do MUSIC como pode ser observado na Tabela 8. Também, outros métodos de otimização de achar o máximo global podem ser implementados para melhorar ainda mais o tempo de execução.

4.3 Validação em tempo real

Os algoritmos foram executados em tempo real utilizando o aplicativo desenvolvido em [1]. Com uma taxa de transmissão do *beacon* de 100 ms, todos os métodos de aprimoramento conseguiram processar as amostras com folga. Utilizando as médias de tempo de processamento por amostra da Tabela 9, pode-se ter uma ideia de quantos transmissores ao mesmo tempo o módulo de processamento utilizado neste trabalho seria capaz de processar.

Para efeito de comparação, foi executado o aplicativo também com o MUSIC sem nenhum aprimoramento e ficou inviável processar os ângulos, demorando aproximadamente 1 segundo por amostra.

5 CONCLUSÃO

A utilização de métodos de estimação de direção de chegada tem se tornado cada vez mais comum. Entre os métodos mais comuns estão o MUSIC e o ESPRIT. O MUSIC é mais genérico podendo ser utilizado em qualquer geometria de um arranjo de antenas, no entanto é esperado que ele seja mais lento que o ESPRIT, pois depende de uma busca exaustiva em um pseudo espectro. Algumas melhorias para o MUSIC foram propostas, mas nem todas são aplicáveis a todas as geometrias. Uma melhoria simples e aplicável a qualquer geometria é a busca em dois estágios, um com resolução menor e outro com alta resolução. Assim consegue-se reduzir consideravelmente o tempo de busca, porém pode acarretar em erros.

Além da análise deste método de aprimoramento, é proposto outras alternativas para a busca de alta resolução, baseadas nas análises do pseudo espectro do MUSIC. Foram estudados ao todo cinco métodos baseados no MUSIC. Todas estas técnicas têm como princípio uma busca grossa e depois, para achar o máximo global, elas usam: busca fina local, Gradiente simples, Gradiente com *learning rate* adaptável, Gradiente *Momentum* e Gradiente *Nesterov Momentum*.

Este trabalho propõe implementar e analisar o tempo de execução e o erro destes aprimoramentos com relação ao MUSIC de busca exaustiva. Além disso, também foi averiguado o uso destes métodos em uma aplicação de estimação de direção de chegada em tempo real.

Os aprimoramentos reduziram drasticamente o tempo de execução (até 220 vezes mais rápido) mantendo o mesmo patamar de acurácia. Com isso, foi possível utilizar o MUSIC em tempo real, o que permite o uso de conjuntos de antenas com geometria nas quais o ESPRIT não poderia ser utilizado (como no caso do URA-8). Além disso, foi observado que na etapa da busca grossa poderia ser feita uma análise no pseudo espectro para excluir amostras nas quais o máximo global é dúbio.

Apesar dos resultados apresentados, o MUSIC mais rápido implementado ainda é 24 vezes mais lento que o ESPRIT, mantendo quase a mesma variância. Assim seria interessante implementar outros métodos de otimização como o *Adadelta* ou o *Adam* pois eles adicionam um *learning rate* adaptável aos métodos com *momentum*. Também poderia ser implementado uma análise que permita o descarte de amostras ruins durante a etapa de busca grossa. Além disso, também é interessante colocar um filtro na saída do resultado do MUSIC implementado neste trabalho e ver se o seu resultado chega próximo ao do API da Silabs. Um outro ponto não abordado neste trabalho é a utilização de outras bibliotecas de álgebra linear como a *OpenBLAS* e verificar se isso melhora o processamento. Por

fim seria interessante observar como o pseudo espectro do MUSIC se comporta em uma simulação que consiga isolar eventos como interferências e reflexões.

Referências

- 1 LEMOS, P. Estimação de ângulo de chegada em arranjos de antenas: comparação e implementação em tempo real. p. 66, 12 2020. TCC - UFSC. Centro Tecnológico. Engenharia Elétrica. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/219643/TCCPedro_vFinal.pdf>. Citado 6 vezes nas páginas 13, 24, 27, 28, 40 e 59.
- 2 HEIDENREICH, P.; ZOUBIR, A. M.; RUBSAMEN, M. Joint 2-d DOA estimation and phase calibration for uniform rectangular arrays. *IEEE Transactions on Signal Processing*, v. 60, n. 9, p. 4683–4693, 09 2012. ISSN 1941-0476. Disponível em: <<https://ieeexplore.ieee.org/document/6213139>>. Citado 2 vezes nas páginas 13 e 28.
- 3 PAU, G. et al. Bluetooth 5.1: An analysis of direction finding capability for high-precision location services. v. 21, n. 11, p. 3589, 05 2021. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/21/11/3589>>. Acesso em: 2022-03-06. Citado 3 vezes nas páginas 13, 37 e 41.
- 4 SILICON Labs. Thunderboard EFR32BG22 User's Guide. p. 34, 06 2021. Disponível em: <<https://www.silabs.com/documents/public/user-guides/ug464-brd4184b-user-guide.pdf>>. Acesso em: 2022-03-06. Citado 3 vezes nas páginas 13, 37 e 38.
- 5 SILICON Labs. EFR32xG22 2.4 GHz 6 dBm QFN32 Wireless Starter Kit User's Guide. p. 32, 02 2020. Disponível em: <<https://www.silabs.com/documents/public/user-guides/ug431-brd4183a-user-guide.pdf>>. Acesso em: 2022-03-06. Citado 3 vezes nas páginas 13, 37 e 38.
- 6 SILICON Labs. EFR32BG22 Direction Finding Radio Board BRD4185A Reference Manual. p. 20, 06 2020. Disponível em: <<https://www.silabs.com/documents/public/reference-manuals/brd4185a-rm.pdf>>. Acesso em: 2022-03-06. Citado 3 vezes nas páginas 13, 37 e 39.
- 7 WIKIPEDIA. *Raspberry Pi*. 2022. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=62858502>. Acesso em: 2022-03-11. Citado 2 vezes nas páginas 13 e 39.
- 8 QUUPPA. *Real Time Location System applications*. 2022. Disponível em: <<https://www.quuppa.com/solutions/>>. Acesso em: 2022-03-01. Citado na página 23.
- 9 KJAERGAARD, M. B. et al. Indoor positioning using GPS revisited. In: FLOREEN, P.; KRUGER, A.; SPASOJEVIC, M. (Ed.). *Pervasive Computing*. [S.l.]: Springer Berlin Heidelberg, 2010. p. 38–56. ISBN 978-3-642-12654-3. Citado na página 23.
- 10 CONSTANDACHE, I. et al. EnLoc: Energy-efficient localization for mobile phones. In: *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*. IEEE, 2009. p. 2716–2720. ISBN 978-1-4244-3512-8. Disponível em: <[http://ieeexplore.ieee.org/document/5062218/](https://ieeexplore.ieee.org/document/5062218/)>. Acesso em: 2022-02-22. Citado na página 23.
- 11 ZAFARI, F.; GKELIAS, A.; LEUNG, K. K. A survey of indoor localization systems and technologies. *Conference Name: IEEE Communications Surveys*

Tutorials, v. 21, n. 3, p. 2568–2599, 08 2019. ISSN 1553-877X. Disponível em: <<https://ieeexplore.ieee.org/document/8692423?arnumber=8692423>>. Citado 2 vezes nas páginas 23 e 37.

12 ROY, R.; KAILATH, T. ESPRIT-estimation of signal parameters via rotational invariance techniques. v. 37, n. 7, p. 984–995, 1989. ISSN 0096-3518. Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing. Citado na página 23.

13 PIAZZESE, N. I.; CHEPYK, O.; PAU, D. P. Investigating the complexity-performance tradeoff of ura8 topology for bluetooth 5.1 haat. In: *2021 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*. [s.n.], 2021. p. 109–114. Disponível em: <<https://ieeexplore.ieee.org/document/9650486/>>. Citado 3 vezes nas páginas 23, 24 e 43.

14 WAWERU, N. P.; KONDITI, D. B. O.; LANGAT, P. K. Performance Analysis of MUSIC, Root-MUSIC and ESPRIT DOA Estimation Algorithm. Zenodo, 03 2014. Disponível em: <<https://doi.org/10.5281/zenodo.1091734>>. Citado 3 vezes nas páginas 23, 29 e 30.

15 SCHMIDT, R. Multiple emitter location and signal parameter estimation. v. 34, n. 3, p. 276–280, 1986. ISSN 1558-2221. Conference Name: IEEE Transactions on Antennas and Propagation. Citado na página 23.

16 HWANG, H. et al. Direction of arrival estimation using a root-music algorithm. *Lecture Notes in Engineering and Computer Science*, v. 2169, 03 2008. Citado na página 23.

17 ZHUANG, J. et al. Low complexity 2-d DOA estimator for arbitrary arrays: A hybrid MUSIC-based method. In: *2015 10th International Conference on Information, Communications and Signal Processing (ICICS)*. IEEE, 2015. p. 1–4. ISBN 978-1-4673-7218-3. Disponível em: <<http://ieeexplore.ieee.org/document/7459890/>>. Citado na página 23.

18 BALANIS, C. *Antenna Theory: Analysis and Design*. [S.l.]: Wiley, 1997. ISBN 978-1-118-58573-3. Citado na página 27.

19 JAIN, A. K. *Fundamentals of Digital Image Processing*. [S.l.]: Prentice-Hall, Inc., 1989. ISBN 0-13-336165-9. Citado na página 29.

20 KOCHENDERFER, M. J.; WHEELER, T. A. *Algorithms for Optimization*. The MIT Press, 2019. ISBN 0-262-03942-7. Disponível em: <<https://algorithmsbook.com/optimization/files/optimization.pdf>>. Acesso em: 2022-03-08. Citado 4 vezes nas páginas 34, 35, 47 e 48.

21 Bluetooth SIG. *Bluetooth Technology Overview*. 2022. Disponível em: <<https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>>. Acesso em: 2022-03-06. Citado na página 37.

22 RASPBERRY Pi Foundation. Raspberry pi 4 model b datasheet. 2019. Disponível em: <<https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>>. Acesso em: 2022-03-06. Citado na página 38.

- 23 ARM. ARM Cortex-A72 MPCore Processor Technical Reference Manual. p. 575, 2016. Disponível em: <<https://documentation-service.arm.com/static/60368ce38f952d2e4134dc2e?token=>>. Acesso em: 2022-03-06. Citado na página 38.
- 24 EIGEN. *Benchmark Eigen*. 2021. Disponível em: <<https://eigen.tuxfamily.org/index.php?title=Benchmark>>. Acesso em: 2022-03-06. Citado na página 40.
- 25 SILICON Labs. *Real-Time Locationing (RTL) Library - v3.2 - RTL Library API Documentation Silicon Labs*. 2021. Disponível em: <<https://docs.silabs.com/rtl-lib/3.2/>>. Acesso em: 2022-03-11. Citado na página 51.