

UNIVERSIDADE FEDERAL DO ABC
CENTRO DE ENGENHARIA, MODELAGEM E CIÊNCIAS SOCIAIS APLICADAS - CECS

Heitor Rodrigues Savegnago

**ANÁLISE DE SISTEMAS DE GEOLOCALIZAÇÃO BASEADOS EM
GPS E AOA PARA FOGUETES DE SONDAGEM ATMOSFÉRICA**

Santo André, SP
2024

Heitor Rodrigues Savegnago

ANÁLISE DE SISTEMAS DE GEOLOCALIZAÇÃO BASEADOS EM GPS E AOA PARA FOGUETES DE SONDAGEM ATMOSFÉRICA

Trabalho de Conclusão de Curso apresentado ao Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas da Universidade Federal do ABC como requisito parcial à obtenção do título de Bacharel em Engenharia de Informação.

Orientador: Prof. Dr. Ivan Roberto de Santana Casella.

Santo André, SP
2024

Agradecimentos

Resumo

Alterar

O objetivo principal do projeto é a concepção de um sistema de localização relativa para um foguete de sondagem atmosférica. Este sistema servirá para guiar um grupo de busca em campo na tarefa de localizar o veículo após o aterrizagem.

O direcionamento da busca seria feita utilizando o método de ângulo de chegada relativo ao sinal oriundo da telemetria do foguete. O sinal será recebido por uma matriz de antenas, viabilizando os cálculos de ângulo.

O objetivo secundário do projeto é comparar a performance deste sistema com a performance de outro sistema de localização relativa, que utiliza coordenadas geográficas para os cálculos. Este outro sistema se baseia em cálculos de azimuth entre as coordenadas do veículo e do grupo de busca.

Por se tratar de um sistema com dados processados no veículo, se houverem problemas internos com o GPS de bordo, estes cálculos são inviabilizados. Outro problema deste sistema é relativo à precisão da coordenada geográfica, fazendo com que, à certa distância, a precisão da direção perca sua confiabilidade.

Por outro lado, a versão utilizando o ângulo de chegada não depende dos dados transmitidos no sinal, bem como poderá funcionar bem a distâncias mais curtas.

Figuras

1	Caption	1
2	Representação geométrica de distância e ângulo em relação ao norte entre coordenadas geográficas A e B	3
3	Cálculo do ângulo de <i>Bearing</i> β entre as coordenadas dos Campi Santo André e São Bernardo do Campo da UFABC.	5
4	Característica de frente de onda a cada λ a partir da antena.....	5
5	Ângulo de Chegada (<i>Angle of Arrival</i> , AoA) com par de antenas em diversas direções equivalentes.	7
6	Diferentes valores para d	7
7	Possível disposição de matriz de antenas.....	8

Tabelas

Códigos

../code/azimuth.m	16
../code/xyt.m	16
../code/generate_fig.m.....	19
../code/signal_r.m.....	23
../code/ref_cos.m.....	24
../code/ref_sin.m	24
../code/argument_r.m	24
../code/isoctave.m.....	24

Abreviaturas e Siglas

AoA	Ângulo de Chegada (<i>Angle of Arrival</i>)
FFT	Transformada Rápida de Fourier (<i>Fast Fourier Transform</i>)
GNSS	Sistema Global de Navegação por Satélite (<i>Global Navigation Satellite System</i>)
GPS	Sistema de Posicionamento Global (<i>Global Positioning System</i>)
IoT	Internet das Coisas (<i>Internet of Things</i>)
LoRa	<i>Longe Range</i>
LoS	Linha de visão (<i>Line of Sight</i>)
NLoS	Sem Linha de visão (<i>Non Line of Sight</i>)
RF	Radiofrequência (<i>Radio Frequency</i>)
SDR	Rádio Definido por Software (<i>Software-Defined Radio</i>)
SNR	Relação Sinal-Ruído (<i>Signal-Noise Ratio</i>)
UCA	Matriz Circular Uniforme (<i>Uniform Circular Array</i>)
ULA	Matriz Linear Uniforme (<i>Uniform Linear Array</i>)

Sumário

1	INTRODUÇÃO	1
1.1	Motivação	1
1.2	Objetivos	2
1.3	Estrutura do documento	2
2	REVISÃO BIBLIOGRÁFICA	3
2.1	Fundamentação teórica	3
2.1.1	Direcionamento por coordenadas geográficas	3
2.1.2	Estimar AoA utilizando matriz de antenas	4
2.2	Trabalhos relacionados	8
3	METODOLOGIA	11
4	RESULTADOS	12
5	CONCLUSÃO	13
	REFERÊNCIAS	14
	APÊNDICES	16
A	CÓDIGOS DESENVOLVIDOS PARA SIMULAÇÃO	16
A.1	Simulação de direcionamento GNSS	16
A.2	Simulação de AoA	16

1 Introdução

1.1 Motivação

Foguetes de sondagem atmosférica são veículos aeroespaciais sub-orbitais utilizados para levar sensores e experimentos científicos a altos níveis atmosféricos, com o intuito de realizar estudos e análises relacionados às diversas condições ali presentes [1]. Estes veículos geralmente utilizam motor de propelente sólido, de um ou dois estágios, e são equipados com sistemas de controle, telemetria e recuperação, além de transportarem o experimento científico, denominado carga-paga [2, 3]. Algumas das vantagens desses veículos são o baixo custo e a menor necessidade de alcance pra sistemas de telemetria e rastreo, tendo em vista que não entram em órbita [4].

No contexto de foguetes de sondagem, existem competições que fomentam o desenvolvimento e competitividade em equipes universitárias de foguetemodelismo [5]. Algumas dessas competições tem grande parte de suas categorias definidas nas bases de foguetes de sondagem, com apogeu de voo entre 1 km e 10 km de altura acima do nível do solo. Nestes casos, a sequência de operações normal do foguete, apresentada na Figura 1, consiste em: ignição do primeiro estágio do motor, decolagem, período propulsionado, término de queima do primeiro estágio, desacoplamento do primeiro estágio, ignição do segundo estágio, segundo período propulsionado, término de queima do segundo estágio, início do período inercial balístico, apogeu, detecção do apogeu pelos sistemas embarcados e liberação do paraquedas piloto, liberação do paraquedas principal a certa altitude e finalmente o pouso [2, 3]. Desacoplamento do primeiro estágio, ignição e fase propulsionada do segundo somente se aplicam a foguetes de dois estágios.

Figura 1: Caption



Fonte: Autor

A partir do momento do pouso, o próximo objetivo nessas competições consiste em localizar o foguete, vários métodos podem ser empregados nessa situação, desde cores chamativas no veículo e paraquedas, até sinais sonoros. Essas competições geralmente recomendam, e até exigem, a presença de um Sistema Global de Navegação por Satélite (*Global Navigation Satellite System*, GNSS), capaz de transmitir as coordenadas do veículo após o pouso para localização, como um Sistema de Posicionamento Global (*Global Positioning System*, GPS) [6].

O processo de localização baseada em dados simples de GNSS, latitude e longitude, pode se tornar mais complicado se o grupo de busca não tem certeza de como encontrar essas coordenadas. Existem dispositivos de GNSS portáteis, porém estes podem criar dificuldades na interface com os dados recebidos da telemetria do foguete. Neste caso, seria possível desenvolver um dispositivo capaz de lidar diretamente com as informações de localização fornecidas pela telemetria e guiar o grupo de buscar na direção correta.

Os dados recebidos da telemetria ainda precisam de certo grau de confiabilidade para que sejam devidamente processados e tratados, o que pode ser um problema se o veículo está longe do grupo de busca ou o dispositivo de GNSS a bordo não esteja apto a fornecer dados corretamente. Nesse caso, ainda é possível buscar o foguete utilizando o próprio sinal da telemetria, independente dos dados transmitidos.

1.2 Objetivos

O principal objetivo deste trabalho é desenvolver e projetar um dispositivo portátil capaz de indicar a direção da origem de um sinal de Radiofrequência (*Radio Frequency*, RF) baseado em métodos de detecção de Ângulo de Chegada (*Angle of Arrival*, AoA).

Como objetivo secundário, a análise comparativa com um sistema de utilidade semelhante, porém baseado inteiramente em coordenadas de GNSS.

1.3 Estrutura do documento

Completar isso depois

2 Revisão Bibliográfica

2.1 Fundamentação teórica

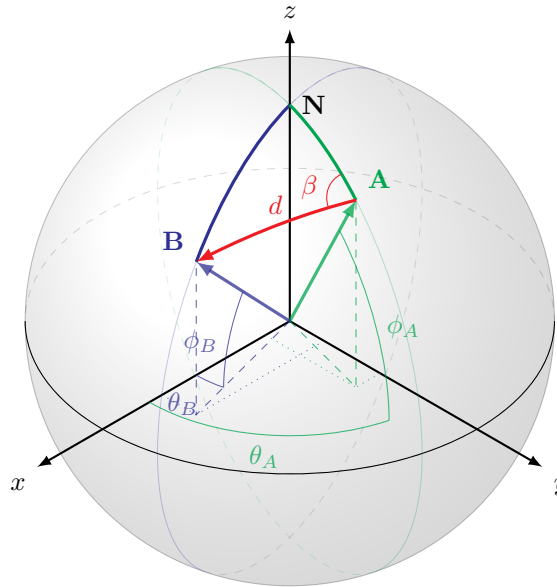
2.1.1 Direcionamento por coordenadas geográficas

Coordenadas geográficas são definidas por dois valores, latitude e longitude, associadas a coordenadas esféricas referenciadas a partir do centro da terra, assumindo o raio da coordenada como o raio médio da superfície do planeta, cerca de $R_{\text{Terra}} = 6371 \cdot 10^3 \text{ m}$ [7, 8]. A latitude equivale à componente polar ϕ centralizada na linha do equador, enquanto a longitude equivale à componente θ centralizado do meridiano de Greenwich [7, 9].

Conhecendo as coordenadas de dois pontos distintos A e B, é possível determinar seu ângulo de *bearing* β relativo, referente ao norte, ou seja, o ângulo da direção a se seguir para chegar do ponto A ao ponto B a partir da direção do norte no ponto de origem A [9].

Sendo **A** e **B** duas coordenadas geográficas, ϕ_A e ϕ_B suas respectivas latitudes, θ_A e θ_B suas respectivas longitudes, conforme ilustrado na Figura 2.

Figura 2: Representação geométrica de distância e ângulo em relação ao norte entre coordenadas geográficas **A** e **B**.



Fonte: Autor.

Calculam-se Δ_ϕ e Δ_θ .

$$\Delta_\phi = \phi_B - \phi_A \quad (1)$$

$$\Delta_\theta = \theta_B - \theta_A \quad (2)$$

Através da lei dos haversines é possível obter a distância mínima d entre as coordenadas, sobre a superfície, e também o ângulo de *Bearing* β formado no vértice **A** do triângulo esférico **NAB** [8]. Para o cálculo de distância, os ângulos devem ser tratados em radianos.

$$X = \cos(\theta_B) \cdot \sin(\Delta_\phi) \quad (3)$$

$$Y = \cos(\theta_A) \cdot \sin(\theta_B) - \sin(\theta_A) \cdot \cos(\theta_B) \cdot \cos(\Delta_\phi) \quad (4)$$

$$Z = \sin^2\left(\frac{\Delta_\theta}{2}\right) + \cos(\theta_B) \cdot \cos(\theta_A) \cdot \sin^2\left(\frac{\Delta_\phi}{2}\right) \quad (5)$$

$$\beta = \arctan\left(\frac{X}{Y}\right) - \frac{\pi}{2} \quad (6)$$

$$d = R_{\text{Terra}} \cdot 2 \cdot \arctan\left(\frac{\sqrt{Z}}{\sqrt{1-Z}}\right) \quad (7)$$

O ângulo β calculado aqui é referente à direção cardinal Norte, assim, uma equipe de busca equipada com uma bússola simples seria capaz de seguir a direção correta. A Figura 3 apresenta a aplicação desenvolvida por Veness, capaz de calcular o ângulo de *Bearing* entre duas coordenadas, note que, neste caso, o ângulo referido é relacionado à direção cardinal Leste [8].

2.1.2 Estimar AoA utilizando matriz de antenas

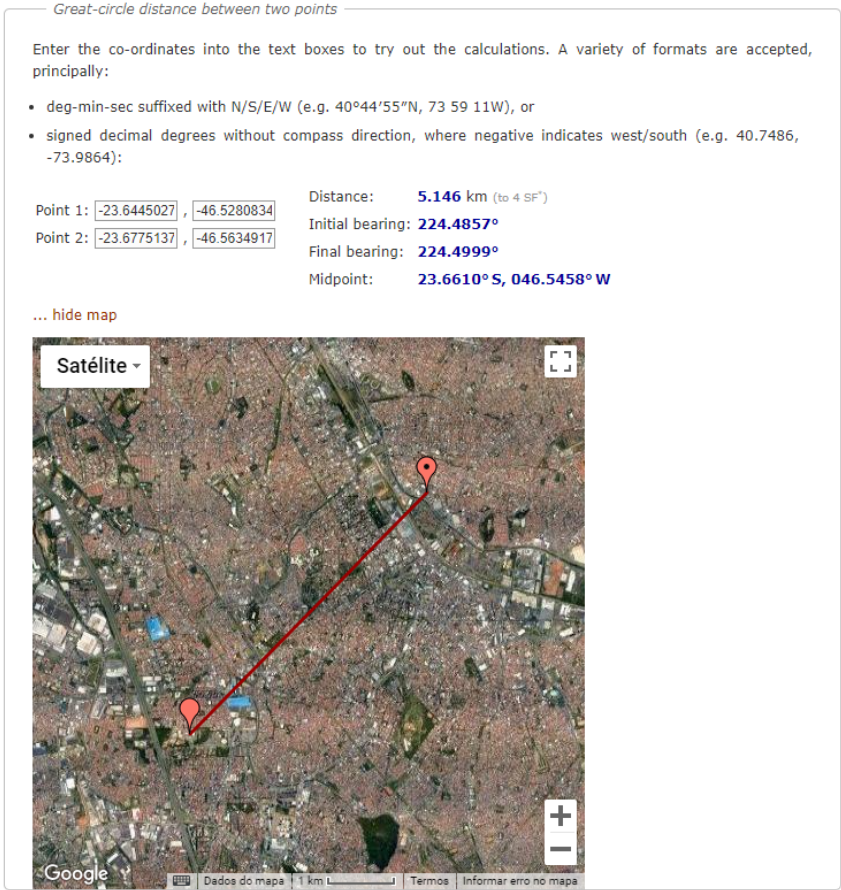
Analisando a defasagem de sinal RF em um conjunto de antenas, é possível estimar seu Ângulo de Chegada (*Angle of Arrival*, AoA), essa técnica pode ser utilizada para determinar a direção do emissor em relação à matriz de antenas utilizada. Baseando-se em dados como a distância entre as antenas, o comprimento de onda λ do sinal e a velocidade da luz no meio, usualmente tomada como $c = 299792458,6 \pm 0,3 \text{ m s}^{-1}$ no ar [10, 11, 12]. A Equação 8 apresenta a relação do comprimento de onda λ com a frequência f , com a frequência angular ω e a velocidade da luz c .

$$\lambda = \frac{c}{f} = \frac{2\pi \cdot c}{\omega} \quad (8)$$

Se um emissor estiver distante o bastante, é possível considerar que sua frente de onda tem um comportamento planar, essa característica simplifica as operações envolvidas. A distância de Fraunhofer (d_F) é a mínima para essa condição, que define o início da região de *far-field*, conforme apresentado na Equação 9, onde D é a maior dimensão da antena [13]. Tomando $D = 2\lambda$, para uma antena de dipolo, obtém-se $d_F = 8\lambda$. A Figura 4 ilustra o comportamento planar de uma frente de onda além de d_F .

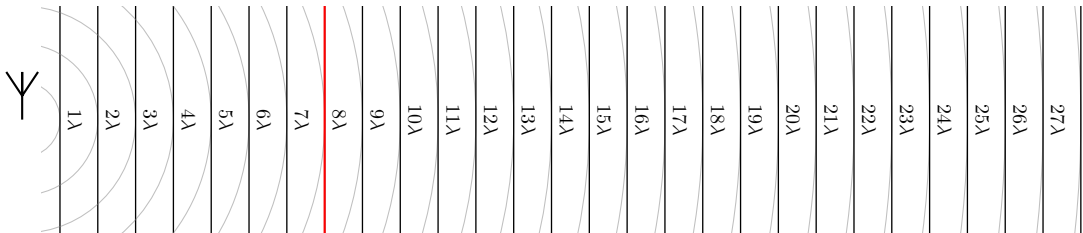
$$d_F = \frac{2 \cdot D^2}{\lambda} \Rightarrow d_F = \frac{2 \cdot (2 \cdot \lambda)^2}{\lambda} = 8\lambda \quad (9)$$

Figura 3: Cálculo do ângulo de *Bearing* β entre as coordenadas dos Campi Santo André e São Bernardo do Campo da UFABC.



Fonte: Veness 2019 [8]

Figura 4: Característica de frente de onda a cada λ a partir da antena.



Fonte: Autor.

Tomando agora um par de antenas separadas por uma distância fixa d , torna-se viável fazer a análise trigonométrica entre as antenas e a frente de onda que chega, onde essa distância d será a hipotenusa do triângulo. A Figura 5 apresenta a disposição das antenas em dois casos de chegada de sinal RF. Para realizar a análise, é necessário conhecer uma segunda dimensão do triângulo retângulo envolvido, esta é obtida analisando a defasagem entre o sinal das antenas, conforme apresentado na Equação 10.

$$d \cdot \sin(\beta) = \lambda \cdot \frac{\Delta\Phi}{2\pi} \Rightarrow \beta = \arcsin\left(\frac{\lambda}{d} \cdot \frac{\Delta\Phi}{2\pi}\right) \quad (10)$$

É importante ressaltar que um sistemas com um único par de antenas não é capaz de determinar completamente o AoA, já que o valor calculado de β é igual para casos simétricos, a relação é apresentada nas Figuras 5a e 5b. Existem ainda dois casos notáveis, onde o sinal chega alinhado com a linha das com as antenas ou perpendicular a ela, apresentados respectivamente nas Figuras 5c e 5d.

A escolha da distância d entre as antenas deve ser feita de forma a otimizar a otimizar a resolução da medida de defasagem, com a maior distância possível, porém é necessário evitar ambiguidades na análise, como o sinal é periódico, o valor se repetirá a cada λ , e terá valores simétricos quando $d > \lambda/2$, ilustrado na Figura 6a. Adota-se então $d = \lambda/2$, conforme apresentado na Figura 6b [11, 12].

Para contornar a ambiguidade gerada pela simetria no sistema, é possível adicionar uma terceira antena, de forma que não esteja alinhada com as duas primeiras. Um exemplo é apresentado na Figura 7.

Para obter o valor de defasagem entre antenas, é interessante representar o sinal recebido como um valor complexo. Uma forma de obter o valor complexo é realizando a integração de período completo no produto do sinal recebido na antena por uma senoide ou cossenoide de mesma frequência. Nas Equações 11 e 12, S e C são respectivamente proporcionais às componentes em fase e em quadratura do sinal w recebido na antena, k é uma função auxiliar que garante frequência correta em todos os operandos. É válido notar que w e k são funções de várias variáveis.

$$S = \int_0^T w(t) \cdot \sin(k(t)) \partial t \quad (11)$$

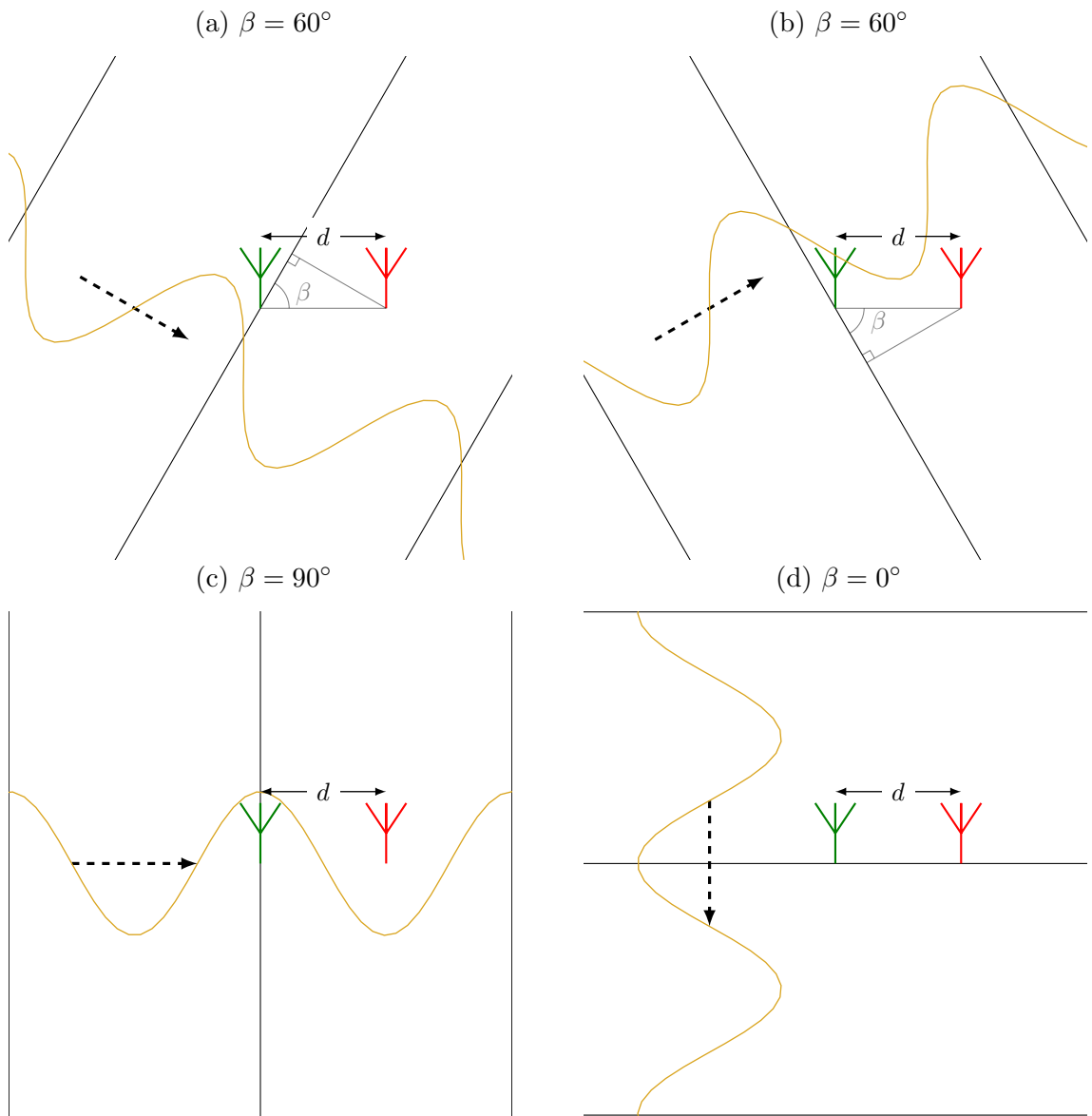
$$C = \int_0^T w(t) \cdot \cos(k(t)) \partial t \quad (12)$$

Finalmente, na Equação 13, Z é o valor complexo associada ao sinal recebido em na antena.

$$Z_{x,y} = 2 \cdot (S + \imath C) \quad (13)$$

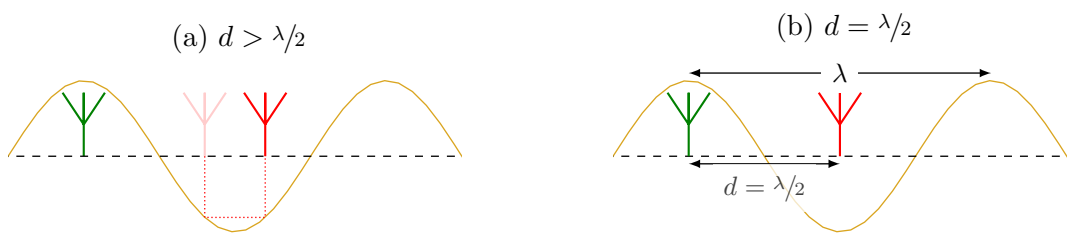
A defasagem entre um par de antenas é dado pelo ângulo do valor resultante da multiplicação do valor complexo da primeira antena pelo complexo conjugado da segunda antena, conforme apresentado na Equação 14.

Figura 5: AoA com par de antenas em diversas direções equivalentes.



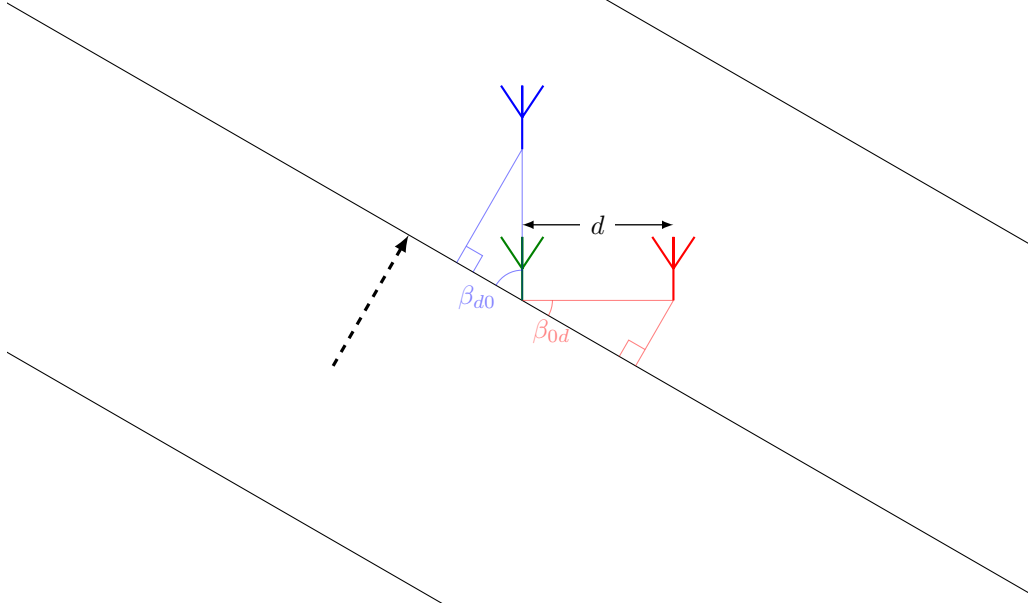
Fonte: Autor.

Figura 6: Diferentes valores para d .



Fonte: Autor.

Figura 7: Possível disposição de matriz de antenas.



Fonte: Autor.

$$\Delta\Phi_{x,y} = \arg(Z_{0,0} \cdot Z_{x,y}^*) \quad (14)$$

Com este valor é possível estimar valor de β no intervalo $0^\circ \leq \beta \leq 180^\circ$.

Utilizando a terceira antena perpendicular ao primeiro par e alinhada com uma das antenas iniciais, conforme Figura 7, é possível estimar o valor de β no intervalo $0^\circ \leq \beta \leq 360^\circ$. Cada par de antena pode indicar o valor da coordenada geométrica associada ao eixo que a caracteriza, conforme indicado na Equação 15. Essa propriedade somente é válida nessa geometria.

$$\text{componente}_{x,y} = -\frac{\Delta_{x,y}}{\pi} \cdot \frac{\cancel{d} \cdot 2}{\cancel{d}} = -\frac{\Delta_{x,y}}{\pi} \quad (15)$$

2.2 Trabalhos relacionados

Em seu trabalho, Horst [12] analisa dois algoritmos de detecção de AoA, realizando as análises em ambientes internos e utilizando matrizes de antenas. O primeiro método analisado consiste em uma aproximação do ângulo, feita utilizando um software fornecido pela Texas Instruments, fabricante do hardware utilizado. Já o segundo método, baseia-se na construção matemática do AoA baseado na diferença de fase instantânea do sinal entre as antenas do sistema, uma abordagem semelhante à proposta neste trabalho. Os resultados obtidos indicam que o método de aproximação teve melhor acurácia nos valores de ângulo.

A proposta de Zeaiter *et al.* [14] busca validar a performance da detecção de AoA em ambiente fechado, realizando a análise em diferentes modulações, larguras de canal e fatores

de espalhamento. Também propõe que, ao combinar de seu algoritmo de localização de AoA com a função de autocorrelação, é possível analisar os dados de dois sinais recebidos simultaneamente.

Outro trabalho de Zeaiter *et al.* [15] consiste em uma aproximação do AoA utilizando um método de autocorrelação em um sinal *Longe Range* (LoRa) de baixa potência. Seu objetivo consiste em detectar o sinal LoRa operando em transmissão de baixa potência, caso onde a vida útil da bateria do sistema transmissor é estendida. O algoritmo apresentado busca picos de autocorrelação no sinal recebido, além de utilizar Transformada Rápida de Fourier (*Fast Fourier Transform*, FFT) para denotá-los e melhorar a Relação Sinal-Ruído (*Signal-Noise Ratio*, SNR). Quando um pico é detectado, o algoritmo é capaz de encontrar o AoA.

BniLam *et al.* [16] propõe uma técnica que, sem qualquer informação prévia de largura de banda, consegue estimar AoA do sinal recebido. O sistema proposto consiste em uma Matriz Circular Uniforme (*Uniform Circular Array*, UCA) seguida de um filtro transversal, também utiliza de vetores especiais de largura de banda variável junto com um estimador de relação sinal-ruído térmico para determinar simultaneamente AoA e largura de banda do sinal recebido.

Em outro trabalho, BniLam *et al.* [17] estudam a possibilidade de estimar AoA para transceptores de Internet das Coisas (*Internet of Things*, IoT) em ambiente interno. Também propõe um modelo probabilístico adaptativo que opera no modelo de estimativa de AoA, incrementando sua performance. Seus resultados indicam que estes métodos superam a performance de modelos probabilísticos estáticos tradicionais, tanto em acurácia de localização quanto em estabilidade no valor obtido.

Neste trabalho, BniLam *et al.* [18] propõe um dispositivo de baixo custo capaz de estimar o AoA, de forma que seja viável sua utilização em dispositivos de IoT. O dispositivo consiste numa conversão de vários Rádio Definido por Software (*Software-Defined Radio*, SDR) individuais de baixo custo num único SDR com múltiplos canais de RF. Seus resultados experimentais indicam que o dispositivo é capaz de estimar valores de AoA de forma estável e acurada.

A proposta de BniLam *et al.* [19] neste trabalho consiste em um novo algoritmo para determinação de AoA chamado ANGLE (*ANGular Location Estimation*), baseado em modelos probabilísticos para a resposta do sinal recebido. Sua proposta ainda sugere duas versões do método, para o caso de amostragem única e de decomposição de subespaço, como utilizado no algoritmo MUSIC (*MUltiple Signal Classification*).

BniLam *et al.* [20] apresenta neste trabalho uma abordagem mais amigável para estimativa de AoA em redes LoRa. O sistema proposto, denominado LoRay (LoRa array) é composto por hardware e software preparados para fazer a estimativa de AoA em ambiente urbano, onde o sistema foi validado. O hardware utilizado foi descrito em um trabalho anterior [18]. Este sistema apresentou resultados estáveis e acurados para estimativa de AoA tanto nos casos Linha de visão (*Line of Sight*, LoS) e quanto nos Sem Linha de visão (*Non Line of Sight*, NLoS).

Em seu trabalho, Niculescu e Nath [21] propõe métodos para detecção de posição

e orientação em cada nó de uma rede *ad hoc*. A proposta parte de possíveis problemas relacionados a utilização de GPS em ambiente fechado

3 Metodologia

4 Resultados

5 Conclusão

Referências

- [1] ISRO, Indian Space Research Organisation, Departament of Space, *Sounding Rockets*. endereço: <https://www.isro.gov.in/soundingRockets.html>.
- [2] ESA, European Space Agency, *Sounding rockets*. endereço: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Research/Sounding-rockets.
- [3] M. Sabbatini e N. Sentse, “ESA User Guide to Low Gravity Platforms,” *Directorate of Human Spaceflight and Operations*, 2014. endereço: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Research/European-user_guide_to_low-gravity_platforms.
- [4] NASA, National Aeronautics and Space Administration, *About Sounding Rockets*. endereço: <https://www.nasa.gov/soundingrockets/overview/>.
- [5] ESRA, Experimental Sounding Rocket Association, *The Intercollegiate Rocket Engineering Competition*. endereço: <https://www.soundingrocket.org/what-is-irec.html>.
- [6] ESRA, Experimental Sounding Rocket Association, *Spaceport America Cup Intercollegiate Rocket Engineering Competition Rules & Requirements Document*. endereço: https://www.soundingrocket.org/uploads/9/0/6/4/9064598/sa_cup_irec_rules_and_requirements_document-2023_v1.3_20231001.pdf.
- [7] P. Guitarrara. “Coordenadas geográficas. ” endereço: <https://brasilecola.uol.com.br/geografia/coordenadas-geograficas.htm>.
- [8] C. Veness, *Calculate distance, bearing and more between Latitude/Longitude points*. endereço: <https://www.movable-type.co.uk/scripts/latlong.html>.
- [9] H. Fleming. “Coordenadas esféricas. ” endereço: <http://fma.if.usp.br/~fleming/diffeo/node4.html>.
- [10] D. Jennings, R. Drullinger, K. Evenson, C. Pollock e J. Wells, “The continuity of the meter: the redefinition of the meter and the speed of visible light,” *Journal of Research of the National Bureau of Standards*, v. 92, n. 1, p. 11, 1987.
- [11] A. Bensky, *Wireless positioning technologies and applications*. Artech House, 2016. endereço: https://scholar.rose-hulman.edu/cgi/viewcontent.cgi?article=1012&context=electrical_grad_theses.
- [12] V. Horst, “Localization and Angle-of-Arrival in Bluetooth Low Energy,” 2021. endereço: <https://www.ds.informatik.uni-kiel.de/en/teaching/bachelor-and-master-theses/completed-master-and-bachelor-theses/2021%20bachelor%20thesis%20Valentin%20Horst.pdf>.
- [13] C. A. Balanis, *Antenna Theory: Analysis and Design*. John Wiley & Sons, 2005.

- [14] H. Zeaiter, O. Baala, F. Spies e V. Thierry, “Performance Evaluation of the Angle of Arrival of LoRa Signals under Interference,” em *9th IEEE International Conference on Communications and Electronics (ICE 2022)*, IEEE, Nha Trang, Vietnam: IEEE, jul. de 2022. endereço: <https://ut3-toulouseinp.hal.science/hal-03693641>.
- [15] H. Zeaiter, F. Spies, O. Baala e T. Val, “Measuring accurate Angle of Arrival of weak LoRa signals for Indoor Positioning,” em *12th International Conference on Indoor Positioning and Indoor Navigation (IPIN 2022)*, Beijing, China, set. de 2022. DOI: 10.1109/IPIN54987.2022.9918114. endereço: <https://hal.science/hal-03932846>.
- [16] N. BniLam, J. Steckel e M. Weyn, “2D angle of arrival estimations and bandwidth recognition for broadband signals,” em *2017 11th European Conference on Antennas and Propagation (EUCAP)*, IEEE, 2017, pp. 2041–2045. endereço: https://www.researchgate.net/publication/317397561_2D_angle_of_arrival_estimations_and_bandwidth_recognition_for_broadband_signals.
- [17] N. BniLam, G. Ergeerts, D. Subotic, J. Steckel e M. Weyn, “Adaptive probabilistic model using angle of arrival estimation for IoT indoor localization,” em *2017 International conference on indoor positioning and indoor navigation (IPIN)*, IEEE, set. de 2017, pp. 1–7.
- [18] N. BniLam, D. Joosens, J. Steckel e M. Weyn, “Low cost AoA unit for IoT applications,” em *2019 13th European Conference on Antennas and Propagation (EuCAP)*, IEEE, 2019, pp. 1–5. endereço: https://www.researchgate.net/profile/Noori-Bnilam-2/publication/332141599_Low_Cost_AoA_Unit_for_IoT_Applications/links/5ca2ee8d299bf116956bf0e/Low-Cost-AoA-Unit-for-IoT-Applications.pdf.
- [19] N. BniLam, E. Tanghe, J. Steckel, W. Joseph e M. Weyn, “ANGLE: ANGular location estimation algorithms,” *IEEE access*, v. 8, pp. 14 620–14 629, 2020. endereço: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8959176>.
- [20] N. BniLam, T. Janssen, M. Aernouts, R. Berkvens e M. Weyn, “LoRa 2.4 GHz communication link and range,” *Sensors*, v. 20, n. 16, p. 4366, 2020.
- [21] D. Niculescu e B. Nath, “Ad hoc positioning system (APS) using AOA,” em *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, vol. 3, 2003, 1734–1743 vol.3. DOI: 10.1109/INFCOM.2003.1209196.

A Códigos desenvolvidos para simulação

A.1 Simulação de direcionamento GNSS

```
1 EARTH_RADIUS = 6371E3;
2
3           % Lat           Lon
4 coord_ufabc_SA = [-23.64450271006082 -46.52808340455618];
5 coord_ufabc_SBC = [-23.67751377362596 -46.563491717644425];
6
7 A = coord_ufabc_SA;
8 B = coord_ufabc_SBC;
9
10 theta_A = deg2rad(A(1));
11 theta_B = deg2rad(B(1));
12
13 phi_A = deg2rad(A(2));
14 phi_B = deg2rad(B(2));
15
16 delta_theta = theta_B - theta_A;
17 delta_phi = phi_B - phi_A;
18
19 X = cos(theta_B)*sin(delta_phi);
20 Y = cos(theta_A)*sin(theta_B) ...
21     - sin(theta_A)*cos(theta_B)*cos(delta_phi);
22 Z = (sin(delta_theta/2))^2 ...
23     + cos(theta_B) * cos(theta_A) * (sin(delta_phi/2))^2;
24
25 beta = atan2(X,Y) - pi/2;
26
27 if beta < -pi
28     beta = beta + 2*pi
29 end
30
31 d = EARTH_RADIUS * 2 * atan2(sqrt(Z), sqrt(1 - Z))
32
33 d_deg = rad2deg(beta)
```

A.2 Simulação de AoA

```
1 clear all;
2 close all;
3 clc;
4
5 if isoctave()
6     pkg load communications;
7 end %if
```

```

8
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10
11 DEBUG = false;
12
13 NOISE = true;
14
15 ATTENUATION = true;
16
17 CHANGE_PHI = false;
18 CHANGE_DISTANCE = false;
19
20 SAVE_GIF = false;
21
22 SNR = 1/1;
23 SNR_dB = 10*log10(SNR);
24
25 phase = 0;
26 % phase = 2*pi*rand()
27 R_upper = 50;
28 R_lower = 1;
29
30 range_step = 5;
31
32 DoA_range = 0:range_step:360;
33 % DoA_range = [30];
34 DoA = deg2rad(DoA_range);
35
36 limits = 2; % +- * Lambda
37
38 lambda = 2;
39 omega = pi;
40 d = lambda / 2;
41 T = 2 * pi / omega;
42
43 amp_0 = 1;
44
45 C = true;
46 S = true;
47
48 interval = limits*lambda;
49 resolution = 100;
50
51 % o = ones(1,resolution);
52 space = linspace(-interval,interval,resolution+1); % 1-by-100
53 [x, y] = meshgrid(space);
54 % x = space; % 1-by-100
55 % y = space'; % 1-by-100
56 t_0 = 0;
57
58 name = 'simul';

```

```

59
60 if CHANGE_DISTANCE
61     name = [name '_R_' num2str(R_upper) '~' num2str(R_lower) ];
62     % name = sprintf('%s_SNR_%d',name, SNR);
63 else
64     name = [name '_R_' num2str(R_upper)];
65 end %if
66 if NOISE
67     name = [name '_SNR_' num2str(SNR)];
68     % name = sprintf('%s_SNR_%d',name, SNR);
69 end %if
70 if ATTENUATION
71     name = [name '_ATT'];
72 end %if*
73
74 if SAVE_GIF
75     filename = ['Output/' name '.gif'];
76     % filename = sprintf('%s.gif',name, SNR);
77 end %if
78
79 if SAVE_GIF
80     % f = figure(1, 'Position', get(0, 'Screensize'), 'visible',
81         'off');
82     f = figure(1, 'name', name, 'visible', 'off', 'Position', [1
83         1 1000 500]);
84 else
85     f = figure('name', name);
86     % f = figure(1, 'Position', get(0, 'Screensize'));
87     % _ = input('Pressione Enter para continuar... ');
88 end %if
89
90 percent = 0;
91 ref_iteration = 1/length(DoA_range);
92 it = 0;
93 DelayTime = 15*ref_iteration;
94 r = R_upper+R_lower;
95
96 printf('%.2f%% -> %s\n', percent*100, name);
97
98 for DoA = [DoA_range DoA_range] % Duas voltas
99     it = it + 1;
100
101     if CHANGE_PHI
102         phase = phase + pi*ref_iteration;
103     end %if
104
105     if CHANGE_DISTANCE
106         r = r - R_upper/(2*length(DoA_range));
107     end %if
108
109     % disp('')

```

```

108     ang_W = deg2rad(DoA);
109
110     generate_fig(x, y, t_0, amp_0, ang_W, r, phase, lambda,
111                omega, S, C, ...
112                NOISE, SNR_dB, ATTENUATION, DEBUG, interval, resolution, T
113                , d);
114
115     drawnow;
116     if SAVE_GIF
117         frame = getframe(f);
118         im = frame2im(frame);
119
120         [imind, cm] = rgb2ind(im);
121         if it == 1
122             % Create GIF file
123             imwrite(imind, cm, filename, 'gif', 'DelayTime', DelayTime
124                 , 'Compression' , 'lzw');
125         else
126             % Add each new plot to GIF
127             imwrite(imind, cm, filename, 'gif', 'WriteMode', 'append', '
128                 DelayTime', DelayTime , 'Compression' , 'lzw');
129         end %if
130     else
131         % pause(1/30)
132         pause(0.0001)
133     end %if
134
135     percent = percent + ref_iteration/2;
136     printf('%.2f%% -> %s\n', percent*100, name);
137
138 end %for
139
140 if SAVE_GIF
141     printf('Check: %s\n', filename);
142 end %if
143
144 function generate_fig(x_w, y_w, t_w, amp_w, ang_w, r_w,
145                     phase_w, lambda_w, omega_w, S, C, NOISE, SNR_dB,
146                     ATTENUATION, DEBUG, interval, resolution, T, d)
147
148     % Calcular a figura de fundo para visualizacao em imagem
149     z = signal_r(x_w, y_w, t_w, amp_w, ang_w, r_w, phase_w,
150                 ...
151                 lambda_w, omega_w, S, C, NOISE, SNR_dB, ATTENUATION);
152
153     if ~DEBUG
154
155         t = linspace(0,T,resolution);
156
157         % Calculos de amostra I/Q para antena em (0,0)
158         C_00 = trapz(t,ref_cos(t, lambda_w, omega_w) ...
159                     .* signal_r(0, 0, t, amp_w, ang_w, r_w, phase_w, ...

```

```

14     lambda_w, omega_w, S, C, NOISE, SNR_dB, ATTENUATION));
15     S_00 = trapz(t, ref_sin(t, lambda_w, omega_w) ...
16     .* signal_r(0, 0, t, amp_w, ang_w, r_w, phase_w, ...
17     lambda_w, omega_w, S, C, NOISE, SNR_dB, ATTENUATION));
18
19     Z_00 = 2*(S_00 + 1i*C_00);
20
21     % Calculos de amostra I/Q para antena em (0,d)
22     C_0d = trapz(t, ref_cos(t, lambda_w, omega_w) ...
23     .* signal_r(0, d, t, amp_w, ang_w, r_w, phase_w, ...
24     lambda_w, omega_w, S, C, NOISE, SNR_dB, ATTENUATION));
25     S_0d = trapz(t, ref_sin(t, lambda_w, omega_w) ...
26     .* signal_r(0, d, t, amp_w, ang_w, r_w, phase_w, ...
27     lambda_w, omega_w, S, C, NOISE, SNR_dB, ATTENUATION));
28
29     Z_0d = 2*(S_0d + 1i*C_0d);
30
31     % Calculos de amostra I/Q para antena em (d, 0)
32     C_d0 = trapz(t, ref_cos(t, lambda_w, omega_w) ...
33     .* signal_r(d, 0, t, amp_w, ang_w, r_w, phase_w, ...
34     lambda_w, omega_w, S, C, NOISE, SNR_dB, ATTENUATION));
35     S_d0 = trapz(t, ref_sin(t, lambda_w, omega_w) ...
36     .* signal_r(d, 0, t, amp_w, ang_w, r_w, phase_w, ...
37     lambda_w, omega_w, S, C, NOISE, SNR_dB, ATTENUATION));
38
39     Z_d0 = 2*(S_d0 + 1i*C_d0);
40
41     % Calculo de defasagem entre antenas 0d e 00 (Eixo x)
42     Z_0d_x_00 = Z_0d * conj(Z_00);
43     delta_0d = angle(Z_0d_x_00);
44     % delta_0d -= ifelse(delta_0d>pi,2*pi,0);
45
46     % Calculo de defasagem entre antenas d0 e 00 (Eixo Y)
47     Z_d0_x_00 = Z_d0 * conj(Z_00);
48     delta_d0 = angle(Z_d0_x_00);
49     % delta_d0 -= ifelse(delta_d0>pi,2*pi,0);
50
51     % Parametros r
52     r_0d = delta_0d * lambda_w / (2 * pi);
53     r_d0 = delta_d0 * lambda_w / (2 * pi);
54
55     % Calculos de angulo de chegada
56     componente_x = -r_d0/d; % Conjunto do eixo X
57     componente_y = -r_0d/d; % Conjunto do eixo T
58
59     % Variaveis auxiliares
60     DoA_x = cos(ang_w);
61     DoA_y = sin(ang_w);
62
63     % Exibir valores
64     if false

```

```

65     printf('C_00 = %.2f\n',C_00);
66     printf('C_0d = %.2f\n',C_0d);
67     printf('C_d0 = %.2f\n',C_d0);
68     disp('')
69     printf('S_00 = %.2f\n',S_00);
70     printf('S_0d = %.2f\n',S_0d);
71     printf('S_d0 = %.2f\n',S_d0);
72     disp('')
73     printf('Z_00 = %.2f*\n',rad2deg(Z_00));
74     printf('Z_0d = %.2f*\n',rad2deg(Z_0d));
75     printf('Z_d0 = %.2f*\n',rad2deg(Z_d0));
76     disp('')
77     printf('delta_0d = %.2f*\n',rad2deg(delta_0d));
78     printf('delta_d0 = %.2f*\n',rad2deg(delta_d0));
79     disp('')
80     end %if
81
82 end %if
83
84 % set(0, 'CurrentFigure', fig3d)
85
86 set(0, 'defaultlinelinerwidth', 1);
87 set(0, 'defaultlinemarkersize', 5);
88 set(0, 'defaultlinemarkerfacecolor', 'auto');
89
90 clf;
91 subplot(1,2,2, 'align');
92 surf(x_w,y_w,z);
93 view(-45, 45)
94
95 colormap('gray');
96 hold on;
97     plot3([0 0], [0 0], [-lambda_w lambda_w], '-pg', '
98         linewidth', 1);
99     % plot3(0, 0, lambda_w, 'pg', 'linewidth', 1);
100    plot3([0 0], [d d], [-lambda_w lambda_w], '-pb', '
101        linewidth', 1);
102    % plot3(0, d, lambda_w, 'pb', 'linewidth', 1);
103    plot3([d d], [0 0], [-lambda_w lambda_w], '-pr', '
104        linewidth', 1);
105    % plot3(d, 0, lambda_w, 'pr', 'linewidth', 1);
106    shading interp;
107    axis([-interval interval -interval interval -interval
108         interval],'square')
109    caxis([-lambda_w lambda_w]);
110 hold off;
111
112 if ~DEBUG
113
114     % set(0, 'CurrentFigure', fig2d)
115     % clf;

```

```

112 subplot(1,2,1, 'align');
113
114 interval_2d = 1.25*lambda_w;
115 axis([-interval_2d interval_2d -interval_2d interval_2d], '
    square')
116
117 % Antenas
118 % A00 = plot(0, 0, 'pg');
119 A00 = compass(0, 0, 'pg');
120 hold on;
121 % if isoctave()
122 %     set( gca, 'rtick', [ 0 : 0.25*lambda_w : interval_2d ]
123 % );
124 %     set( gca, 'ttick', [ 0 : 15 : 359 ] );
125 % else
126 %     rtick([ 0 : 0.25*lambda_w : interval_2d ]);
127 %     ttick([ 0 : 15 : 359 ]);
128 % end %if
129 A0d = plot(0, d/lambda_w, 'pb');
130 Ad0 = plot(d/lambda_w, 0, 'pr');
131
132 grid on;
133 grid minor;
134 % colormap('jet');
135 % colormap('gray');
136 % colormap('hot');
137 % colormap('cool');
138 % xlabel('x');
139 % ylabel('y');
140 % colorbar();
141
142 % Fundo de referencia
143 % imagesc(x_w,y_w,z);
144
145 % Fase por antena
146 plot(Z_0d/abs(Z_0d)*.5, 'ob');
147 plot(Z_d0/abs(Z_d0)*.5, 'or');
148 plot(Z_00/abs(Z_00)*.5, 'og', 'markerfacecolor', 'none');
149
150 % Defasagem entre antenas
151 plot(Z_0d_x_00/(abs(Z_0d_x_00))*0.25, 'dg', 'markerfacecolor
    ', 'b');
152 plot(Z_d0_x_00/(abs(Z_d0_x_00))*0.25, 'dg', 'markerfacecolor
    ', 'r');
153
154 % Angulo de chegada calculado
155 plot(componente_x*0.75, componente_y*0.75, 'vm');
156 % plot([0 lambda_w*componente_x], [0 lambda_w*componente_y
157 % ],':y', 'linewidth', 1, 'markersize', 5)
158
159 % Angulo de chegada real

```

```

158     plot(DoA_x*0.75, DoA_y*0.75, '^c', 'markerfacecolor', 'none
159         ');
160     % plot([0 DoA_x], [0 DoA_y], '--c', 'linewidth', 1, '
161         markersize', 5)
162     % h = legend('show');
163     % legend (h, 'location', 'east');
164     % set (h, 'fontsize', 20);
165     hold off;
166
167     end %if
168
169 end %function

```

```

1 function res = signal_r(x_w, y_w, t_w, amp_w, ang_w, r_w,
2     phase_w, lambda_w, omega_w, S, C, NOISE, SNR_dB,
3     ATTENUATION)
4 % Funcao de sinal senoidal
5 % x_w = coordenada x associada ao ponto da antena
6 % y_w = coordenada y associada ao ponto da antena
7 % t_w = tempo t associado ao instante de afericao do sinal
8 % ang_w = angulo theta de chegada do sinal em relacao ao
9     sistema de antenas
10 % phase_w = defasagem phi do sinal
11 % lambda_w = comprimento de onda
12 % omega_w = frequencia angular
13 % S = utilizacao de funcao Seno
14 % C = Utilizacao de funcao Cosseno
15 % NOISE = Se o sinal contara com ruido
16 % SNR_dB = Relacao sinal-ruido
17 % ATTENUATION = Se o sinal contara com atenuacao por distancia
18
19 res = 0;
20 if S
21     res = res + sin( argument_r(x_w, y_w, t_w, ang_w, r_w,
22         lambda_w, omega_w) + phase_w);
23 end %if
24 if C
25     res = res + cos( argument_r(x_w, y_w, t_w, ang_w, r_w,
26         lambda_w, omega_w) + phase_w);
27 end %if
28 if S && C
29     res = res / sqrt(2);
30 end %if
31
32 if ATTENUATION
33     %%% Variaveis auxiliares
34     G_t = 1;
35     G_r = 1;
36     R_t = 1;
37     R_r = 1;

```



```

33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34
35 P_t = (amp_w^2)/R_t;
36 P_r = P_t * G_t * G_r * (lambda_w / (4 * pi * r_w));
37
38 amp_r = sqrt(P_r * R_t);
39 res = res * amp_r;
40 else
41     res = res * amp_w;
42 end %if
43
44 if NOISE
45     res = awgn(res, SNR_dB, 'measured');
46     % res = awgn(res, SNR, 'linear');
47     % res += awgn(1, SNR_dB, 1, 0);
48     % res += rand()/SNR_dB;
49 end %if
50 % res = res * 2;
51 end %function

1 function c = ref_cos(t_w, lambda_w, omega_w)
2     % funcao auxiliar de sinal cosseno
3     c = cos(argument_r(0, 0, t_w, 0, 0, lambda_w, omega_w));
4 end %function

1 function s = ref_sin(t_w, lambda_w, omega_w)
2     % funcao auxiliar de sinal seno
3     s = sin(argument_r(0, 0, t_w, 0, 0, lambda_w, omega_w));
4 end %function

1 function res = argument_r(x_w, y_w, t_w, ang_w, r_w, lambda_w,
2     omega_w)
3     % Argumento da funcao senoidal
4     % x_w = coordenada x associada ao ponto da antena
5     % y_w = coordenada y associada ao ponto da antena
6     % t_w = tempo t associado ao instante de afericao do sinal
7     % ang_w = angulo theta de chegada do sinal em relacao ao
8     % sistema de antenas
9     % lambda_w = comprimento de onda
10    % omega_w = frequencia angular
11
12    r = r_w * lambda_w;
13
14    x_0 = r * cos(ang_w);
15    y_0 = r * sin(ang_w);
16    % y_0 = r * 10;
17
18    res = (2*pi/lambda_w) * ( sqrt((y_w-y_0).^2 + (x_w-x_0).^2)
19        ) + omega_w*t_w;
20 end %function

1 function r = isoctave () % Confere se esta executando no
2     octave ou nao

```

```
2 | persistent x;  
3 | if (isempty (x))  
4 |     x = exist ('OCTAVE_VERSION', 'builtin');  
5 | end  
6 | r = x;  
7 | end
```