

Henrique Ferreira dos Santos

Localização de Fontes Sonoras por TDOA e  
Função de Coerência Semblance

Orientador: Prof. Dr. Kenji Nose-Filho

Santo André - SP

2019

Henrique Ferreira dos Santos

# Localização de Fontes Sonoras por TDOA e Função de Coerência Semblance

Monografia apresentada ao programa de Graduação em Engenharia de Informação da Universidade Federal do ABC (UFABC), como requisito parcial à obtenção do título de Bacharel em Engenharia de Informação.

Orientador: Prof. Dr. Kenji Nose-Filho

Santo André - SP

2019

## Agradecimentos

Inicialmente devo agradecer a uma série de pessoas que direta ou indiretamente estiveram envolvidas para a execução deste trabalho. Devo agradecer todo o apoio dos meus pais, irmão, familiares e amigos próximos pela companhia e paciência. Devo um agradecimento especial ao meu pai, Armando Ferreira dos Santos Júnior, pela ajuda na construção dos suportes para o ambiente de teste. Também agradeço particularmente aos meus colegas de curso Rodrigo Hideaki Murakami, André Seiti Murasaki, Jefferson da Silva Kumamoto, Hugo Tamotsu Takahashi, Caio Melo, Ana Paula Pagotti, Isabella Pirani e Alexandre Miccheleti Lucena.

Agradeço enormemente ao professor Stilante Koch Manfrin pela ajuda e prontidão no desenvolvimento de uma primeira ideia de trabalho de graduação, na qual tive a oportunidade de aprender sobre a teoria geral de radares eletromagnéticos, casadores híbridos e teoria de antenas, e, em especial, o funcionamento de Radares Monopulso de Rastreamento. Agradeço ao professor Marcelo Perotoni pela continuação desta ideia ao tocante a construção de antenas. Apesar desse trabalho não ter sido finalizado, foi grande fonte de aprendizado e inspiração.

Devo agradecer ao professor Ricardo Suyama por todas as considerações e por ter me engajado dentro da representação discente do curso. Aos professores que foram coordenadores do curso de Engenharia de Informação, Murilo Bellezoni Loiola e Amaury Kruel Budri, pela presteza em me atender sempre que levava as demandas dos alunos para discussão. Agradeço a todo o corpo docente da UFABC, em especial aos professores da Engenharia de Informação com quem tive a oportunidade de ter aulas. Lembro com gratidão de André Kazuo Takahata, Claudio José Bordin Júnior, Denise Consonni, Helói Francisco Gentil Genari, Irineu Antunes Junior, Ivan Roberto Santana Casella, João Henrique Ranhel Ribeiro, João Henrique Kleinschmidt, Katia Franklin Albertin Torres, Luiz Henrique Bonani do Nascimento, Mario Alexandre Gazziro, Mario Minami e em especial, de Marco Aurélio Cazarotto Gomes, um professor com grande espírito de altruísmo.

Agradeço ao meu grande amigo Tito Caco Curimbaba Spadini pelas profundas discussões, por ter me cedido o Raspberry Pi aqui utilizado, pelo curto período em que pude aprender com o projeto desenvolvido para a competição de veículos autônomos RoboCar Race e pela colaboração com trabalhos de inteligência artificial aplicada a redução de ruído.

Aos meus colegas de equipe da competição Signal Processing Cup do IEEE, Alex En-

rique Crispim, Guilherme Barreto e Kaleb Lucas da Silva Alves, que trouxeram inspiração para o tema aqui tratado e renderam publicações importantes, e especialmente a Guilherme Seidy Imai Aldeia, pela ajuda com a adaptação do algoritmo de Semblance.

Agradeço a todo pessoal da IBM Research Brasil onde tive o prazer de estagiar por mais de um ano, trabalhando com aplicação de inteligência artificial para reconhecimento de fala, processamento de linguagem natural e construção de chatbots habilitados por texto e voz no contexto de interfaces humano-máquina. Agradeço especialmente ao meu supervisor Julio Nogima com quem pude aprender diversos aspectos do trabalho em pesquisa na indústria. Ao meu gerente Claudio S. Pinhanez, pelo apoio e conhecimento transmitido, ao pesquisador Mauro C. Pichiliani com quem discuti diferentes aspectos da localização de fontes sonoras, eletrônica maker e que me cedeu o ReSpeaker utilizado neste trabalho, e ao meu grande amigo Marcelo Carpinette Grave pelo companheirismo, aprendizado e pelas infindáveis discussões acerca da execução e aplicação dos conceitos que este trabalho aborda.

Por fim, devo agradecer ao meu orientador, professor Kenji Nose-Filho, pela paciência e por todo conhecimento e oportunidades gerados nos meses em que trabalhamos juntos. Sua disposição ímpar em auxiliar foi de grande importância para a conclusão deste trabalho.

---

## Resumo

A Localização de Fontes Sonoras (do inglês, *Sound Source Localization* - SSL) constitui uma classe de problemas de Localização Acústica, cujo objetivo é estimar a posição de uma fonte sonora a partir do sinal acústico captado, comumente através de um Arranjo de Microfones. As aplicações dos métodos desenvolvidos para a solução destes problemas são bastante vastas, como em sonares, sistemas de navegação e robôs interagentes por som. Além disso, tais métodos são facilmente intercambiáveis para sinais eletromagnéticos, encontrando mais aplicações, como em antenas inteligentes, sistemas de GPS e radares. Neste trabalho desenvolvemos uma revisão dos conceitos de acústica e processamento de áudio com foco no desenvolvimento de técnicas de SSL. Apresentamos os fundamentos teóricos por trás de diferentes algoritmos de Determinação de Direção de Chegada (do inglês, *Direction of Arrival* - DOA), especialmente aqueles baseados no tempo de atraso da chegada (do inglês, *Time Difference of Arrival* - TDOA), e desenvolvemos a aplicação da Função de Coerência Semblance, originalmente utilizada em sísmica de reflexão, para o contexto de SSL. O algoritmo foi programado na Linguagem Python e embarcado em um computador de placa única Raspberry Pi modelo B, utilizando um Arranjo de Microfones Respeaker de 6 canais de captação. A estimação da direção de chegada foi utilizada como variável de atuação de um sistema de servo-motores operados através de um microcontrolador ATmega 328P em uma placa Arduino UNO, com o intuito de se apontar um laser na direção estimada. Foi testada a condição de utilização do algoritmo para distâncias acima da Distância de Fraunhofer, assim como o desempenho para a estimação da direção de chegada para diferentes direções reais. A solução implementada demonstrou-se eficaz na detecção da direção da fonte, considerando-se o limite aceitável do erro de  $10^\circ$ . Entretanto, a aplicação não funciona em tempo real, como os testes de tempo feito sugerem. Por fim, apresentamos um resultado para a detecção de múltiplas fontes, mostrando que o algoritmo também permite essa funcionalidade.

**Palavras chave:** Processamento de Sinais de Áudio, Localização de Fontes Sonoras, Estimação de Direção de Chegada, Algoritmos de DOA.

---

## Abstract

The Sound Source Localization (SSL) problem is a particular class of Acoustic Localization problems, that aims to estimate the source position/location of an acoustic signal. This signal is often obtained using a microphone array. The applications of the methods developed to solve these problems are vast, such as sonars and sound interagents robots. Besides that, these methods are easily applicable on electromagnetic signals, finding applications in intelligent antennas and radars. In this work, we review the concepts of acoustic and audio processing to develop SSL techniques. We show theoretical foundations behind different Direction of Arrival (DOA) algorithms, specially those based on Time Difference of Arrival (TDOA). To estimate the time difference, we used the Semblance Coherence Function, originally used in seismic reflection, for the SSL context. This method was implemented in Python and embedded in a Raspberry Pi 3 model B single-boarder computer, using a 6 channels Respeaker Microphone Array. The estimated direction was used as input for an ATmega 328P microcontroller to control a servomotor system to point out the sound source with a laserpointer. The algorithm was tested for different distances above the Fraunhofer Distance as well as for different direction of arrival for the source location. The implemented solution shown effective in detecting the source direction considering the acceptable error limit of  $10^\circ$ . However, as time tests suggest, the application does not work in real time. Finally, we present a result for the detection of multiple sources, showing that the algorithm also allows this functionality.

**Keywords:** Audio Signal Processing, Sound Source Localization, Direction of Arrival Estimation, DOA Algorithms.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>13</b>
<b>2</b>	<b>Sinais Acústicos</b>	<b>15</b>
2.1	Ondas Acústicas . . . . .	15
2.2	Potência Acústica . . . . .	19
2.3	Elementos de Captação Acústica . . . . .	21
2.4	Sinal de Áudio . . . . .	23
2.5	Processamento Digital de Áudio . . . . .	26
2.6	Processamento de Áudio com Arranjo de Microfones . . . . .	27
<b>3</b>	<b>Técnicas de DOA</b>	<b>31</b>
3.1	TDOA . . . . .	31
3.1.1	Técnicas baseadas em Medidas de Coerência . . . . .	33
3.1.1.1	Correlação Cruzada Generalizada . . . . .	34
3.1.1.2	Semblance . . . . .	34
3.1.2	Técnicas baseadas em Decomposição por Autovalor . . . . .	37
3.1.2.1	MUSIC . . . . .	37
3.2	Beamforming . . . . .	38
<b>4</b>	<b>Metodologia</b>	<b>41</b>
4.1	Objetivos . . . . .	41
4.2	Hardware Utilizado . . . . .	41
4.3	Ambiente de Desenvolvimento . . . . .	42
4.4	Avalização de Desempenho . . . . .	42
4.4.1	Testes . . . . .	43
4.4.2	Elementos suplementares . . . . .	44
<b>5</b>	<b>Resultados e Discussão</b>	<b>45</b>
5.1	Hardware . . . . .	45
5.2	Software . . . . .	47
5.3	Desempenho . . . . .	48
5.3.1	Demonstração de Funcionamento . . . . .	48
5.3.2	Testes de Distância de Fraunhofer . . . . .	49
5.3.3	Testes de Azimute . . . . .	49

## SUMÁRIO

---

5.3.4	Testes de Elevação . . . . .	50
5.3.5	Desempenho de Azimute e Elevação . . . . .	51
5.3.6	Desempenho de tempo . . . . .	52
5.3.7	Teste com Múltiplas Fontes . . . . .	53
<b>6</b>	<b>Considerações Finais</b>	<b>55</b>
6.1	Perspectivas Futuras . . . . .	55
6.2	Conclusão . . . . .	56

# Lista de Figuras

2.1	Visualização de uma onda acústica plana em um gás movendo-se para a direita com velocidade $c$ . (a) Regiões de compressão (pontos densos) e de rarefação (pontos espaçados). (b) Regiões de pressão constante (frente de onda), linhas cheias representam regiões de máxima pressão $+p_0$ e linhas tracejadas regiões de mínima pressão $-p_0$ . (c) Visualização da onda de pressão com solução senoidal em relação ao espaço $x$ para dois instantes do tempo. (adaptado de [12]). . . . .	17
2.2	Visualização de uma onda acústica esférica se propagando a partir de uma fonte puntual (monopolo acústico). (a) Regiões de pressão constante (frente de onda), linhas cheias representam regiões de máxima pressão $+p_0$ e linhas tracejadas regiões de mínima pressão $-p_0$ (adaptado de [12]). (b) Visualização bidimensional das regiões de compressão determinada pelos pontos densos e de rarefação determinada pelos pontos espaçados (adaptado de [13]). (c) Visualização tridimensional (adaptado de [14]). . . . .	18
2.3	Representação da propagação da frente de onda acústica esférica a partir de uma fonte puntual, onde $A$ representa elementos de área superficial pelos quais a frente de onda passa (adaptado de [21]). . . . .	20
2.4	Exemplo de diretividade de microfones. (a) Ominidirecional. (b) Cardióide. (c) Hipercardióide. (d) Bidirecional. (Adaptado de [26]). . . . .	23
2.5	Exemplo de sinal de áudio de fala digitalizado. (a) Sinal $s(k)$ mapeado para o tempo $t$ da aquisição. (b) Espectro de amplitude do sinal $ A(\omega) $ . (c) Espectrograma do sinal. . . . .	25
2.6	Esquema de um sistema de captação de áudio por arranjo de microfones atuando dentro de um cenário hipotético com diferentes fontes de sinais (adaptado de [30]). . . . .	28
2.7	Modelo de raio para fonte sonora única com captação por arranjo de microfones. (a) Para campo próximo. (b) Para campo distante. Adaptado de [30]. . . . .	28
2.8	Distância de Fraunhofer em função da frequência sonora para três exemplos de alto-falantes. . . . .	29
3.1	Ilustração da estimação de DOA em um problema bidimensional com dois microfones, com condição de campo distante. (Adaptado de [30]). . . . .	31

3.2	Ilustração da estimação de DOA em um problema bidimensional com $M$ microfones, na condição de campo próximo. (Adaptado de [35]). . . . .	33
3.3	Fluxo de processamento para o DOA-GCC (extraído de [42]). . . . .	34
3.4	Visualização do resultado do método de Semblance. (a) Superfície 3D representando o resultado da função de Semblance. Os picos correspondem a alta correlação. (b) Vista 2D, curvas de nível de (a), denominada Painel de Semblance. . . . .	35
3.5	Exemplo de alinhamento dos sinais. (a) Sinal de 8 canais original representado em amplitude e tempo. (b) Sinal original com representação por canal. (c) Sinal corrigido para o pico de Semblance. (d) Sinal corrigido com representação por canal. (Adaptado de [43]) . . . . .	36
3.6	Representação do feixe de radiação (diretividade). (a) De uma antena (adaptado de [33]). (b) De um arranjo de microfones (adaptado de [34]). . .	38
3.7	Representação polar da radiação acústica a partir de diferentes fontes, onde $d$ é a distância entre os elementos radiadores e $D$ é a maior dimensão da fonte. (a) Monopolo. (b) Dipolo. (c) Quadripolo lateral. (d) Quadripolo longitudinal. (Adaptado de [20]). . . . .	39
3.8	Mapa Acústico. (a) Diagrama de radiação NPS em coordenadas polares (adaptado de [35]). (b) Diagrama de NPS em coordenadas retangulares sobre um mapa de vista área de uma região metropolitana (adaptado de [37]).	39
3.9	Esquema representativo do método de Beamforming (extraído de [35]). . . .	40
4.1	Dispositivos comerciais utilizadas. (a) Raspberry Pi 3, Model B (adaptado de [46]). (b) Arranjo circular de 6 microfones ReSpeaker (adaptado de [47]).	41
4.2	Exemplo de cenário de teste. . . . .	42
4.3	Elementos suplementares para os testes de avaliação. (a) Gabarito de azimute e elevação. (b) Suporte para alto-falante. (c) Sistema amplificador para alto-falante com entrada estéreo tipo P2. . . . .	44
5.1	Hardware do sistema de DOA desenvolvido. (a) ReSpeaker 6-Mic Circular. (b) Arduino Uno. (c) Servomotores e laser. (d) Detalhe do Raspberry Pi, localizado abaixo do ReSpeaker. . . . .	45
5.2	Erro angular para o posicionamento do laser. . . . .	46
5.3	Esquema do algoritmo de Semblance implementado. . . . .	47
5.4	Exemplo de funcionamento do algoritmo de Semblance. . . . .	48
5.5	Teste da condição de Franhoufer. . . . .	49
5.6	Acurácia no algoritmo para a variação de azimute. . . . .	50
5.7	Acurácia no algoritmo para a variação de elevação. . . . .	51
5.8	Erros médios para azimute e elevação. . . . .	51
5.9	Tempo de execução do Semblance para diferentes ângulos de teste. . . . .	52
5.10	Tempo de execução médio do algoritmo de Semblance. . . . .	52
5.11	Painel de Semblance para a localização de múltiplas fontes. . . . .	53

# Lista de Tabelas

2.1	Exemplos de nível sonoro SPI em diferentes contextos. . . . .	21
2.2	Exemplos de tecnologias de microfone. . . . .	22



# Capítulo 1

## Introdução

Um sinal pode ser visto como uma representação matemática de uma variável, que pode ser física ou não. O poder desta abstração pode ser entendido ao se observar sua aplicação em diferentes áreas do conhecimento, expressando-se sobretudo no funcionamento de todos os aparelhos eletrônicos. Este conceito é fundamental para a construção de sistemas de comunicação complexos (telefone, celular, rádio, televisão e internet) sejam estes estabelecidos de maneira cabeada ou sem fio, sistemas computacionais compactos (computadores desktop, notebooks, GPUs e servidores), equipamentos biomédicos (eletrocardiógrafo, ressonância nuclear magnética, ultrassom, analisador hematológico) e sistemas de atuação e controle (robôs industriais, impressoras de tinta e 3D, veículos autônomos terrestres, aquáticos e aéreos). Atualmente estes mesmos conceitos também estão sendo empregados em biologia [1, 2], economia [3] e neurociência [2, 4].

Mais precisamente, um sinal é uma variável (ou conjunto de variáveis) que varia ao longo do tempo ou do espaço e que carrega informação [5]. Uma imagem monocromática, por exemplo, pode ser entendida como um sinal cuja variável é a luminosidade ao longo do espaço; o nível de uma boia dentro de um tanque de água pode ser entendido como um sinal dado pela variável altura que varia somente com o tempo; já a comunicação por celular pode ser entendida através da variação do campo eletromagnético ao longo do tempo e do espaço nas imediações que ligam o aparelho à torre de transmissão. Fica evidente que é sobretudo na variação da propriedade medida que a informação está contida, caso contrário, teríamos uma imagem toda da mesma cor, sem contornos definidos, um tanque cujo volume de água nunca se altera e que, portanto, não necessita de sensores, ou um campo eletromagnético constante entre o transmissor e o receptor, denunciando que nada foi comunicado entre os dois.

Dentro do contexto de processamento de sinais, um problema de grande interesse é aquele em que deseja-se saber a posição da fonte de um sinal após sua recepção. Neste caso, estamos tratando de um sistema não cabeado, como por exemplo a transmissão e/ou recepção de ondas eletromagnéticas de estações de rádio, torres de celular, pontos de acesso sem fio e radares passivos. Este mesmo problema é válido para ondas acústicas, onde o sinal de interesse é um sinal sonoro, dentro ou não do espectro audível humano. Diferentemente do primeiro caso, onde são empregadas antenas e arranjos de antenas, para a captação de sinais sonoros são usados microfones ou arranjos de microfones que atuam

como transdutores mecânicos, transformando a vibração mecânica em um sinal elétrico.

A Localização de Fontes Sonoras (do inglês, *Sound Source Localization* - SSL) é um problema de grande interesse em engenharia, tendo sido empregado em diferentes contextos, sobretudo, militares. Atualmente com a automatização e a autonomização de veículos, surgem novos cenários para a aplicação destes conceitos, como em drones [6] e carros autônomos [7]. Não se limitando ao objetivo da localização, as técnicas empregadas tem um leque de aplicações ainda maior, como na sísmica de reflexão aplicada à prospecção geofísica [8, 9], e na construção de radares sonoros (sonares).

Uma parte da Localização de Fontes Sonoras é a determinação da Direção de Chegada (do inglês, *Direction of Arrival* - DOA). Neste caso, o objetivo é estimar os ângulos de elevação e azimute com que um sinal é recebido em relação a um ponto de referência. A etapa de DOA é necessária mas não suficiente para se solucionar um problema de SSL de propósito geral. Todavia, para alguns contextos de aplicação, a determinação da distância da fonte não é importante, de forma que a DOA é suficiente para estimar a localização da fonte.

Neste trabalho aprofundamos conceitos de acústica e processamento de sinais de áudio, com especial interesse na estimação de DOA dentro do contexto de SSL. Para isso descrevemos as principais técnicas para a determinação dos ângulos de chegada de sinais acústicos e construímos um dispositivo prático utilizando um arranjo de microfones e sistemas eletrônicos embarcáveis.

No Capítulo 2, fazemos uma breve revisão dos conceitos de acústica. No Capítulo 3 descrevemos as principais técnicas de estimação de direção de chegada, dando ênfase a técnica de TDOA por medida de coerência, aqui denominada *Semblance*. No Capítulo 4 descrevemos a metodologia empregada neste trabalho, assim como softwares e dispositivos utilizados. Os resultados obtidos são apresentados no Capítulo 5. Por fim, a conclusão é apresentada no Capítulo 6, conjuntamente com uma lista de possíveis aperfeiçoamentos e trabalhos futuros.

## Capítulo 2

# Sinais Acústicos

### 2.1 Ondas Acústicas

São denominadas ondas acústicas, ondas de pressão em um meio material, seja ele sólido, líquido ou gasoso. Para entender o que isso significa, inicialmente precisamos nos familiarizar com o termo onda, que se aplica a uma vasta gama de fenômenos físicos como, por exemplo, as ondas mecânicas, as ondas eletromagnéticas, as ondas de calor e as ondas gravitacionais. Iremos fazer uma breve revisão do arcabouço matemático por traz da representação de ondas.

Em três dimensões, uma onda pode ser descrita por:

$$\frac{1}{c^2} \frac{\partial^2}{\partial t^2} u(\vec{r}, t) = \nabla^2 u(\vec{r}, t) \quad (2.1)$$

onde  $c$  é a velocidade de propagação desta onda,  $\vec{r} = x\hat{i} + y\hat{j} + z\hat{k}$  é o vetor posição,  $u$  é a propriedade física que oscila ao longo do tempo  $t$  e através do espaço (codificado pelo operador Laplaciano  $\nabla^2$ ). Em se tratando de uma onda em uma corda, por exemplo, a Eq. 2.1 pode ser reduzida para:

$$\frac{1}{c^2} \frac{\partial^2}{\partial t^2} y(x, t) = \frac{\partial^2}{\partial x^2} y(x, t) \quad (2.2)$$

Podemos entender a Eq. 2.2 a partir da visualização de uma corda esticada ao longo do eixo  $x$  com  $y = 0$ , com a condição de contorno de ter uma das extremidades presa em um anteparo. Assumindo como condição inicial que a corda estava parada, imaginemos que em um determinado momento, uma fonte impulsiva (como a mão de uma pessoa) faz a extremidade livre da corda vibrar, isto é, gera um impulso sobre a mesma. Observaremos então o impulso a partir da elevação de um determinado elemento da corda ( $y \neq 0$ ). Isso significa que este elemento, posicionado a  $x$  da origem do sistema de coordenadas, em um determinado instante de tempo  $t$ , tem sua posição em  $y$  alterada de acordo com uma expressão  $y(x, t)$ , que é uma solução da Eq. 2.2:

$$y(x, t) = Ae^{i(kx \pm \omega t)} \quad (2.3)$$

Nesta Equação (2.3),  $A$  é a amplitude máxima da onda,  $k$  é o número de onda que pode

ser determinado através do comprimento de onda  $\lambda$  (através de  $k = \frac{2\pi}{\lambda}$ ) e  $\omega$  é a frequência angular da onda.

Se continuarmos olhando para o mesmo elemento da corda (mesma posição em  $x$ ) veremos sua posição em  $y$  aumentar e, em seguida, diminuir, sinalizando que o impulso passou por aquela região. Da mesma forma, se observarmos a corda como um todo, veremos que o impulso percorre ao longo de  $x$ , conforme a elevação em  $y$  se propaga da ponta livre impulsionada até a ponta presa no anteparo. Este exemplo descreve uma onda transversal, na qual a dimensão de variação ( $y$ ) é ortogonal a direção de propagação da onda ( $x$ ). Existem ainda as ondas longitudinais, nas quais a direção de propagação é a mesma da variação do parâmetro observado. Em meios sólidos, as ondas transversais podem ser entendidas como de cisalhamento e as longitudinais como ondas de pressão.

As ondas também podem ser classificadas em dispersivas e não-dispersivas. As não-dispersivas possuem a propriedade de se propagarem com a mesma velocidade, independente do comprimento de onda e da frequência que possuam. Já as dispersivas, possuem uma velocidade de propagação que depende do comprimento de onda e da frequência. [10]

A velocidade de propagação de uma onda é dada pela relação de dispersão:

$$c = \frac{\partial\omega}{\partial k} \quad (2.4)$$

Para ondas dispersivas temos que a frequência angular  $\omega = 2\pi f$  é uma função do número de onda  $\omega = \omega(\vec{k})$ . Para tratar esse tipo de onda devemos utilizar o conceito de velocidade de grupo, que carrega as informações sobre o envelope que define a onda. Este envelope pode ser entendido como o contorno da sobreposição de diferentes ondas que viajam com diferentes velocidades (ditas velocidades de fase). Já a velocidade de uma onda não-dispersiva possui uma relação linear entre  $\omega(\vec{k}) = ck$ , de forma que a velocidade de fase e de grupo são a mesma, podendo ser expressa como:

$$c = f\lambda = \frac{\omega}{k} \quad (2.5)$$

Ondas acústicas são ondas mecânicas, como a descrita pelo exemplo da corda. Ondas mecânicas devem se propagar através de um meio material, seja ele líquido, sólido ou gasoso. Evidentemente, cada tipo e estado da matéria determinam diferentes restrições das ondas que podem se formar. Líquidos e gases não transmitem ondas de cisalhamento, por exemplo. Em se tratando do som que ouvimos, a propagação ocorre através de ondas de pressão nos gases que compõem a atmosfera.

Assumindo um fluido quiescente<sup>1</sup>, as equações que regem a mecânica dos fluidos (transferência de massa e continuidade) podem ser elaboradas de maneira a renderem como resultados uma expressão análoga à Eq. 2.1 [11]. Neste caso, o parâmetro  $u$  pode ser substituído pela densidade  $\rho$  do fluido ou pela pressão  $p$ . Ambas as interpretações são intercambiáveis para o entendimento das ondas que trataremos aqui, porém, expressaremos isso em termos de pressão, como é comumente feito em acústica. Assim, em três e em uma dimensão temos que:

---

<sup>1</sup>Fluido uniforme em repouso [19].

$$\frac{1}{c^2} \frac{\partial^2}{\partial t^2} \vec{p} = \nabla^2 \vec{p} \quad (2.6)$$

$$\frac{1}{c^2} \frac{\partial^2}{\partial t^2} \vec{p}(x, t) = \frac{\partial}{\partial x} \vec{p}(x, t) \quad (2.7)$$

A solução geral desta equação é do tipo  $p_D(x - ct) + p_E(x + ct)$ , onde o subscrito  $D$  representa a onda que se propaga para a direita e o subscrito  $E$ , a que se propaga para a esquerda [12]. Observe que a função de pressão descrita aqui é a mesma do caso da corda (Eq. 2.3), onde o operador  $\pm$  codifica a informação de direção de propagação da onda. Na Figura 2.1 é apresentada a interpretação física deste equacionamento. Nela podemos ver três representações alinhadas da mesma onda de pressão em um gás ideal. Em (a) observamos como as partículas que compõem o gás tornam-se mais densas ou mais rarefeitas conforme a onda se propaga, revelando uma frente de onda com pressão constante (b). Em (c) temos a representação matemática de uma solução senoidal similar a descrita pela Eq. 2.3.

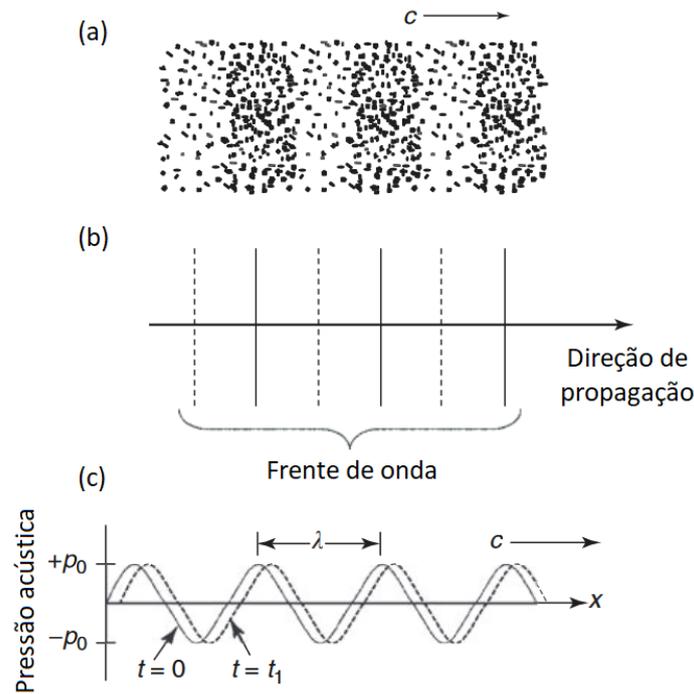


Figura 2.1: Visualização de uma onda acústica plana em um gás movendo-se para a direita com velocidade  $c$ . (a) Regiões de compressão (pontos densos) e de rarefação (pontos espaçados). (b) Regiões de pressão constante (frente de onda), linhas cheias representam regiões de máxima pressão  $+p_0$  e linhas tracejadas regiões de mínima pressão  $-p_0$ . (c) Visualização da onda de pressão com solução senoidal em relação ao espaço  $x$  para dois instantes do tempo. (adaptado de [12]).

Em três dimensões, para uma fonte simétrica (como uma explosão), as ondas acústicas apresentam simetria esférica, podendo ser escritas como [12]:

$$\frac{1}{c^2} \frac{\partial^2}{\partial t^2} (rp) = \frac{\partial}{\partial r} (rp) \quad (2.8)$$

Na Figura 2.2 são apresentadas ilustrações que facilitam o entendimento do conceito de onda esférica. Nela podemos ver as frentes de onda com geometria circular, se propagando a partir da origem. O efeito nos gases das imediações é o mesmo que para a onda unidimensional observada na Figura 2.1.

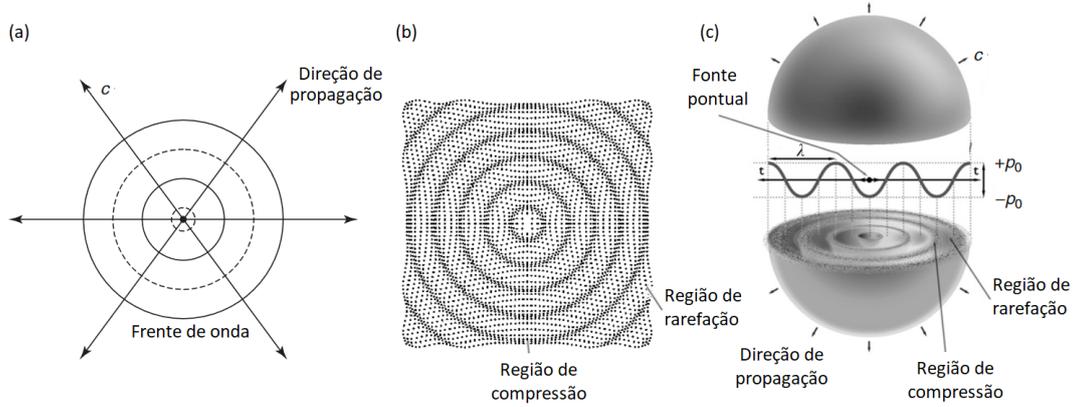


Figura 2.2: Visualização de uma onda acústica esférica se propagando a partir de uma fonte pontual (monopolo acústico). (a) Regiões de pressão constante (frente de onda), linhas cheias representam regiões de máxima pressão  $+p_0$  e linhas tracejadas regiões de mínima pressão  $-p_0$  (adaptado de [12]). (b) Visualização bidimensional das regiões de compressão determinada pelos pontos densos e de rarefação determinada pelos pontos espaçados (adaptado de [13]). (c) Visualização tridimensional (adaptado de [14]).

Como as áreas da frente de onda crescem com o quadrado da distância, longe da fonte sonora, os elementos de área que definem a região de pressão constante se tornam aproximadamente planos. Isso significa que a curvatura da frente de onda é negligenciável e estamos tratando, portanto, de uma onda plana como apresentada na Figura 2.1. A solução da Eq. 2.6 para uma onda plana assume o Laplaciano como sendo em duas coordenadas, e pode ser escrita como:

$$p(x, y, t) = Ae^{i(kx+ly-\omega t)} \quad (2.9)$$

Esta solução é bastante semelhante a solução unidimensional apresentada na Eq. 2.3. A única diferença é o aparecimento da componente  $l$  do vetor de onda  $\vec{k} = k\hat{x} + l\hat{y} + m\hat{z}$ , caracterizando a propagação bidimensional.

A velocidade de propagação do som, por sua vez, é descrita pela equação de Newton-Laplace [16, 17], onde  $\rho$  é a densidade do meio e  $K$  é o módulo volumétrico do meio:

$$c^2 = \frac{K}{\rho} \quad (2.10)$$

Durante a compressão do gás no deslocamento da onda acústica, ocorre liberação de calor isentropicamente, uma vez que os processo de compressão e rarefação são mais rápidos que o tempo necessário para o calor se dissipar [16]. A compressão isentrópica só é válida na condição de ondas acústicas de baixa frequência, uma vez que em alta frequência (maior que 100 MHz para líquidos) ocorre dispersão e absorção do som pelo meio em que se propaga (devido ao acoplamento com os processos moleculares) [17]. Para um gás ideal a dispersão

das ondas acústicas quase não ocorre, sendo ligeiramente perceptível apenas para ondas com comprimentos curtos. O processo de propagação se dá pela compressão/expansão isentrópica do gás, de maneira que o módulo volumétrico  $K$  é dito a entropia constante  $K = K_s$ . Assim o quadrado da velocidade de propagação da onda  $c$  é determinado pela Eq. 2.11, onde  $\gamma$  é o coeficiente de expansão adiabática oriundo da relação  $K_s = \gamma P$  [18]:

$$c^2 = \gamma \frac{P}{\rho} \quad (2.11)$$

Como para um gás ideal vale a Equação de Estado de Clapeyron  $PV = nRT$ , onde  $V$  é o volume do gás,  $n$  a quantidade de matéria do gás contida neste volume,  $T$  é a temperatura do gás e  $R$  é a constante universal dos gases perfeitos, obtém-se que:

$$c^2 = \gamma \frac{nRT}{V\rho} = \gamma \frac{nRT}{m} = \gamma \frac{RT}{M} \quad (2.12)$$

onde  $m = \rho V$  é a massa de gás,  $M = \frac{m}{n}$  é a massa molar do gás. Para o ar seco  $\gamma \approx 1,4$  (gás diatômico),  $M \approx 0,0289645$  kg/mol e  $R = 8,3145$  J/mol K. Convertendo a temperatura  $T$  de Kelvin para a escala Celsius  $T_k = T_c + 273,15$ , temos a expressão prática para a velocidade do som como:

$$c = 20,047 \sqrt{T_c + 273,15} \quad (2.13)$$

Para o tratamento dos algoritmos de DOA feito no capítulo 3 é importante ter em mente que estamos tratando de ondas acústicas não-dispersivas (Eq. 2.5) se propagando em um gás ideal (Eq. 2.13), com frente de onda plana (Eq. 2.9). Isso é uma condição para que a onda mantenha sua forma da origem até os detectores (microfones) permitindo que as considerações sobre sua chegada sejam puramente geométricas. Além disso, é importante notar que a não-dispersividade garante que a velocidade de fase da onda seja constante, condições essas que são exploradas pelas técnicas de diferença de tempo de chegada (TDOA).

## 2.2 Potência Acústica

As fontes sonoras puntiformes radiam ondas esféricas cuja intensidade acústica  $I$  decai com o quadrado da distância. Isso ocorre pois uma mesma potência acústica  $P_{ot}$  da fonte fornece uma taxa de energia que atravessa uma área superficial  $A_s$  da esfera circundante que cresce com o quadrado do raio  $r$ . Na Figura 2.3 isso fica evidente, uma vez que vemos a intensidade para cada frente de onda futura nas posições  $\vec{r} = r, \vec{r} = 2r$  e  $\vec{r} = 3r$ . Matematicamente, temos que a intensidade acústica radiada é dada por:

$$I = \frac{P_{ot}}{4\pi r^2} \quad (2.14)$$

Mas a intensidade também é definida como:

$$I = \frac{p^2}{Z} \quad (2.15)$$

onde  $p$  é a pressão e  $Z$  é a impedância acústica.

A impedância acústica mede o impedimento do movimento induzido por uma pressão aplicada a uma superfície [19]. Para uma onda plana no espaço livre, a impedância acústica é dada por  $Z = \rho c$  [19, 31]. Assim, para este tipo de onda, a potência acústica da fonte pode ser escrita como:

$$P_{ot} = \frac{4\pi r^2 p^2}{\rho c} \quad (2.16)$$

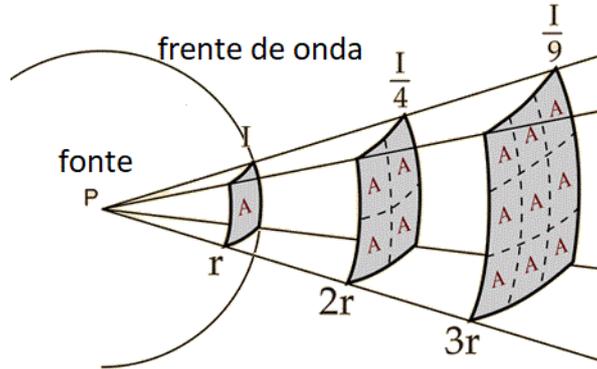


Figura 2.3: Representação da propagação da frente de onda acústica esférica a partir de uma fonte pontual, onde  $A$  representa elementos de área superficial pelos quais a frente de onda passa (adaptado de [21]).

Para monopolos acústicos, isto é, fontes pontuais, a irradiação das ondas acústicas ocorre igualmente para todos os lados [20]. A pressão é dada, na condição de simetria esférica e campo distante, por:

$$p(r, \theta, t) = i \frac{Q \rho c k}{4\pi r} e^{i(\omega t - kr)} \quad (2.17)$$

Com amplitude dada pelo módulo:

$$|p(r, \theta, t)| = \frac{Q \rho c k}{4\pi r} \quad (2.18)$$

onde  $Q e^{i\omega t} = \iint \vec{u} \cdot \hat{n} dA$  é uma constante que representa o volume do fluido (por exemplo, o ar) deslocado pela fonte (com  $\vec{u}$  sendo a velocidade do fluido em cada ponto da superfície em que se calcula o fluxo e  $\hat{n}$  o versor normal a esta superfície) [20]. Observa-se que a magnitude da pressão não depende do ângulo de orientação em relação a fonte sonora. Dessa forma, substituindo em 2.16, a potência acústica do monopolo é dada por:

$$P_{ot} = \frac{Q^2 \rho c k^2}{4\pi} \quad (2.19)$$

Para determinar potências de fontes não pontuais como dipolos, quadripolos e multipolos, em geral devemos conhecer a expressão do campo de pressão  $p$ , o que não se torna nada trivial e foge ao escopo central deste trabalho.

Não obstante, é comum referir-se ao som através do termo volume (referente a amplitude). O volume sonoro normalmente é definido em termos de uma escala relativa,

denominada SPL (do inglês, Sound Pressure Level) [12, 20, 22] ou NPS (Nível de Pressão Sonora, em português) [35]. Dada uma onda de pressão definida em coordenadas esféricas  $p(r, \theta, \phi, t)$ , definimos a pressão eficaz  $p_{rms}$  como sendo:

$$p_{rms}^2 = \frac{1}{2T} \int_{-T}^T p(r, \theta, \phi, t)^2 dt \quad (2.20)$$

A Eq.2.20 representa a média temporal da amplitude acústica. Assim, o nível SPL é definido por:

$$SPL = 20 \log_{10} \left( \frac{p_{rms}}{p_{ref}} \right) \quad (2.21)$$

onde  $p_{ref} = 20 \mu\text{Pa}$  e SPL é dado em dB. A Tabela 2.1 contém alguns exemplos de níveis sonoros em diferentes contextos.

Tabela 2.1: Exemplos de nível sonoro SPL em diferentes contextos.

SPL (dB)	Exemplo
0	Limiar de audição [12, 23].
10	Respiração humana a três metros [22].
20	Folhas caindo [12].
40	Conversa baixa [12].
60	Conversa normal [12, 22], escritório [23].
80	Rua com tráfego médio [12], aspirador de pó [22], shopping [23].
90	Caminhão a diesel [12].
94	Caminhão grande [23].
100	Britadeira [23].
110	Motocicleta acelerando a cinco metros [22].
120	Jato decolando [12], concerto de Rock [22].
130	Limiar de dor [22].
140	Limiar de dor [12, 23], três metros de um motor a jato [22].

## 2.3 Elementos de Captação Acústica

Os sinais acústicos normalmente são referidos como aqueles que se propagam no ar e que estão dentro do espectro audível humano. Apesar disso, os princípios acústicos discutidos anteriormente deixam claro que pode-se considerar outros meios de propagação e outras faixas de frequências. Aqui, contudo, focaremos nos elementos de captação acústica para o espectro audível humano, em ondas que se propagam através da atmosfera imediata. Estes dispositivos são chamados de microfones.

O princípio de funcionamento de qualquer microfone é realizar a transdução das ondas de pressão puramente mecânicas em ondas de eletricidade, por exemplo, da tensão elétrica entre dois terminais. Isso pode ser feito de maneira direta utilizando um diafragma, isto é, uma membrana flexível e elástica que vibra com as ondas acústicas, e um elemento transdutor, que converte a vibração captada pelo diafragma no sinal elétrico.

O primeiro microfone provavelmente foi feito por Johann Philipp Reis por volta de 1861,

Tabela 2.2: Exemplos de tecnologias de microfone.

Tecnologia	Descrição	Referências
Carvão	Ondas de pressão comprimem os grãos de carvão modificando a resistência elétrica do circuito.	[23, 24]
Capacitivo	Ondas de pressão vibram um diafragma que é uma das placas metálicas de um capacitor, alterando a capacitância do circuito.	[23, 24]
Dinâmico	Ondas de pressão vibram um diafragma acoplado a elemento magnético dentro de uma bobina, alterando o fluxo magnético do circuito.	[23, 24]
Piezoelétrico	Ondas de pressão comprimem um elemento piezo-elétrico que converte sua deformação mecânica em tensão elétrica.	[23, 24, 25]
MEMS	São capacitivos, piezoelétricos ou piezoresistivos feitos utilizando cristais e microeletrônica. MEMS significa <i>MicroElectrical-Mechanical System</i> , Sistema Micro Eletro-Mecânico.	[25]
Laser e Fibra Ótica	Ondas de pressão vibram um diafragma utilizado de anteparo reflexivo por um laser, modificando a luz captada por um sensor óptico. A deflexão da fonte luminosa pode ser usada para modular o sinal luminoso em um fibra ótica.	[27, 28]

utilizando uma tira metálica apoiada em uma ponta metálica sobre uma membrana [24]. Neste caso, a ponta metálica realizava contato conforme a membrana era movimentada pela pressão do ar. Contudo, este dispositivo não era sensível o bastante para captar variações como as produzidas pelos sinais de voz.

Ao longo dos tempos, as tecnologias de microfone evoluíram principalmente no tocante aos parâmetros físicos utilizados na transdução dos sinais. Hoje há uma vasta classe de dispositivos que usam diferentes variáveis físicas para converter ondas de pressão acústica em sinais elétricos ou luminosos. Na Tabela 2.2, elencamos algumas das tecnologias de microfone e seu princípios de funcionamento:

Algumas características gerais dos microfones são apresentadas a seguir:

- **Impedância:** A impedância elétrica é importante para o correto dimensionamento do circuito eletrônico ao qual o microfone será acoplado, desejando-se realizar o casamento das impedâncias destes elementos para que ocorra a máxima transferência de potência entre eles.
- **Sensibilidade:** A sensibilidade determina a intensidade do sinal que o microfone é capaz de fornecer, normalmente sendo dada pela tensão pico-a-pico entre os terminais do microfone. Este tipo de parâmetro nos informa a necessidade, ou não, de utilizar um circuito de pré-amplificação para que o sinal fornecido pelo microfone possa ser processado corretamente.
- **Resposta em Frequência:** A resposta em frequência determina a intensidade com que cada frequência é captada. Espera-se um microfone com resposta plana unitária,

isto é, sem ganho ou atenuação em nenhuma frequência específica, o que nem sempre é possível no mundo real.

- **Diretividade:** A diretividade representa o ganho direcional do microfone, isto é, a direção na qual ele possui maior sensibilidade. Normalmente são representados por diagramas polares. Algumas das possíveis diretividades são apresentadas na Figura 2.4.

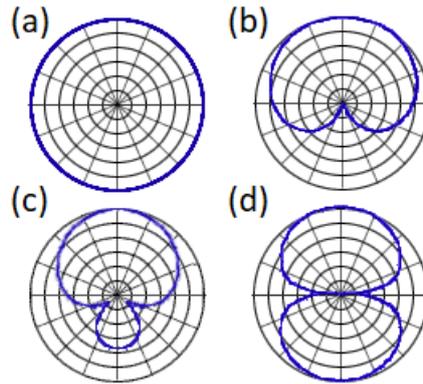


Figura 2.4: Exemplo de diretividade de microfones. (a) Ominidirecional. (b) Cadióide. (c) Hipercardióide. (d) Bidirecional. (Adaptado de [26]).

Os microfones ominidirecionais captam com a mesma sensibilidade em todas as direções, enquanto os cardióides e hipercardióides são ditos unidirecionais, captando com maior sensibilidade em uma direção específica. Há ainda os bidirecionais que captam bem em duas direções.

## 2.4 Sinal de Áudio

As ondas acústicas descritas anteriormente podem ser captadas por sensores específicos comumente denominados microfones. O microfone é responsável por transformar a vibração mecânica do gás atmosférico em um sinal elétrico, por exemplo. Existem diferentes formas de fazer isso e fizemos uma breve revisão das tecnologias de microfone na seção 2.3.

Este sinal elétrico inicialmente representa o som de maneira analógica, ou seja, as variações de pressão captadas pelo microfone são diretamente proporcionais as variações de tensão elétrica do sinal, isto é,  $p \propto v$  (admitindo-se, neste caso, a linearidade de resposta do sensor). Este sinal codifica informações sobre as ondas acústicas captadas que podem ser expressas através de uma representação matemática. Admitindo-se um microfone fixo em uma determinada posição  $\vec{r}$ , ele irá medir apenas a variação da pressão ao longo do tempo, isto é,  $p(t)$ . Uma vez que assumimos a periodicidade  $L$  da onda de pressão, podemos aproximar a variação de pressão  $p(t)$  utilizando uma representação  $s(t)$  do sinal captado através da série de Fourier:

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi t}{L}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi t}{L}\right) \quad (2.22)$$

onde

$$a_0 = \frac{2}{L} \int_0^L p(t) dt \quad (2.23)$$

$$a_n = \frac{2}{L} \int_0^L p(t) \cos\left(\frac{n2\pi t}{L}\right) dt \quad (2.24)$$

$$b_n = \frac{2}{L} \int_0^L p(t) \sin\left(\frac{n2\pi t}{L}\right) dt \quad (2.25)$$

A Eq. 2.22 representa uma soma infinita de cossenos e senos com diferentes magnitudes, determinadas através dos coeficientes  $a_n$  e  $b_n$ , enquanto  $a_0$  representa a componente constante do sinal (componente DC). Além disso ela é a solução temporal da Eq. 2.6, se assumirmos que a função da pressão é separável sendo dada pelo produto  $p(\vec{r}, t) = p_r(\vec{r})p_t(t)$ . A observação desta descrição pela soma de cossenos e senos é importante, uma vez que os sinais de áudio são extensivamente expressos através de combinações deste tipo (devido a periodicidade que apresentam).

Além disso, é importante notar que os sinais podem ser contínuos na variável de interesse (dependente) e na variável independente (tempo ou espaço), discretos na variável de interesse e contínuos na variável independente, contínuos na variável dependente e discretos na variável independente, ou, por fim, discretos tanto na variável dependente quanto na variável independente [5]. Sinais denominados analógicos são representações contínuas análogas de uma variável real, como a variação da tensão elétrica em relação ao tempo nos terminais de um microfone.

Podemos observar um sinal  $s(t)$  contínuo no domínio do tempo  $t$  em outros domínios como no da frequência  $\omega$  (através da Transformada de Fourier  $\mathcal{F}$ ) e no domínio  $s$  (através da Transformada de Laplace  $\mathcal{L}$ ). Podemos expressar isso matematicamente através de:

$$F(\omega) = \mathcal{F}(s(t)) = \int_{-\infty}^{\infty} s(t)e^{i\omega t} dt \quad (2.26)$$

$$L(s) = \mathcal{L}(s(t)) = \int_0^{\infty} s(t)e^{-st} dt \quad (2.27)$$

Por outro lado, podemos estar interessados em processar essa informação através de um microcontrolador ou processador. Estes dispositivos são condicionados para trabalhar com dados binários. Devemos utilizar o conceito de discretização da variável dependente, convertendo o valor medido para algum código binário equivalente, ou seja, quantizando os valores que um sinal pode assumir. Como sempre há limitação de memória, isto é, a quantidade de bits disponíveis em um determinado instante em um dispositivo real, teremos somente uma determinada quantidade de valores para descrever nosso sinal, que determinarão a resolução do nosso sistema. Por exemplo, se usarmos 4 bits para representar a amplitude do sinal medido e a amplitude máxima que se deseja medir for de  $5V$  (fundo de escala), então nossa resolução será de  $\frac{5}{2^4} = 0,3125V$  e teremos 16 valores diferentes de amplitude para amostrar. A amostragem é a discretização do tempo em que fazemos cada

medição de valor do sinal. Tendo valores quantizados e amostrados, temos a discretização da variável dependente e independente. O mesmo procedimento vale caso a variável independente fosse o espaço, e, neste caso, nossa resolução seria o menor comprimento medido por um único pixel em uma imagem. Sinais discretizados desta forma, são denominados sinais digitais.

Um sinal digital  $s(k)$  representa uma amostra quantizada da amplitude de  $s(t)$ , para um instante de tempo discretizado  $k$ . Aqui ainda valem as condições de espaço transformado, em especial através da Transformada de Fourier de Tempo Discreto (do inglês, Discrete Time Fourier Transform - DTFT) e da Transformada  $Z$ , respectivamente:

$$X(\omega) = \sum_{k=-\infty}^{\infty} x(k)e^{-i\omega k} \quad (2.28)$$

$$X(z) = Z(s(t)) = \sum_{k=-\infty}^{\infty} x(k)z^{-k} \quad (2.29)$$

Em se tratando de sinais analógicos de áudio, é comum o tratamento com  $s(t)$  e  $F(\omega)$ , no qual  $s(t)$  é uma função periódica como a descrita pela Série de Fourier da Eq. 2.22, e  $F(\omega)$  é denominado espectro do sinal, dividido em duas componentes, de magnitude  $|X(\omega)|$  e de fase  $\phi(X(\omega))$ . A magnitude é definida como a norma  $X(\omega)$  e a fase é o ângulo deste número complexo no plano de Argand-Gauss. O mesmo vale para a Transformada de Fourier de Tempo Discreto. Para sinais de áudio digitalizados é comum o tratamento a partir de um vetor  $s(k)$  onde  $k$  é o  $k$ -ésimo elemento do vetor (i.e., a  $k$ -ésima amostra) que contem valores discretos (quantizados) do sinal.

Como exemplo, a Figura 2.5 traz um sinal de áudio de fala humana capturado por um computador, com taxa de amostragem de 44100 Hz (cada segundo do sinal digital corresponde a 44100 amostras do sinal original):

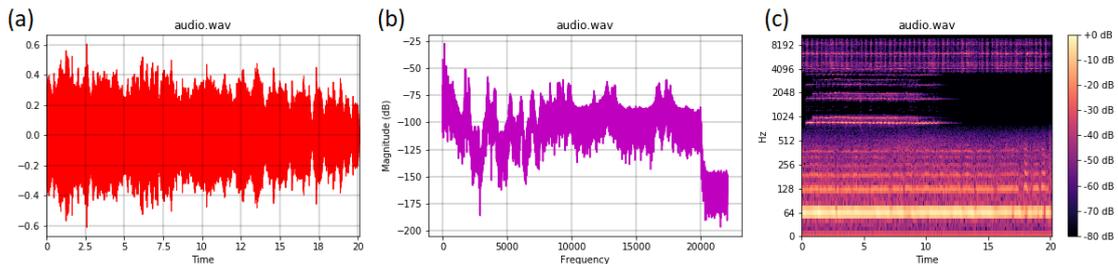


Figura 2.5: Exemplo de sinal de áudio de fala digitalizado. (a) Sinal  $s(k)$  mapeado para o tempo  $t$  da aquisição. (b) Espectro de amplitude do sinal  $|A(\omega)|$ . (c) Espectrograma do sinal.

O espectrograma em (c) é uma maneira de representar ambos os domínios do sinal, realizando a DFT (*Discrete Fourier Transform*) para intervalos definidos (frames) do sinal de áudio. Assim é possível observar a evolução temporal das magnitudes das frequências do sinal, como as componentes de aproximadamente 1000 Hz que cessam após aproximadamente 10 segundos de áudio. Outra observação interessante é o comportamento do microfone e do sistema de captação que atenuam frequências maiores do que 20 kHz como

pode-se ver com o corte abrupto no final do gráfico do espectro de amplitudes em (b). Isto ocorre pois há no sistema uma componente de filtro passa-baixas como um aperfeiçoamento do sistema devido ao fato do ouvido humano não perceber sons acima de 20 kHz.

Outro ponto importante para o tratamento de sinais de áudio (e sinais em geral) é a energia  $E$  do sinal, definida como:

$$E = \int_{-\infty}^{\infty} |s(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\hat{s}(f)|^2 df \quad (2.30)$$

onde a segunda igualdade é definida através do Teorema de Parseval, para o mesmo sinal  $s(t)$  agora no domínio da frequência,  $\hat{s}(f)$ .

Por outro lado, a potência do sinal é definida pela Eq. 2.31, onde  $\omega = 2\pi f$ :

$$S_{xx}(\omega) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T |s(t)|^2 dt \quad (2.31)$$

Usando a transformada de Fourier para transformar  $x(t)$  para  $\hat{x}(f)$ , e usando a definição de Esperança Estatística, podemos definir a Densidade Espectral de Potência (*Power Spectral Density*, PSD)  $S_{xx}$  como :

$$S_{xx}(\omega) = \lim_{T \rightarrow \infty} E[|\hat{x}(\omega)|^2] = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-i\omega\tau} d\tau = \hat{R}_{xx}(\omega) \quad (2.32)$$

onde  $R_{xx}$  é a função de Autocorrelação dos sinais para um atraso  $\tau$ . A Eq. 2.32 nos fornece uma forma prática de estimar a PSD do sinal, calculando-se a Transformada de Fourier da função de Autocorrelação <sup>2</sup>.

## 2.5 Processamento Digital de Áudio

Para realizar o processamento digital do áudio é necessário transformar o sinal analógico, captado pelo microfone, para um sinal digital. Essa etapa é chamada de conversão Analógico/Digital. Um sinal contínuo no tempo e na amplitude é discretizado no tempo, através do processo de amostragem, e quantizado na amplitude, devido a resolução finita dos sistemas digitais.

O processo de amostragem significa tirar amostras dos valores de amplitude em intervalos de tempo bem definidos. A frequência desses intervalos é denominada taxa de amostragem. O Teorema da Amostragem de Nyquist-Shannon introduz o conceito de que a taxa de amostragem  $f_s$  deve ser maior que a taxa de Nyquist  $f_N$  que equivale ao dobro da maior frequência  $f_{max}$  do sinal que se deseja amostrar. Isto é:

$$f_s \geq f_N = 2f_{max} \quad (2.33)$$

Um circuito eletrônico para se trabalhar com áudio no espectro audível deve amostrar o sinal de tensão elétrica na saída de um microfone a taxas maiores do que 40 kHz, uma vez que a audição humana é sensível até aproximadamente 20 kHz. Isso garante que serão digitalizadas todas as frequências audíveis.

---

<sup>2</sup>Dependendo da definição da Transformada realizada, divide-se o resultado por  $2\pi$ .

Este Teorema também adverte sobre o fenômeno de aliasing temporal, que ocorre quando amostramos o sinal a uma taxa menor do que  $f_N$ . Quando isso ocorre, uma componente de alta frequência pode ser identificada como uma componente de baixa frequência que não existe realmente no sinal amostrado.

Levando em consideração o Teorema de Nyquist-Shannon e as modernas tecnologias de digitalização e armazenamento de dados digitais, A Sociedade de Engenharia de Áudio norte-americana, recomenda utilizar 48 kHz como frequência de amostragem em sistemas de áudio digitalizado por Modulação por Código de Pulso (*Pulse-code Modulation*, PCM). Essa frequência é compatível com sistemas de vídeo e cobre o espectro audível humano. [29]

Por outro lado, existem diferentes técnicas de quantização. A quantização é necessária para mapear valores de amplitude em códigos binários que armazenam a informação da amplitude medida. Como os sistemas possuem uma memória finita, não é possível representar infinitos valores de amplitude de maneira unívoca. Dessa forma, são estabelecidos valores discretizados, isto é, quantidades definidas de amplitude que o sistema mede. Isso determina a resolução do sistema e o seu fundo de escala. De maneira mais precisa, o procedimento é dividido em duas etapas: a quantização do valor de amplitude seguida da codificação no valor binário. A quantização normalmente é definida pelo número de bits que descreve a base da codificação binária. Desta forma, um canal de captação com 24 bits pode representar  $2^{24} = 16777216$  valores de amplitude (uma extensão de até 144 dB).

Um dispositivo extremamente importante no contexto de sinais é o Processador Digital de Sinais (do inglês, *Digital Signal Processor* - DSP). Ele é responsável por implementar o hardware necessário para realizar a conversão analógico/digital, levando em conta uma taxa de amostragem do sinal captado, muitas vezes através de múltiplos canais de captação. Esta classe de dispositivos muitas vezes permite o processamento dos sinais em tempo real, podendo ser embarcados em diversos sistemas e equipamentos.

## 2.6 Processamento de Áudio com Arranjo de Microfones

Uma boa prática de aquisição de sinais de áudio, quando possível, é aquela que usa múltiplos canais, como através de um arranjo de microfones. A captação do sinal através desse tipo de arranjo permite um processamento mais sofisticado do áudio, possibilitando a aplicação de técnicas como Beamforming e TDOA (que serão apresentados no Capítulo 3). Neste caso, os múltiplos canais possuem uma diferença da localidade espacial onde captam o som. Outras possibilidades envolvem, por exemplo, utilizar múltiplos canais para aumentar o espectro da aquisição ou aumentar a resolução para regiões específicas do espectro, neste caso, utilizando canais acoplados a sensores especiais. Na Figura 2.6 é apresentado um cenário hipotético onde deseja-se determinar a localização de uma determinada pessoa que está falando através de um arranjo de microfones.

O sinal de interesse (do inglês, *Signal of Interest* - SOI) é captado por um arranjo de microfones em um ambiente com diferentes tipos de fontes sonoras concorrentes, adicionando ruído e interferência. A utilização de apenas um microfone para esta tarefa seria impeditiva uma vez que a informação da posição é extraída da diferença entre as propriedades

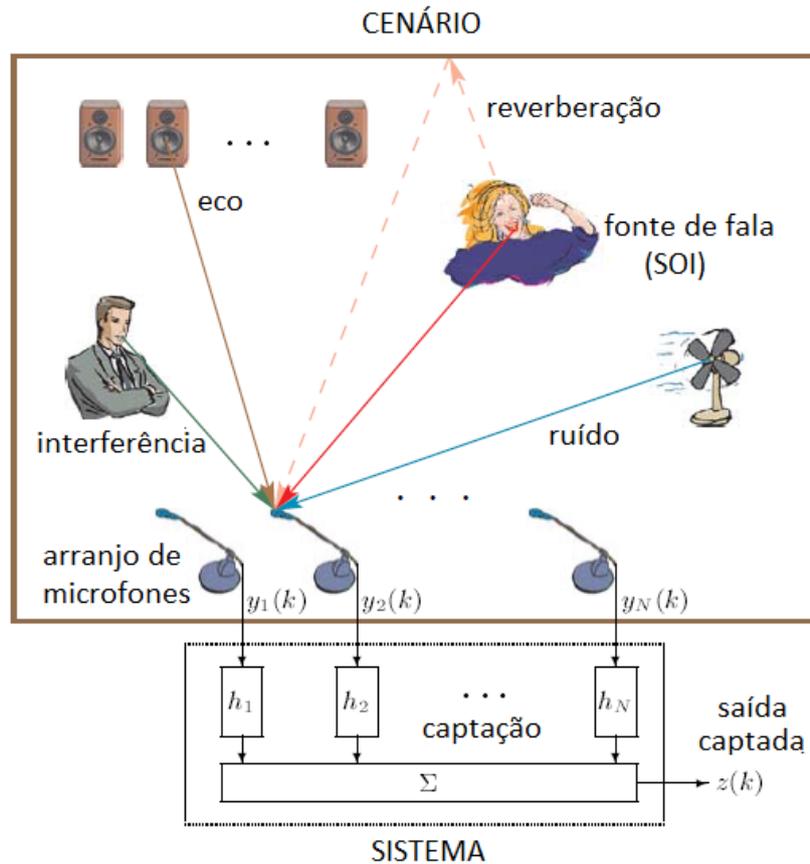


Figura 2.6: Esquema de um sistema de captação de áudio por arranjo de microfones atuando dentro de um cenário hipotético com diferentes fontes de sinais (adaptado de [30]).

de pelo menos dois sinais recebidos para um mesmo sinal emitido. Além disso, a aplicação de um arranjo de microfones não só permite determinar a localização das fontes sonoras, como também permite a redução de ruído, a redução de eco, a remoção da reverberação, a estimação do número de fontes, e a separação de múltiplas fontes [30].

Considerando que o espaço entre a fonte e o receptor é livre, e é usado um arranjo linear de sensores (microfones), a geometria dos raios que determinam a direção das ondas acústicas pode ser definida como:

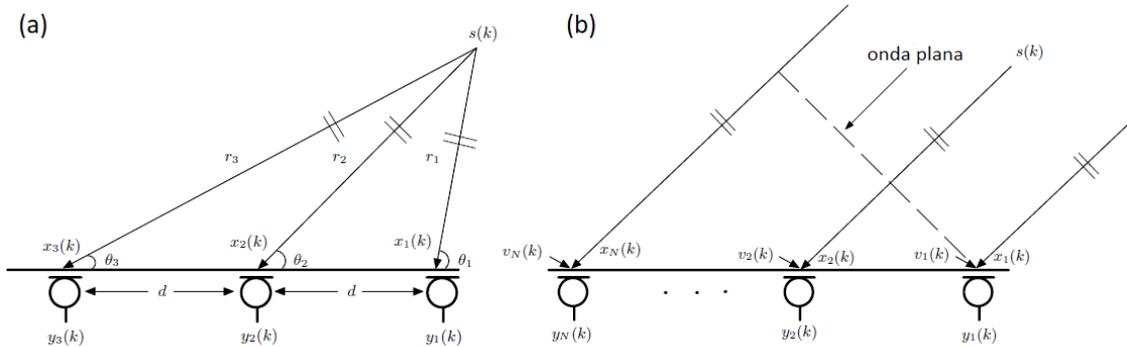


Figura 2.7: Modelo de raio para fonte sonora única com captação por arranjo de microfones. (a) Para campo próximo. (b) Para campo distante. Adaptado de [30].

Na Figura 2.7,  $s(k)$  denota o sinal da fonte, enquanto  $y_n(k)$  denota o sinal captado pelo  $n$ -ésimo microfone,  $v_n(k)$  é o ruído aditivo em cada microfone,  $r_n$  a distância entre o  $n$ -ésimo microfone e a fonte e  $d$  a distância entre cada um dos microfones.

Em campo distante, apesar da direção da fonte poder ser estimada, a determinação da distância  $r$  é um problema em aberto [31]. Já para campo próximo, é possível estimar tanto a direção da fonte quanto a sua distância.

A condição de campo distante é obtida sempre que o elemento receptor estiver a uma distância  $d$  maior do que a Distância de Fraunhofer  $d_F$ :

$$d \geq d_F = \frac{2D^2}{\lambda} \quad (2.34)$$

onde  $D$  é a maior dimensão do elemento radiador (antena ou alto-falante) e  $\lambda$  é o comprimento da onda radiada. Por exemplo, na Figura 2.8 são apresentados alguns resultados numéricos para três diferentes alto-falantes. A uma distância de 40 cm de um alto-falante de *home theater*, a onda acústica radiada está aproximadamente plana, para todo o espectro de frequência audível do ser humano.

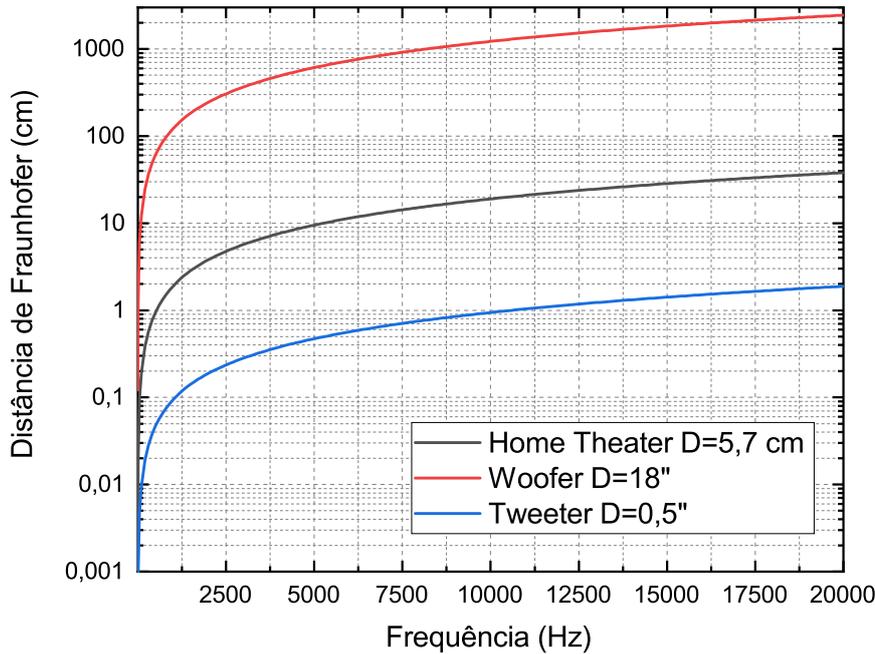


Figura 2.8: Distância de Fraunhofer em função da frequência sonora para três exemplos de alto-falantes.

Para o problema de uma única fonte, assumindo o modelo de campo aberto, o modelo do sinal captado pelo arranjo de  $N$  sensores, pode ser formulado, com tempo denotado por  $k$ , como:

$$y_n(k) = \alpha_n s(k - t - F_n(\tau)) + v_n(k) = x_n(k) + v_n(k) \quad (2.35)$$

onde  $n = 1, 2, \dots, N$ ,  $0 \leq \alpha \leq 1$  é a atenuação devido a efeitos de propagação,  $v_n(k)$  é o ruído aditivo no  $n$ -ésimo sensor,  $s(k) = s(k - t - F_n(\tau))$  é o sinal da fonte,  $t$  é o tempo de propagação do sinal da fonte até o sensor  $n = 1$ ,  $\tau$  é o atraso relativo entre os sensores 1 e 2,  $F_n(\tau)$  é o atraso relativo entre o sensor 1 e o sensor  $n$ , sendo que  $F_1(\tau) = 0$ ,  $F_2(\tau) = \tau$  [30]. É importante notar que a função  $F_n(\tau)$  reflete somente disposição geométrica dos sensores.

## Capítulo 3

# Técnicas de DOA

### 3.1 TDOA

As técnicas de TDOA baseiam-se no atraso entre os sinais recebidos devido a disposição geométrica dos microfones. Este tipo de técnica também é chamada de TDE, do inglês *Time Delay Estimation* [39, 38] ou multilateração. Como os sinais percorrem distâncias diferentes em tempos diferentes, assumindo uma velocidade de propagação constante, é possível utilizar essa informação temporal, em conjunto com a posição dos microfones, para estimar a direção de origem do som. Essa direção é a direção de chegada do sinal que estamos procurando.

Na Figura 3.1 é apresentado um modelo geométrico simplificado para o processamento do sinal de áudio captado por um arranjo com dois microfones, na condição de frente de onda plana (campo distante):

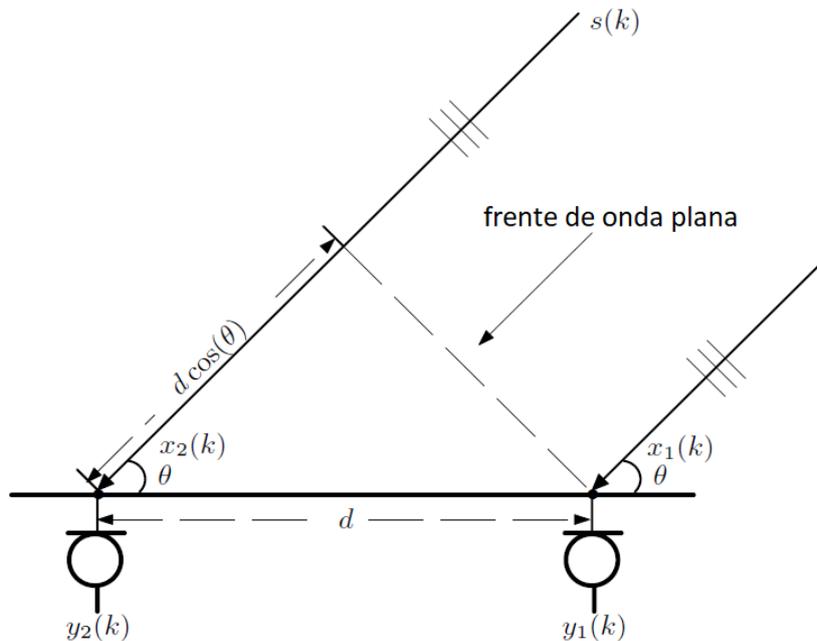


Figura 3.1: Ilustração da estimação de DOA em um problema bidimensional com dois microfones, com condição de campo distante. (Adaptado de [30]).

O tempo de atraso  $\tau_{12}$  entre a chegada do sinal  $s(k)$  no microfone  $n = 1$  e no microfone  $n = 2$  depende da distância  $r_1$  e  $r_2$  destes microfones com a fonte do sinal e da velocidade de propagação  $c$  da onda acústica:

$$\tau_{12} = \frac{r_2 - r_1}{c} \quad (3.1)$$

Como  $r_2 = r_1 + d \cos \theta$ , temos:

$$\tau_{12} = \frac{r_1 + d \cos \theta - r_1}{c} = \frac{d \cos \theta}{c} \quad (3.2)$$

Dessa forma, lembrando que  $\tau_{12} = \tau$ , o ângulo de chegada  $\theta$  pode ser determinado por:

$$\theta = \arccos\left(\frac{c\tau}{d}\right) \quad (3.3)$$

Em geral  $\tau$  não é conhecido mas a geometria dos sensores sim. O problema torna-se, então, estimar o atraso  $\hat{\tau}$  mais próximo do atraso  $\tau$  (real) devido ao caminho geométrico.

Para microfones equidistantes em um arranjo linear:

$$F_n(\tau) = (n - 1)\tau \quad (3.4)$$

Para microfones em um arranjo linear, mas não equidistantes:

$$F_n(\tau) = \frac{\tau}{d_1} \sum_{i=1}^{n-1} d_i \quad (3.5)$$

Para ambos os casos  $n = 2, \dots, N$ .

Os valores de  $\tau$  que podem ser determinados univocamente para arranjos lineares estão entre  $-\frac{d}{c}$  e  $+\frac{d}{c}$ , de forma que a maior frequência que pode ser detectada sem ambiguidades é dada por  $\frac{c}{d}$  (uma vez que se deve ter, no máximo, um comprimento de onda inteiro entre os dois sensores) [38]. Isto é um exemplo de limitação devido a taxa de amostragem espacial, e pode ser desprezado se as componentes de baixa frequência do som possuírem a maior proporção de potência do sinal.

Para uma geometria não linear,  $F_n$  pode ser uma função não linear. Uma forma de estimar o atraso, neste caso, é através do produto escalar entre a direção de chegada e o vetor posição do microfone. Essas correções podem ser expressas como [43]:

$$\tau(n) = -\frac{\vec{m}_n \cdot \vec{q}_n}{c} \quad (3.6)$$

onde  $\vec{m}_n$  é o vetor de posição do canal (microfone)  $n$ ,  $\vec{q}_n$  é o vetor que determina a direção de chegada da onda sonora no mesmo microfone,  $c$  é a velocidade de propagação da onda no meio e  $\tau(m_n)$  é o atraso do sinal recebido no microfone em relação a sua posição no sistema de coordenadas. O sinal de menos representa o atraso. Essa equação é equivalente a que definimos para o arranjo linear (Eq. 3.2) onde  $d \cos \theta$  é o resultado de  $\vec{m}_n \cdot \vec{q}_n$ . Ela é válida para arranjos com geometria arbitrária, como linear e circular. Caso façamos a origem do sistema em um dos microfones, digamos no microfone  $n = 1$ , então  $\vec{m}_n = (0, 0, 0)$  e  $\vec{m}_n \cdot \vec{q}_n = 0$ , de forma que  $\tau(m_n) = 0$ . Isso significa que o sinal capturado pelo microfone

$n = 1$  não está atrasado (com relação a nossa parametrização do problema). É importante notar que o método não tem nenhuma informação da distância entre a fonte do som e o microfone, apenas sua orientação com relação ao sistema de coordenadas utilizado.

Para uma onda esférica, cuja origem é uma fonte pontual (monopolo acústico), a frente de onda não será plana nas proximidades da fonte (campo próximo). Dessa forma o ângulo de incidência não será constante para cada microfone. Na Figura 3.2 é apresentada essa situação.

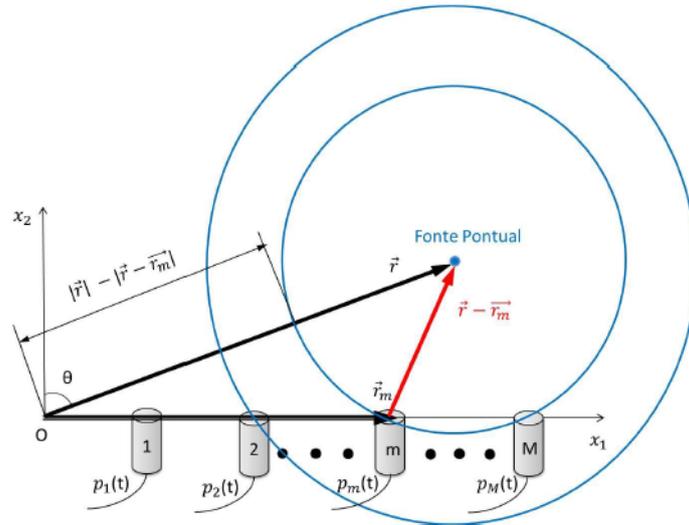


Figura 3.2: Ilustração da estimação de DOA em um problema bidimensional com  $M$  microfones, na condição de campo próximo. (Adaptado de [35]).

Neste caso a correção dos tempos deve levar em consideração cada vetor  $\vec{r} - \vec{r}_n$  que determina a distância do microfone  $N$  para a fonte. A expressão do tempo de atraso  $\tau(n)$  para o  $n$ -ésimo microfone pode ser reescrita como:

$$\tau(n) = \frac{|\vec{r}| - |\vec{r} - \vec{r}_n|}{c} \quad (3.7)$$

Para verificar o alinhamento dos sinais com as correções de atraso, podem ser utilizadas medidas de coerência entre os sinais [41]. Essa medidas podem ser usadas para se determinar correções que ampliam os picos dos sinais recebidos o que torna a reverberação do som uma importante desvantagem das técnicas de TDOA, pois estas podem causar picos de maior amplitude que não correspondem a correção da origem real do sinal [39].

### 3.1.1 Técnicas baseadas em Medidas de Coerência

Estas técnicas se baseiam na necessidade de medir a similaridade entre sinais em algum momento do processamento. Contudo, existem diversos parâmetros físicos que podem ser usados para comparar os sinais. Alguns deles são a correlação cruzada, funções de momentos, função de verossimilhança, medidas de ruído por sinal, energia e entropia [41]. A seguir iremos tratar duas medidas de coerência, a Correlação Cruzada Generalizada e o Semblance.

### 3.1.1.1 Correlação Cruzada Generalizada

O método de correlação cruzada generalizada (*Generalized Cross-correlation*, GCC) é um dos mais simples e mais utilizados para a estimação de TDOA [31, 41]. É um algoritmo que usa um estimador do tipo ML (*maximum likelihood*, um estimador de probabilidade máxima), presumindo estacionariedade, pelo menos para um tempo finito de aquisição [42].

Para o caso de DOA, assume-se  $x_n$ , onde  $n = 1, \dots, N$  representa o  $n$ -ésimo microfone, ou seja, o  $n$ -ésimo sinal captado. A correlação cruzada contínua é calculada entre os canais, por exemplo  $n = 1$  e  $n = 2$ , como  $R_{x_1x_2}$ , através de:

$$R_{x_1x_2}(\tau) = \frac{1}{T - \tau} \int_{\tau}^T x_1(t)x_2(t - \tau)dt \quad (3.8)$$

A aplicação da correlação cruzada no método GCC para dois canais é apresentado na Figura 3.3, onde  $\hat{D}$  é o atraso estimado,  $H_1$  e  $H_2$  são filtros propriamente escolhidos para facilitar a estimativa do atraso real  $D$ . O procedimento de multiplicação dos sinais e posterior integração é exatamente o cálculo da correlação cruzada. Por fim, o algoritmo busca um pico no vetor  $R_{x_1x_2}$  elevado ao quadrado. Desta forma, o método tenta determinar para qual atraso a função de correlação cruzada é máxima [39].

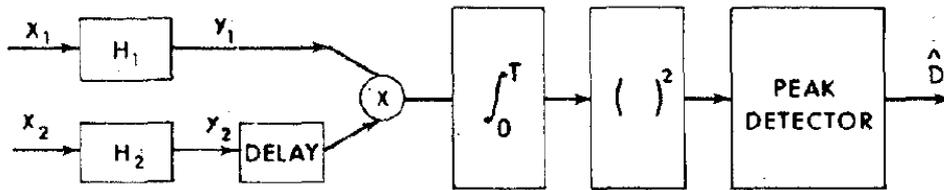


Figura 3.3: Fluxo de processamento para o DOA-GCC (extraído de [42]).

Para sinais aleatórios representados por vetores (sequências ordenadas de valores), a matriz de correlação cruzada  $R_{x_1x_2}$  pode ser calculada como:

$$R_{x_1x_2} = E[X_1X_2^T] \quad (3.9)$$

onde  $E$  é o operador de esperança estatística,  $X_1$  e  $X_2$  são os vetores que representam os sinais dos quais quer se calcular a correlação cruzada. Para sinais discretos a matriz de correlação cruzada também pode ser calculada através de:

$$R_{x_1x_2}(l) = \sum_{k=0}^L x_1(k)x_2(k-l) \quad (3.10)$$

onde  $L$  é o total de amostras do sinal (tamanho do vetor que representa o sinal) e  $l$  representa o instante da  $k$ -ésima amostra de atraso (mesma função do  $\tau$  na Eq. 3.8).

### 3.1.1.2 Semblance

*Semblance* (do inglês, aparência, semelhança) é uma técnica de medida de coerência feita entre sinais no domínio do tempo relacionada diretamente a energia dos sinais [41].

É uma técnica comumente aplicada no contexto de sismica de reflexão [9, 41]. A medida de coerência é feita com dados multicanal. A energia é calculada do empilhamento<sup>1</sup> dos sinais e do somatório das energias dos canais. A razão entre essas duas energias, determina a função de Semblance, um grau de medida da coerência entre os sinais. Desta forma, quanto mais alto o valor de Semblance, mais coerente são os sinais.

A Eq. 3.11 determina a função de Semblance  $Z$ , onde os somatórios no numerador dividido por  $N$  determinam a energia média por canal do sinal resultante do empilhamento dos sinais de todos os canais, e o denominador determina o somatório das energias dos sinais de cada canal [43]:

$$Z = \frac{1}{N} \frac{\sum_k \left| \sum_{n=1}^N \hat{s}_n(k) \right|^2}{\sum_k \sum_{n=1}^N |\hat{s}_n(k)|^2} \quad (3.11)$$

sendo  $\hat{s}_n(n)$  é o sinal no  $n$ -ésimo microfone,  $k$  é a  $k$ -ésima amostra e  $N$  é o número total de microfones utilizados.

A partir da Eq. 3.6, assumindo  $c$  conhecido e constante, deseja-se obter a direção de chegada  $q_0$ , que é a direção de chegada da onda sonora na origem do nosso sistema de coordenadas,  $(0, 0, 0)$ . Para isso, é possível corrigir o atraso para cada microfone, supondo uma determinada direção de chegada. A direção de chegada que corrige corretamente o atraso de todos os microfones é escolhida como a direção estimada. Quando isto acontece, a função coerência Semblance atinge o seu valor máximo, uma vez que a energia do empilhamento dos sinais em fase é maior.

Portanto, se o vetor de direção  $\vec{q}_n$  for definido pelo par ordenado  $(\theta, \phi)$  representando o ângulo de azimute e de elevação em um sistema de coordenadas esférico, usando um parâmetro de passo para a varredura iterativa, podemos calcular  $Z(\theta, \phi)$ , obtendo uma matriz de coerência que representa as orientações que corrigem o atraso dos sinais. Isso pode ser visto na Figura 3.4:

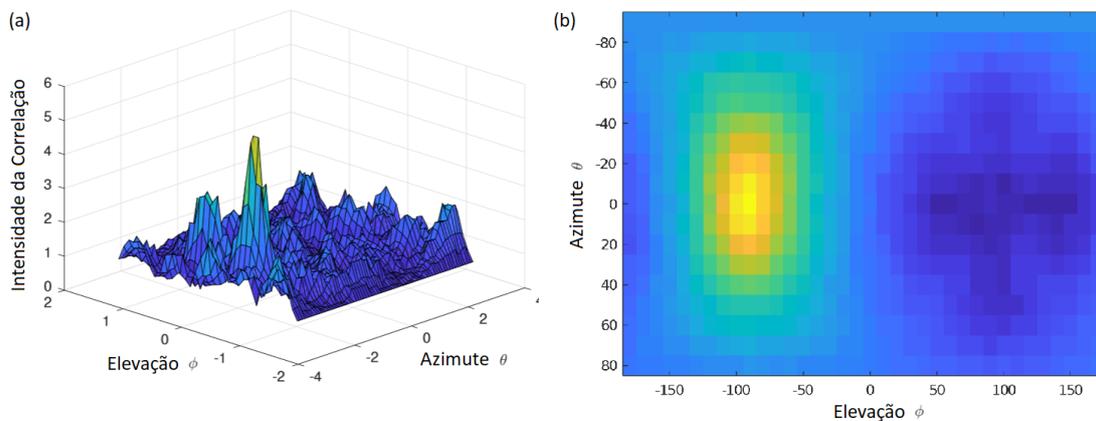


Figura 3.4: Visualização do resultado do método de Semblance. (a) Superfície 3D representando o resultado da função de Semblance. Os picos correspondem a alta correlação. (b) Vista 2D, curvas de nível de (a), denominada Painel de Semblance.

<sup>1</sup>Empilhamento neste caso significa a soma dos sinais, amostra a amostra.

É importante notar que o Painel de Semblance contém informações sobre múltiplas fontes, sendo representada por múltiplos picos. Apesar disso ser uma evidente vantagem, dependendo da capacidade de resolução e do ambiente onde se realiza o SSL, ruídos sonoros podem comprometer a localização exata. Como o algoritmo se baseia em uma busca exaustiva, o custo computacional pode ser considerado elevado.

O pico de coerência representa os ângulos que aumentam a semelhança entre os sinais de cada canal, do ponto de vista da energia. Uma vez que, com a correção de tempo espera-se que todos os sinais estejam alinhados, então a energia calculada da combinação dos canais será máxima. Na Figura 3.5 é apresentado um exemplo do alinhamento dos sinais através da correção de tempo para cada canal que corresponde ao pico de coerência no Painel de Semblance:

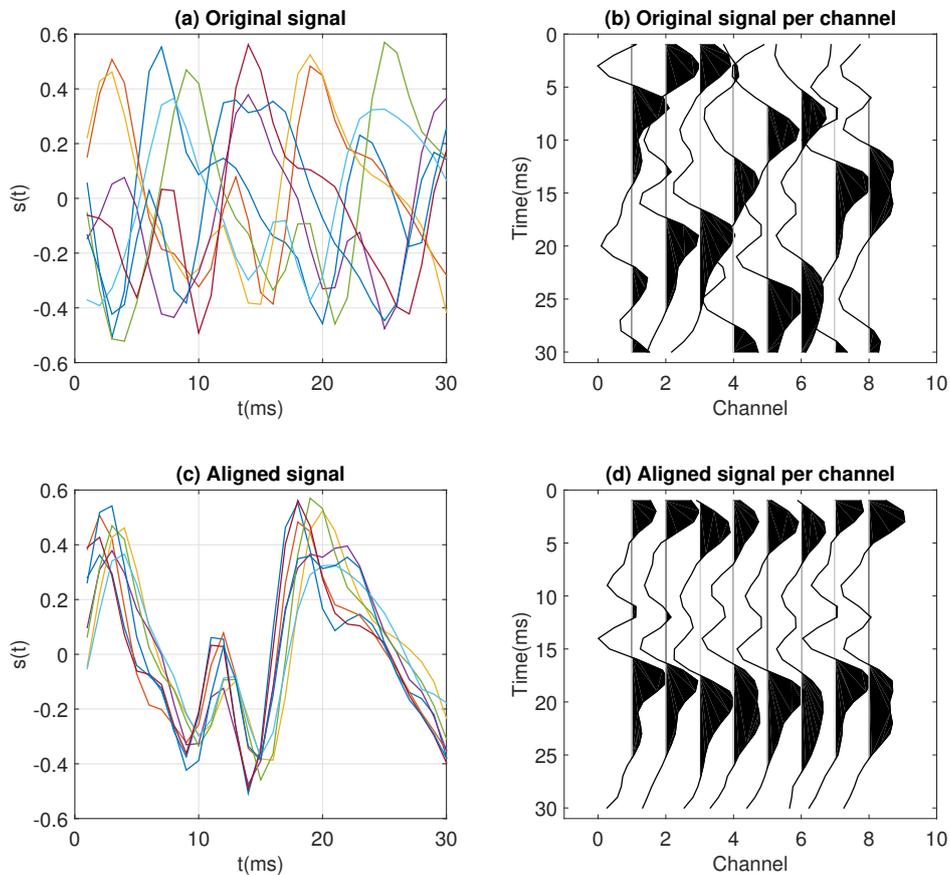


Figura 3.5: Exemplo de alinhamento dos sinais. (a) Sinal de 8 canais original representado em amplitude e tempo. (b) Sinal original com representação por canal. (c) Sinal corrigido para o pico de Semblance. (d) Sinal corrigido com representação por canal. (Adaptado de [43])

Outras considerações sobre a implementação deste algoritmo para o problema de TDOA podem ser obtidas em [43], e um procedimento para caso especial de tratamento de ruído utilizando inteligência artificial para aprimoramento da acurácia na detecção da direção de chegada em [44].

### 3.1.2 Técnicas baseadas em Decomposição por Autovalor

#### 3.1.2.1 MUSIC

É possível utilizar a decomposição em autovalores do sinal através de um Algoritmo de Decomposição por Autovalor (do inglês, *Eigenvalue Decomposition Algorithm* - EDA) de tal forma que o autovetor correspondente ao autovalor mínimo da matriz de covariância de uma aquisição multicanal conterá a resposta ao impulso entre a fonte do sinal e os elementos de captação. Dessa forma é possível estimar o tempo de atraso do sinal em cada canal. [39].

O algoritmo de classificação múltipla de sinais (em inglês, *Multiple Signal Classification*, MUSIC) foi primeiramente proposto por Ralph Schmidt em 1986 [45] e é, provavelmente, o algoritmo por decomposição por autovalor mais conhecido [38].

Os dados utilizados por está técnica podem ser representados através de matrizes, de forma que os sinais  $x$  recebidos em  $N$  canais são iguais aos sinais da fonte  $F$  corrigidos pelos ganhos nas direções  $a(\theta)$  para as  $D$  frentes de onda incidentes:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} a(\theta_1) & a(\theta_2) & \cdots & a(\theta_D) \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_D \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \quad (3.12)$$

onde  $v_n$  representa o ruído aditivo em cada canal e  $\theta$  o parâmetro de coordenada utilizado. Para um sistema em que se quer medir elevação e azimuth, por exemplo, assume-se  $a(\theta, \phi)$ .

Simplificando a notação:

$$X = AF + V \quad (3.13)$$

Assumindo que os sinais recebidos e o ruído são descorrelacionados, a matriz de covariância  $R_{xx}$  pode ser escrita como[45]:

$$R_{xx} = \hat{X}\hat{X}^* = A\hat{F}\hat{F}^*A^* + \hat{V}\hat{V}^* = APA^* + \lambda S_0 \quad (3.14)$$

Determinando os autovalores e autovetores de  $R_{xx}$ , reduz-se  $R_{xx}$  ao subespaço  $G$  dos autovetores, chamado de pseudoespectro, tal que [31, 45]:

$$P_{MUSIC}(\theta, \phi) = \frac{1}{a^*(\theta, \phi)GG^*a(\theta, \phi)} \quad (3.15)$$

Como  $P = \hat{F}\hat{F}^*$ , obtém-se dessa forma, como resultado, uma medida da direção de chegada do sinal.

Apesar do algoritmo MUSIC ter uma resolução angular melhor [45], ele necessita conhecer previamente a frequência da fonte, tendo que realizar cálculos prévios para solucionar isso, que contribuem para um custo computacional elevado.

## 3.2 Beamforming

Conformação de Feixe (do inglês, *Beamforming*) é uma técnica utilizada em diferentes contextos, como em radares, em sonares, em sismologia e em telecomunicações, sendo usada para diferentes propósitos como a detecção da presença de um sinal, a estimação da direção de chegada e a focalização de um sinal desejado em cenários com ruído, reverberação e outras fontes competindo pelo canal. Essa técnica é formulada como um filtro espacial que opera sobre um arranjo de sensores (antenas, microfones) para formar um determinado padrão de feixe (modificar a diretividade do arranjo). [30, 32]

Por exemplo, na Figura 3.6 vemos um feixe formado por uma antena altamente direcional e outro feixe formado por um arranjo linear (com distância variável) de microfones. Os feixes observados representam o diagrama de radiação do elemento. Enquanto antenas, em geral, podem transmitir e receber sinais, os microfones apenas recebem e os alto-falantes apenas emitem. De toda forma, o conceito abordado é equivalente e deve ser entendido como a direção de máximo ganho do elemento (ou arranjo de elementos).

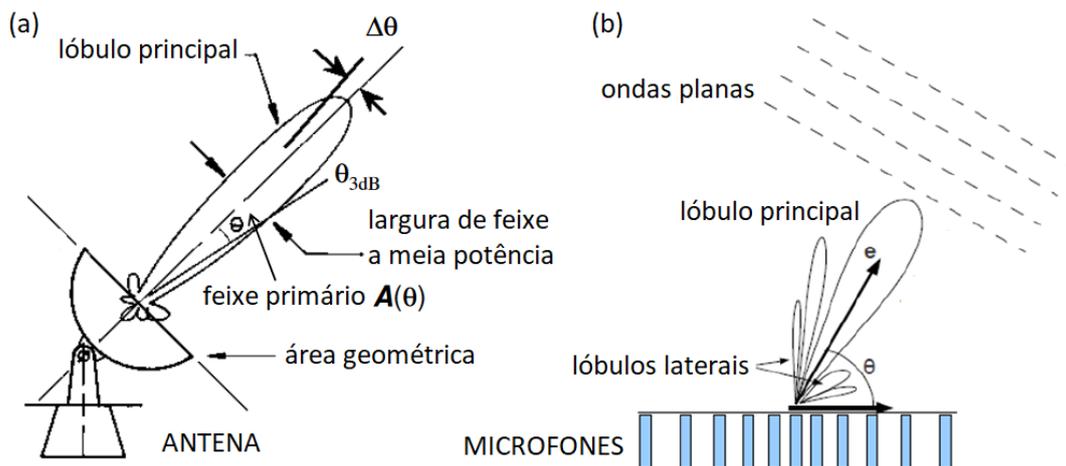


Figura 3.6: Representação do feixe de radiação (diretividade). (a) De uma antena (adaptado de [33]). (b) De um arranjo de microfones (adaptado de [34]).

Assim como para antenas, alto-falantes também possuem um diagrama de radiação. Enquanto para aquelas os diagramas são construídos para o campo elétrico e para o campo magnético, para estes os diagramas representam o campo de pressão. Ambos os diagramas podem ser convertidos em escalas relativas e expressos em decibéis, o que é mais usual e prático. Na Figura 3.7 é apresentado o diagrama de radiação de alguns arranjos de alto-falantes, onde é possível ver a formação dos lóbulos de radiação (aumento da diretividade) para cada arranjo.

O mesmo tipo de diagrama por ser formulado em uma representação tridimensional, denominada Mapa Acústico [35], similar a representação do diagrama de radiação eletromagnética. O termo Mapa Acústico ainda é empregado em um contexto mais amplo (como de urbanismo [36, 37]) no qual o Nível de Pressão Sonora (NPS) é apresentado em termos de coordenadas retangulares, normalmente sobre a planta de um estabelecimento ou localidades, indicando algum comportamento do som naquele local. A Figura 3.8 representa

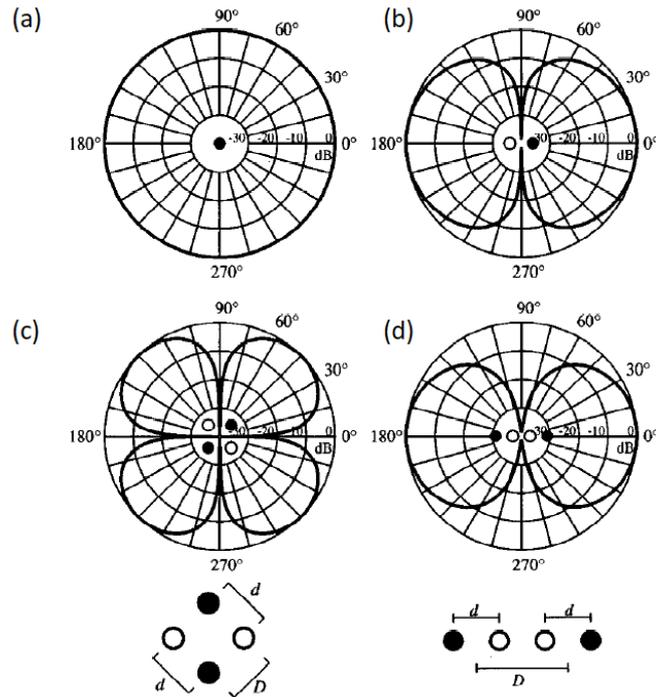


Figura 3.7: Representação polar da radiação acústica a partir de diferentes fontes, onde  $d$  é a distância entre os elementos radiadores e  $D$  é a maior dimensão da fonte. (a) Monopolo. (b) Dipolo. (c) Quadripolo lateral. (d) Quadripolo longitudinal. (Adaptado de [20]).

essas duas situações, dando ênfase que para o Beamforming é prevalectente a representação do diagrama radiador principal, como apresentado em (a). Em (b) o fenômeno acústico pode ser mais complexo, envolvendo múltiplas fontes de ondas acústicas, efeitos de reflexão e reverberação, não sendo um exemplo de diagrama de radiação, mas uma representação direta do NPS resultante naquela região, no instante do levantamento do Mapa Acústico.

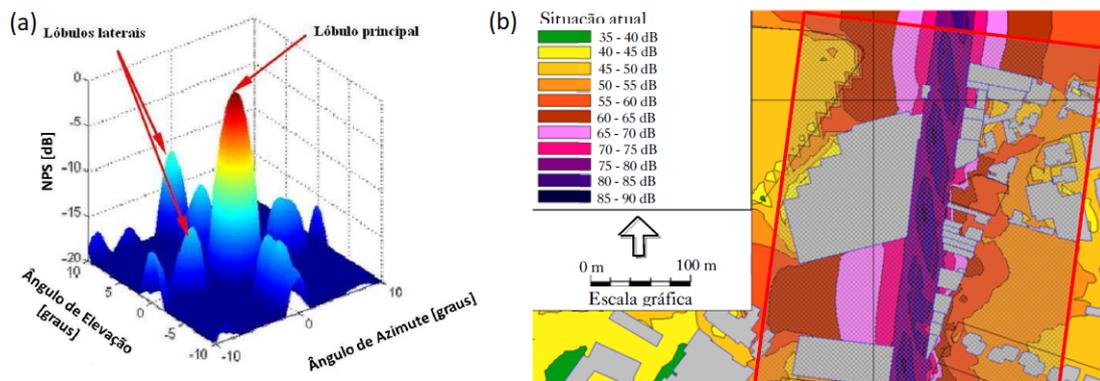


Figura 3.8: Mapa Acústico. (a) Diagrama de radiação NPS em coordenadas polares (adaptado de [35]). (b) Diagrama de NPS em coordenadas retangulares sobre um mapa de vista área de uma região metropolitana (adaptado de [37]).

O Beamforming pode ser decomposto em duas etapas: a sincronização e a soma ponderada. A sincronização é a tarefa de atrasar ou adiantar a saída de cada sensor para que o sinal resultante da soma deles seja um sinal de uma dada direção preferencial. A ponde-

ração determina o peso de cada sinal nesta soma. Desta forma a sincronização determina a direção (angulação) do lóbulo principal e a ponderação determina a largura do lóbulo principal e as características dos lóbulos secundários. [30]

O diagrama deste processo é apresentado na Figura 3.9. Nela podemos identificar os principais elementos do processamento do Beamforming, sendo eles o Arranjo de Sensores (microfones), o filtro de atraso (sincronização), o filtro de ponderação, o elemento de soma dos canais e por fim a ponderação pelo número de microfones.

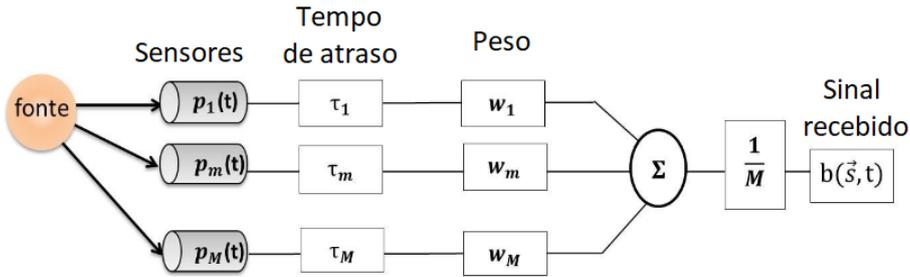


Figura 3.9: Esquema representativo do método de Beamforming (extraído de [35]).

Este tipo de procedimento também é chamado de Apontamento por Soma e Atraso (*Delay and Sum Beamformer*, DSB, em inglês), e uma das principais vantagens é que necessita de apenas um ciclo de cálculos para estimar múltiplas fontes [38]. Da mesma forma, o Beamforming é aplicado no caso de DOA através da busca de máximos na estimativa da distribuição de potência do espectro espacial dos sinais acústicos [39]. Além disso, por causa da simplicidade e do baixo custo computacional, o Beamforming provavelmente é um dos métodos mais utilizados em contexto de robótica [40].

O sinal  $b$  representado na Figura 3.9 pode ser expresso por <sup>2</sup>:

$$b(\vec{q}, t) = \frac{1}{N} \sum_{n=1}^N w_n s_n(t - \tau(\vec{q})) \quad (3.16)$$

onde  $N$  é o número de sensores (microfones),  $w_n$  é o peso da soma do sinal captado no  $n$ -ésimo microfone,  $s_n$  é o sinal captado no  $n$ -ésimo microfone e  $\tau(\vec{q})$  é o atraso corrigido para a direção  $\vec{q}$ .

No caso de uma onda plana, como apresentada na Eq. 2.9, observa-se que a amplitude da onda medida independe da posição do microfone no espaço, logo  $w_n = 1$  para todos os microfones. Já para o caso de uma onda esférica, como apresentada na Eq. 2.17 para o monopolo acústico, há um fator  $r$  no denominador que caracteriza a dependência da amplitude em relação a posição espacial, e o peso do sinal recebido em cada microfone deve ser corrigido como  $w_n = \frac{|\vec{r} - \vec{r}_n|}{|\vec{r}|}$  [35].

O métodos de Beamforming não exigem o cálculo de diferenças de tempo [39] como no caso do TDOA.

<sup>2</sup>Aqui realizamos uma troca de notação com relação a Figura 3.9 para manter o estilo com resto do texto.

# Capítulo 4

## Metodologia

### 4.1 Objetivos

Este trabalho teve como objetivo estudar as técnicas utilizadas para a Localização de Fontes Sonoras, abordando conceitos importantes sobre acústica e processamento de sinais. Para tanto, demonstramos a construção de um sistema real de estimação de direção de chegada utilizando uma técnica de diferença de tempo de chegada (TDOA) que emprega como função de coerência o Semblance.

### 4.2 Hardware Utilizado

Foram utilizados um computador de placa única Raspberry Pi 3, modelo B, e uma placa ReSpeaker com arranjo de 6 microfones dispostos circularmente, conforme apresentados na Figura 4.1, para realizar os procedimentos de DOA.

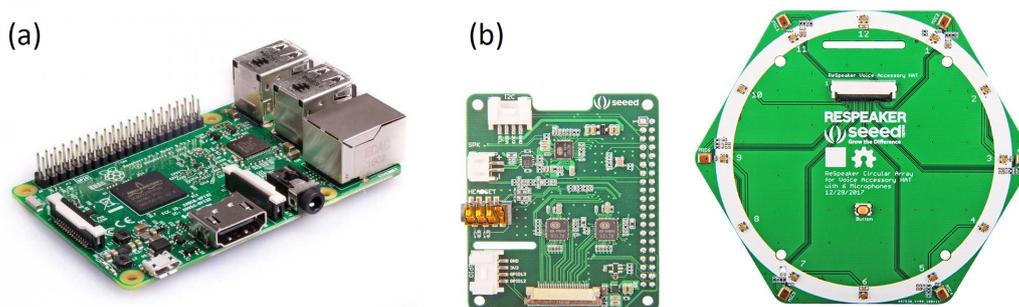


Figura 4.1: Dispositivos comerciais utilizadas. (a) Raspberry Pi 3, Model B (adaptado de [46]). (b) Arranjo circular de 6 microfones ReSpeaker (adaptado de [47]).

Para o processamento do sinal de som, a placa Respeaker utiliza dois conversores A/D AC108 [48], cada um de quatro canais, com suporte a comunicação I2C/I2S e com taxas de amostragem de 8kHz, 16kHz, 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz e 96KHz. Para a captação, são usados 6 microfones MSM321A3729H9CP, tecnologia MEMS, omnidirecionais, com curva de resposta em frequência plana até 4 kHz [49].

O Raspberry Pi possui quatro entradas USB que foram usadas para acoplar periféricos de entrada (teclado e mouse), e como fonte de alimentação e via de programação de uma

placa Arduino Uno.

A placa Arduino Uno utiliza um microcontrolador ATmega328P, com 14 pinos digitais e 6 analógicos, memória flash de 32 KB e SRAM de 2 KB [50]. O Arduino foi utilizado para atuar sobre um sistema de servo motores com o intuito de apontar um laser para a direção predita.

### 4.3 Ambiente de Desenvolvimento

Os códigos deste trabalho foram desenvolvidos em Python [51] que é uma linguagem interpretada de propósito geral de alto nível, com suporte múltiplos paradigma (procedural, funcional e orientada a objeto). Além disso Python é dinamicamente tipado e possui *garbage collector*. O Python utilizado no desenvolvimento foi o 2.7, apesar de, a priori, as bibliotecas utilizadas terem suporte em Python 3.

Para a programação do Arduino Uno através de scripts Python, foi utilizada a biblioteca *nanpy*.

O Raspberry Pi rodava sistema operacional Raspbian GNU/Linux 9, que é uma distribuição Linux baseada no Debian, disponibilizada através da licença GPL [52].

Para a utilização do ReSpeaker foram instalados os drives sugeridos pelo fabricante em [47, 53].

### 4.4 Avaliação de Desempenho

Foram realizados testes para verificar a acurácia na estimação da direção da fonte sonora da solução aqui proposta. Para sistematizar os testes, realizamos separadamente testes de azimute e elevação. Além disso, realizamos um teste preliminar para verificar a condição de campo distante, regime no qual o algoritmo funciona. Um exemplo de configuração experimental é apresentado na Figura 4.2, a seguir:



Figura 4.2: Exemplo de cenário de teste.

Como fonte sonora a ser localizada, foram utilizados sinais de áudio de músicas, com a presença de frequências variadas de todo o espectro audível. Em todos os casos testados, as capturas de áudio foram feitas com uma taxa de amostragem de 48 kHz.

#### 4.4.1 Testes

Foram realizados os seguintes testes da solução de DOA:

- **Distância de Fraunhofer:** para determinar a condição na qual o algoritmo de DOA implementado funciona, isto é, a condição de campo distante;
- **Azimute:** para verificar a acurácia na detecção do azimuth real de uma fonte sonora;
- **Elevação:** para verificar a acurácia na detecção da elevação real de uma fonte sonora;
- **Múltiplas Fontes:** para verificar se o algoritmo consegue detectar múltiplas fontes sonoras.

Para avaliar estes testes foram realizadas medidas do posicionamento real das fontes. Também foi contabilizado o tempo de processamento do algoritmo para alguns casos, com intuito de demonstrar a ordem de tempo do algoritmo.

Também foram realizados testes de elevação e azimuth para o direcionamento do aparato de sinalização (laser pointer), a fim de mensurar a precisão mecânica desta solução.

Para os testes de azimuth, elevação e distância de Fraunhofer, foram efetuadas 30 medidas de 1 segundo de duração para cada valor real de angulação e distância. Foi calculada a média dos valores de ângulos preditos e a incerteza associada a média, conforme as equações 4.1 e 4.2, respectivamente:

$$\hat{\Theta} = \frac{1}{N} \sum_{j=1}^N \hat{\theta}_j \quad (4.1)$$

$$\sigma_{\hat{\Theta}} = \sqrt{\frac{1}{N(N-1)} \sum_{j=1}^N (\hat{\theta}_j - \hat{\Theta})^2} \quad (4.2)$$

O erros de predição foram calculados de maneira direta como:

$$Erro[\hat{\Theta}] = \hat{\Theta} - \theta \quad (4.3)$$

onde  $\theta$  é o ângulo real,  $\hat{\Theta}$  é o ângulo predito médio para aquela posição.

As mesmas equações apresentadas para o azimuth  $\theta$  valem para a elevação  $\phi$ . Além disso, também foram utilizadas da mesma maneira para calcular o tempo médio de execução do algoritmo, assim como a incerteza associada a essa média.

A resolução do experimento está associada a resolução de busca do algoritmo de TDOA-Semblance, isto é, o passo de varredura delta com que os ângulos testados são variados. Há também a resolução máxima dos gabaritos utilizados para determinar o valor real dos ângulos da fonte, que devido a construção mecânica, pode ser estimado como de 5°.

#### 4.4.2 Elementos suplementares

Para realizar a avaliação do desempenho do sistema construído foram elaborados gabaritos de medição. Foram construídos dois gabaritos, um para azimute sobre folha de papel cartão cobrindo de  $0^\circ$  até  $180^\circ$ , com passo de  $10^\circ$ , e outro para elevação, construído com arame e ripas de madeira, cobrindo de  $0^\circ$  até  $90^\circ$ , com passo de  $10^\circ$ . Na Figura 4.3 é apresentado o gabarito, assim como outros elementos suplementares para a realização dos testes. O suporte foi construído para permitir posicionamentos diversos da fonte sonora.

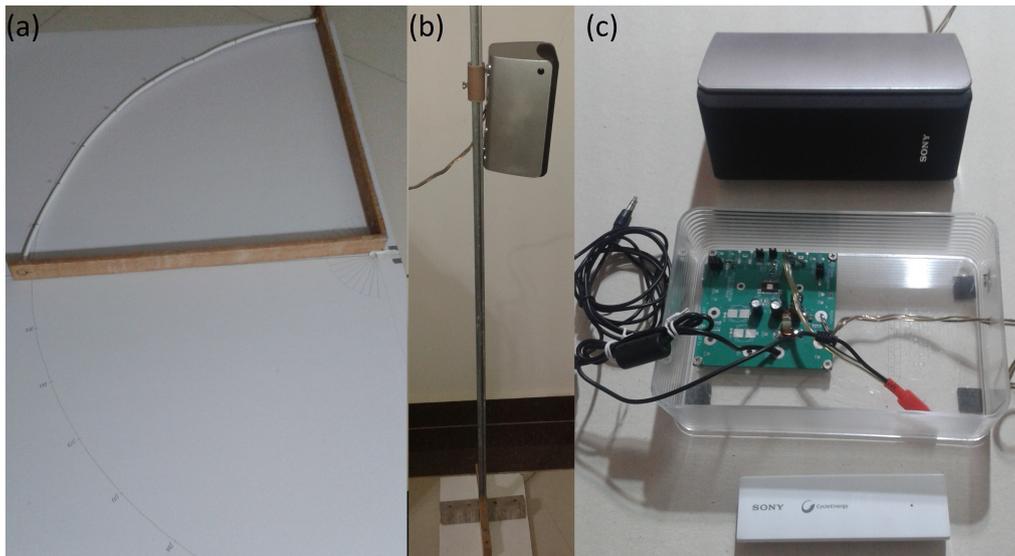


Figura 4.3: Elementos suplementares para os testes de avaliação. (a) Gabarito de azimute e elevação. (b) Suporte para alto-falante. (c) Sistema amplificador para alto-falante com entrada estéreo tipo P2.

O sistema amplificador e alto-falante (Figura 4.3 (c)) não foi desenvolvido neste trabalho, apenas sendo usado como fonte sonora durante os testes. Para o elementos de reprodução, foi utilizado um telefone celular acoplado a entrada P2 do sistema de amplificador.

## Capítulo 5

# Resultados e Discussão

### 5.1 Hardware

Foram desenvolvidos suportes para o hardware utilizado. Quatro suportes para o Raspberry Pi e quatro suportes para o Arduino Uno, no formato de cilindros. Um suporte especial foi feito para o sistema de servomotores utilizados para apontar um laserpointer para a direção predita pelo algoritmo de DOA. O conjunto de hardware utilizado e desenvolvido pode ser visto na Figura 5.1.

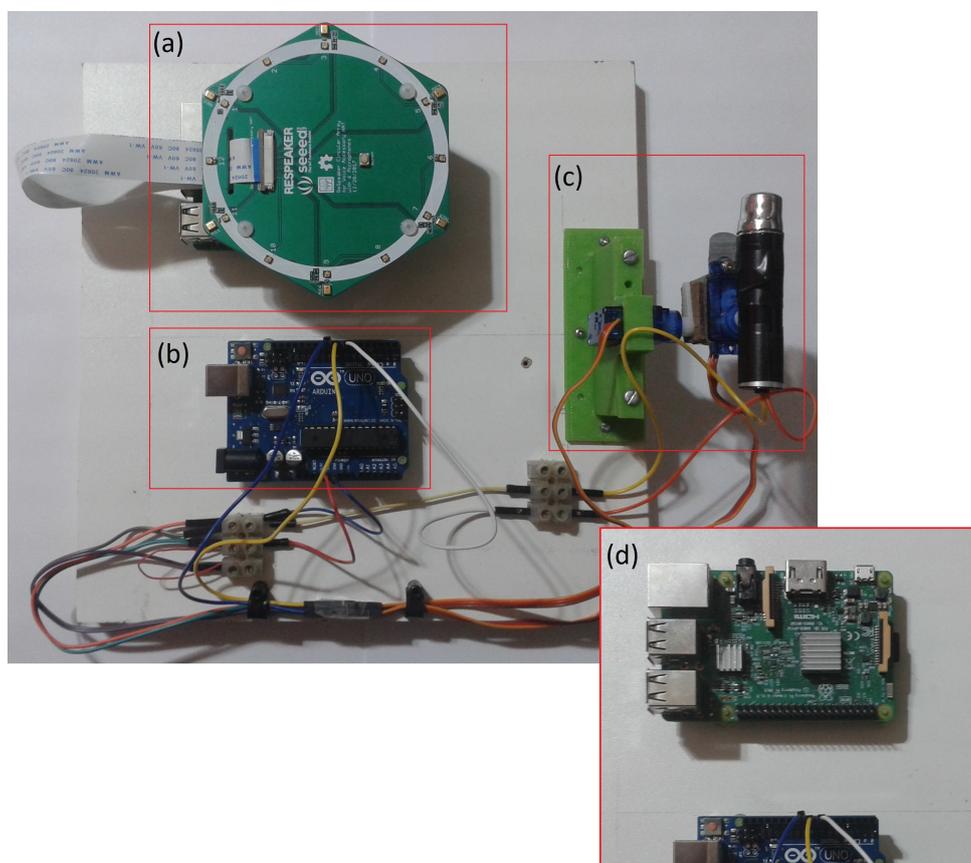


Figura 5.1: Hardware do sistema de DOA desenvolvido. (a) ReSpeaker 6-Mic Circular. (b) Arduino Uno. (c) Servomotores e laser. (d) Detalhe do Raspberry Pi, localizado abaixo do ReSpeaker.

Os suportes foram projetados utilizando o software SolidWorks, com os desenhos em anexo (Apêndice A). O projeto desenvolvido foi impresso em uma impressora 3D Finder Flashforge CX, usando plástico de poliácido láctico (PLA).

O laserpointer foi adaptado para que a tensão de alimentação fosse controlada pela saída digital do Arduino. Foi considerado tensão de alimentação em nível alto (5V) e nível baixo (0V), para ligar e desligar o laser, respectivamente.

Foram usados dois Micro Servo Motor SG90 9g para controle de azimute e elevação. O controle de ambos foi feito através das saídas digitais do Arduino. Foram utilizadas bibliotecas de software prontas, diretamente para Python. A forma de controle dos servo motores é através de Modulação por Largura de Pulso (do inglês, *Pulse Width Modulation* - PWM), controlando a alimentação entregue à carga (motores). Este tipo de servo apresenta uma oscilação (*jitter*) devido à modulação PWM.

Os fios de alimentação e terra dos servomotores e do laser foram conectados conjuntamente através de conectores barra, como pode ser visto na parte inferior da Figura 5.1. A alimentação do Arduino foi realizada através do cabo USB conectado diretamente ao Raspberry Pi.

Na Figura 5.2 são apresentados os erros de posicionamento do laser. O sistema construído foi alinhado ao gabarito para cada caso, testando-se a calibração para os ângulos extremos (cujo erro está zerado nos gráficos). Observa-se que há certa dependência angular do erro, que pode ter origem na construção mecânica do aparato e na precisão dos gabaritos.

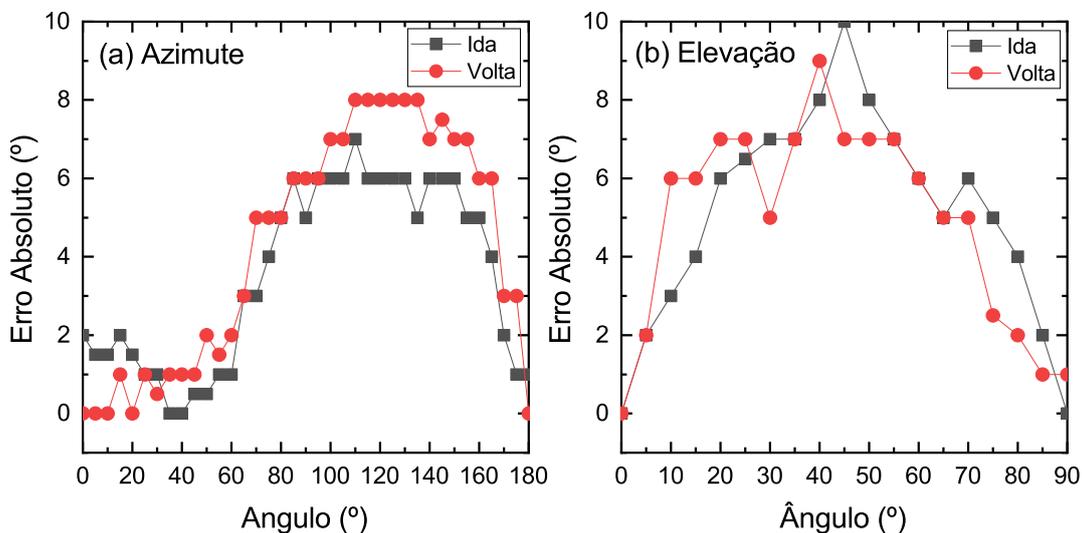


Figura 5.2: Erro angular para o posicionamento do laser.

Um dos principais problemas observados foi o desalinhamento dos eixos dos servo motores. Isso acrescentou um erro para angulações combinadas e dificultou o correto alinhamento entre o sistema e os gabaritos de teste. Assim, foram procuradas soluções para a construção mecânica. Uma das possíveis soluções é a utilização de um sistema *pan-tilt* como utilizado em câmeras de vigilância.

## 5.2 Software

Para a realização do DOA, foram implementadas funções que possibilitam o cálculo do Semblance. Estas funções foram baseadas na implementação proposta em [43]. No Apêndice B, *Algoritmo de DOA - Semblance* é apresentada a implementação dessas funções. Chamamos atenção para a função de  $z\_painel$  que estabelece o procedimento efetivo do DOA, calculando o valor de Semblance para diferentes valores de atraso dos sinais. Os valores de atraso são calculados de acordo com a posição dos microfones e para cada canal. Em seguida, é aplicado esse atraso e calculado o Semblance. Cada cálculo é feito em cima de um determinado valor angular, de forma que se estabelece um grid de busca, isto é, uma matriz onde cada elemento possui o valor de Semblance calculado e representa um par de coordenadas (azimute e elevação). O valor de Semblance é obtido através da Eq. 3.11. Na Figura 5.3 é apresentado um esquema do fluxo de processamento do algoritmo, com detalhe para as equações mais importantes usadas em cada etapa:

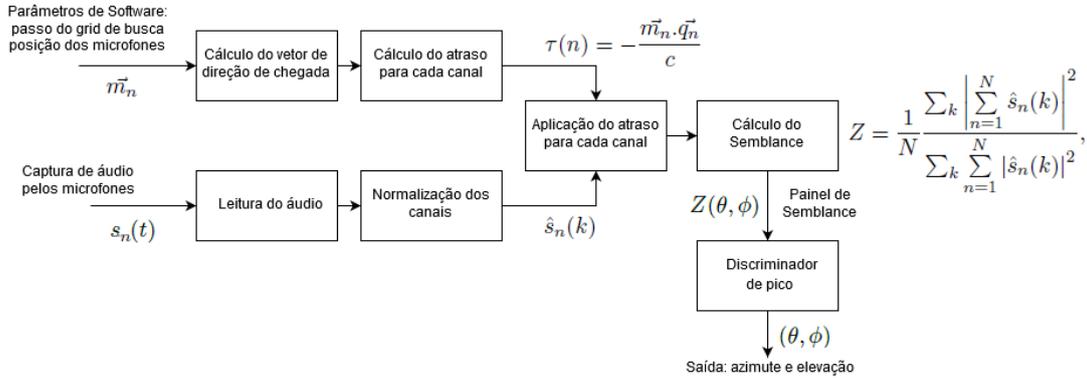


Figura 5.3: Esquema do algoritmo de Semblance implementado.

Aqui escolhemos o sistema de coordenadas para coincidir com o eixo de orientação do laser. Dessa forma, os ângulos preditos pelo algoritmo de DOA podem ser diretamente passados para o algoritmo de controle dos servos. Para a posição dos 6 microfones do Respeaker, utilizamos a seguinte matriz, onde cada linha representa um microfone, na ordem de 1 à 6, com distâncias da origem do sistema dadas em metros:

$$\vec{m} = \begin{pmatrix} -0.175 & 0.035 & 0.015 \\ -0.175 & 0.085 & 0.015 \\ -0.135 & 0.112 & 0.015 \\ -0.095 & 0.085 & 0.015 \\ -0.095 & 0.035 & 0.015 \\ -0.135 & 0.012 & 0.015 \end{pmatrix} \quad (5.1)$$

Contudo, devido algum defeito, o microfone 1 do Respeaker não estava funcionando. Dessa forma, no código, comentamos a primeira linha da matriz e excluímos o sinal captado por esse canal, antes de realizar a normalização.

O fluxo de processamento utilizando as funções de DOA e o estabelecimento dos sistemas de coordenadas pode ser visto Apêndice B, no código de *Estimacao do DOA*. O

algoritmo foi implementado de maneira a atuar sobre um áudio de duração de 1 segundo. Poderia-se utilizar outros valores para o tamanho do frame de áudio usado no DOA, por exemplo, de 200 milissegundos, calculando-se um valor de média ou máximo de detecção entre os painéis. Essas janelas de frame de áudio também podem ser aperfeiçoadas para detectar a movimentação de uma fonte sonora, conforme os ângulos preditos variam ao longo de uma sequência de frames.

Para a implementação do controle dos servos utilizamos uma biblioteca Python que realiza a compilação para o código de mais baixo nível a ser embarcado no Arduino. A implementação pode ser vista no Apêndice B, *Controle dos Servos*. Apesar da solução ser completamente funcional, como o código deve ser carregado a cada momento no Arduino, há um gasto de tempo excessivo (na ordem de 3 segundos). Para a solução tornar-se de tempo real, deve-se realizar uma alteração do código e do hardware, por exemplo, de maneira que o Arduino receba sinais de comando do Raspberry Pi através de algum dos seus pinos. Isso poderia ser feito com adaptação de um cabo USB.

Foram desenvolvidos alguns scripts complementares, como para a automatização dos testes, plotagem de alguns gráficos e para a realização do DOA e do direcionamento do laser em um pipeline contínuo. Para os testes e o laço de repetição do pipeline foram feitos scripts shell (.sh), apresentados no Apêndice B. Para o pipeline, foi feito um script conjunto do DOA e do controle dos Servos, em Python, também apresentado no Apêndice.

## 5.3 Desempenho

### 5.3.1 Demonstração de Funcionamento

A seguir apresentamos uma demonstração do funcionamento do algoritmo implementado:

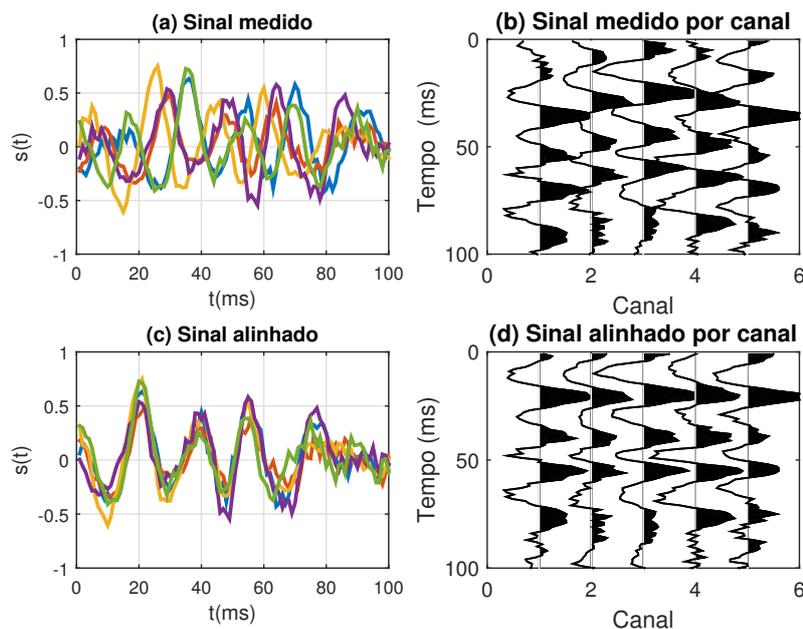


Figura 5.4: Exemplo de funcionamento do algoritmo de Semblance.

Quando o algoritmo corrige o atraso dos sinais de maneira a sincroniza-los, o valor de Semblance calculado é máximo. Isso permite estabelecer o atraso exato que corrige a direção de chegada, uma vez que cada atraso calculado depende apenas do produto escalar entre a posição dos microfones e a direção pressuposta. Dessa forma, a Figura 5.4 deixa bem claro essa importante etapa de sincronização estabelecida pelo máximo de Semblance.

### 5.3.2 Testes de Distância de Fraunhofer

Primeiramente, determinamos a condição de campo distante, através da Distância de Fraunhofer. O cálculo sugere uma distância limite de 38 cm para um sinal de 20 kHz, uma vez, que para frequências menores, a distância será menor. Os testes realizados confirmam essa condição, já que o ângulo estimado pelo algoritmo aproxima-se do ângulo real conforme a distância entre a fonte e a origem do sistema de coordenadas, aumenta, como pode ser visto na Figura 5.5, a seguir:

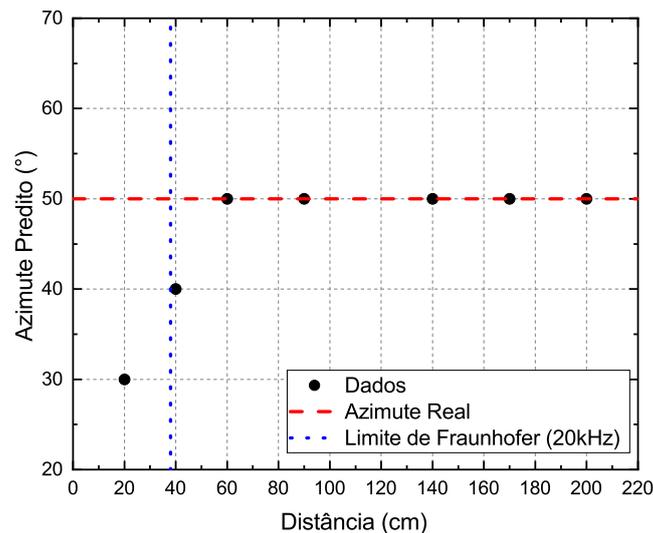


Figura 5.5: Teste da condição de Fraunhofer.

Para a execução desse experimento foi utilizada uma resolução fixa de  $\Delta = 10^\circ$ , com azimute e elevação constantes de  $50^\circ$  e  $10^\circ$ , respectivamente.

### 5.3.3 Testes de Azimute

Foram realizados testes para verificar a acurácia na determinação do ângulo de azimute. Para isso, a fonte sonora foi colocada em uma posição conhecida, emitindo uma música. Foram feitas 30 capturas da mesma música para uma mesma posição de azimute. O azimute foi então variado, mantendo-se a elevação constante. Foi plotada a relação entre o azimute real e o azimute predito pelo algoritmo, assim como o erro associado a essa medição. Também foi verificada a elevação predita em cada caso, assim como seu erro associado. Os resultados são apresentados na Figura 5.6, onde podemos ver que o algoritmo apresentou um comportamento linear na predição, aproximado-se da reta de relação direta.

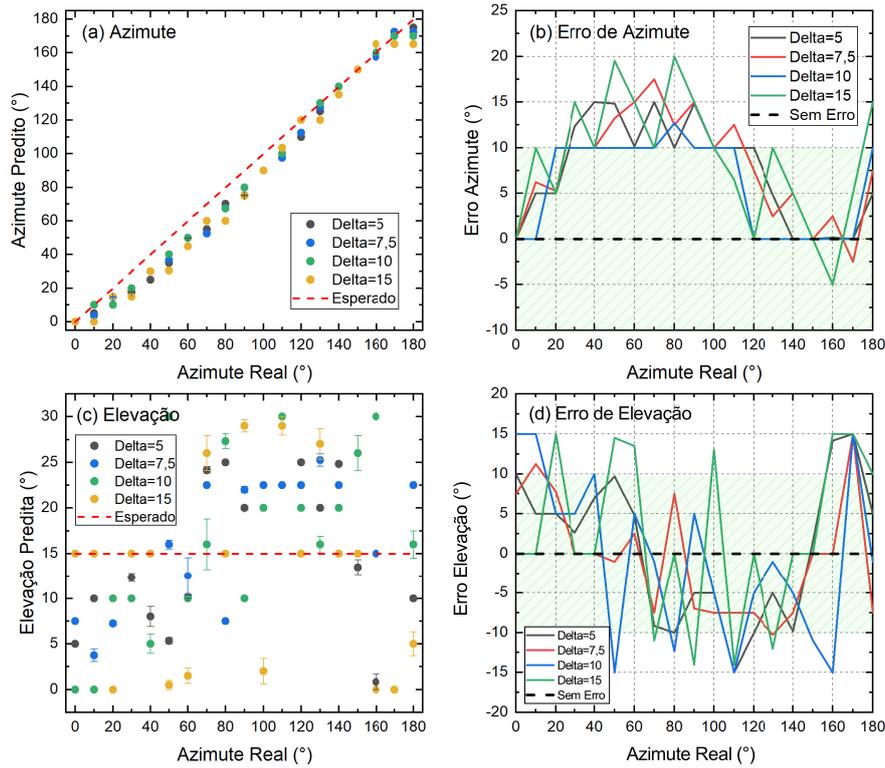


Figura 5.6: Acurácia no algoritmo para a variação de azimute.

Observa-se que o algoritmo consegue prever o azimute conforme ele é variado. O desvio observado (erro) para alguns ângulos pode indicar que, para essa configuração espacial, podem existir reflexões e sombras. A região hachurada em verde nos gráficos indica um erro aceitável de até  $10^\circ$ . Em geral, o erro manteve-se dentro do aceitável, tendo contudo um forte desvio, como já ressaltado, para a região que vai de  $40^\circ$  até  $100^\circ$ . É importante notar que, para um passo de  $\text{delta}=10^\circ$ , o erro se mantém dentro da faixa aceitável para mais casos que qualquer outra resolução. Um dos motivos das resoluções de 7.5 e 15 apresentarem alguns dos erros observados é que, como o ângulo real foi variado em passo de  $10^\circ$ , as medidas de ângulos invariavelmente apresentariam um determinado erro de resolução para determinados valores de ângulos que não fossem múltiplos exatos de 7,5 e 10.

Chamamos atenção para as Figuras 5.6 (a) e 5.7 (c) que demonstram visualmente que a solução aqui proposta consegue desempenhar a localização da fonte sonora, como pretendido.

### 5.3.4 Testes de Elevação

Repetimos os mesmos passos que para os testes de azimute, observando o mesmo tipo de comportamento para os dois casos. Os resultados são apresentados na Figura 5.7. Há, contudo, um erro de azimute crescente conforme a elevação real aproxima-se de  $90^\circ$ . Para esse valor exato há uma ambiguidade, uma vez que a composição  $\theta = 70^\circ$  e  $\phi = 90^\circ$  localiza o mesmo ponto que  $\theta = 0^\circ$  e  $\phi = 90^\circ$ , que pode explicar o erro encontrado. Esta ambiguidade existe pois todos os microfones formam um único plano no espaço. Se o

arranjo de microfones possuísse microfones fora do plano, essa ambiguidade seria resolvida.

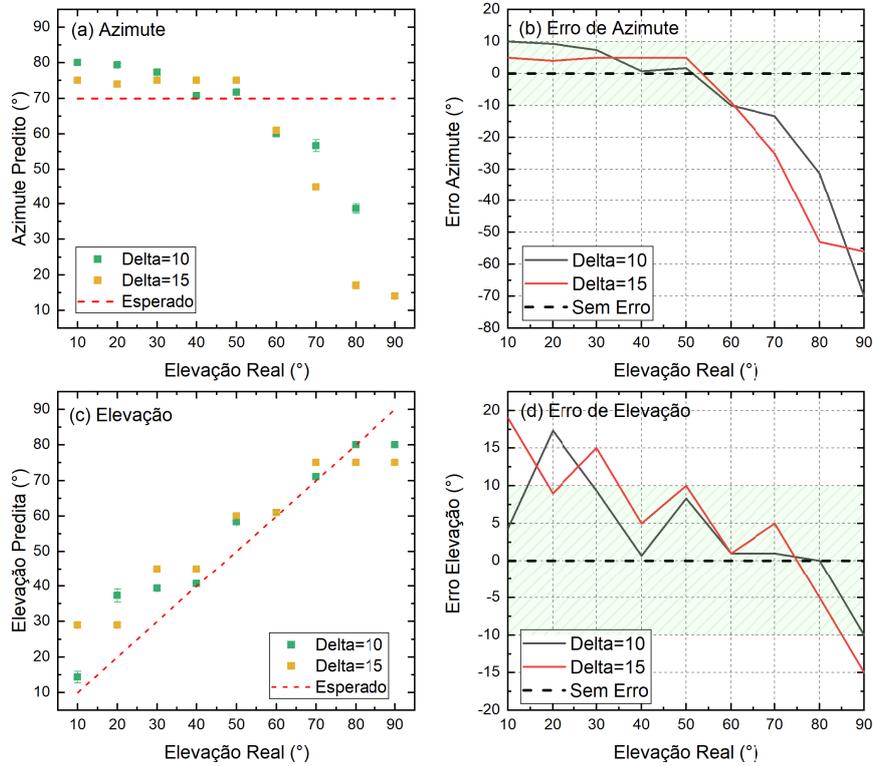


Figura 5.7: Acurácia no algoritmo para a variação de elevação.

### 5.3.5 Desempenho de Azimute e Elevação

Dos resultados anteriores podemos sintetizar o erro médio para cada resolução. Estes resultados são apresentados na Figura 5.8:

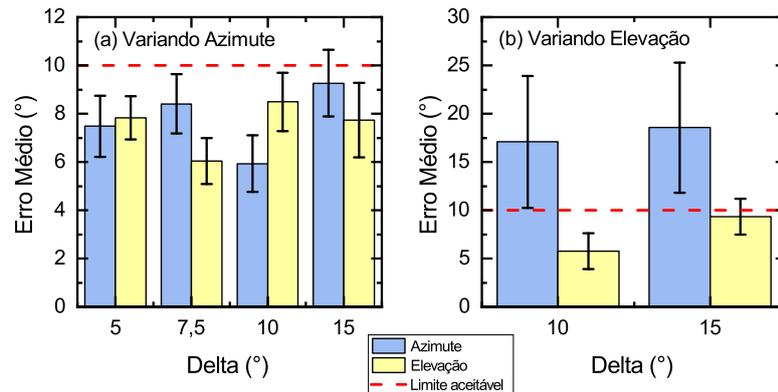


Figura 5.8: Erros médios para azimute e elevação.

Durante a varredura de azimute, o erro médio manteve-se abaixo do limite aceitável. Há, contudo, um erro médio muito grande de azimute durante a variação de elevação, que provavelmente está associado a ambiguidade quando o ângulo de elevação aproxima-se de  $90^\circ$  (ver Figura 5.7 (a) e (b)). Em ambos os casos o erro obtido é menor quando a

resolução do grid é igual ao passo de variação angular do experimento, para os casos em que se observa o mesmo ângulo que está sendo variado.

### 5.3.6 Desempenho de tempo

O tempo de execução do algoritmo de Semblance não se modificou para o ângulo real medido, como pode ser visto na Figura 5.9, e como era de se esperar. Há, contudo, forte dependência do passo delta.

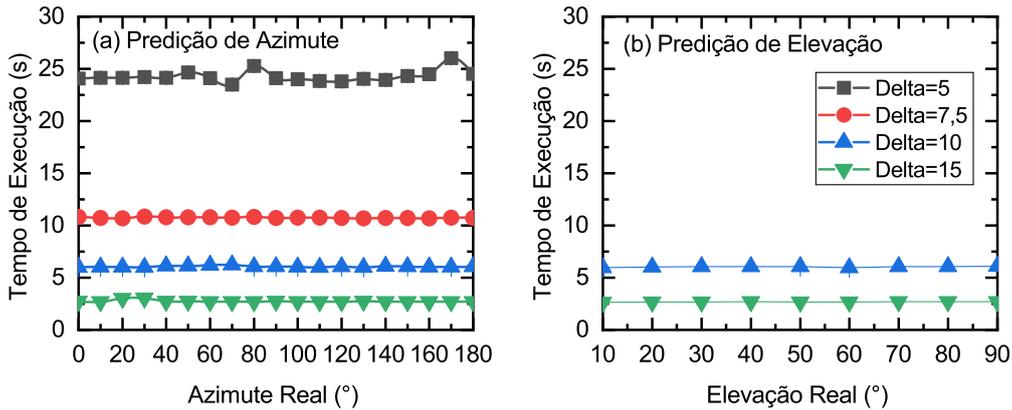


Figura 5.9: Tempo de execução do Semblance para diferentes ângulos de teste.

O tempo de execução do algoritmo de Semblance depende fortemente do tamanho do painel de Semblance, isto é, a quantidade de elementos (pares de ângulos) que são calculados durante a busca dos atrasos que sincronizam os sinais captados. Na Figura 5.10 são apresentados os tempos médios de execução para diferentes valores de passo para a varredura dos ângulos (parâmetro delta). Conforme a resolução do grid aumenta (diminuição de delta), o tempo de execução aumenta, como pode ser visto na Figura 5.10 (a). Da mesma forma, conforme a resolução aumenta, aumenta-se o número de elementos varridos (calculados o atraso). O grid aumenta de maneira quadrática com a resolução delta (já que estamos varrendo uma matriz  $N \times N$ ), e de maneira linear com a quantidade de elementos  $EL = N^2$ , como mostrado em (b).

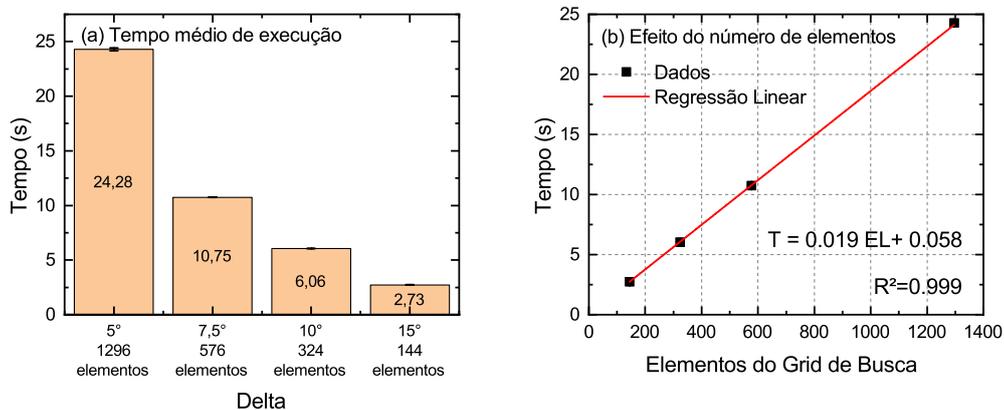


Figura 5.10: Tempo de execução médio do algoritmo de Semblance.

A regressão linear realizada nos dados experimentais, como ilustrada pela Figura 5.10 (b) sugere que cada operação de atraso aplicada aos canais, para cada elemento do grid de busca dos ângulos, leva, em média, 19 ms para ser executada no Raspberry Pi. A aplicação está rodando em cima do sistema operacional, ou seja, competindo tempo de execução com outros processos no escalonador, o que pode influenciar diretamente nesse tempo observado. Há um gasto constante de 56 ms que pode estar relacionado à outras etapas do algoritmo, como leitura do áudio e das posições dos microfones (carregamento na memória do programa).

Os resultados demonstram que, da forma como o algoritmo foi implementado, ele não age como uma aplicação de tempo real. Várias melhorias são sugeridas na Seção de Perspectivas Futuras, 6.1, para corrigir isso.

### 5.3.7 Teste com Múltiplas Fontes

O algoritmo de Semblance implementado permite a localização de múltiplas fontes. Na Figura 5.11 é apresentado um exemplo com dois alto-falantes localizando em posições diferentes e emitindo áudios (música) distintos. Foi usada uma resolução de  $\Delta = 5^\circ$  e o valor de Semblance  $Z(\theta, \phi)$  foi normalizado em relação ao valor máximo para estabelecer uma escala de 0 à 1. Podemos ver que o algoritmo consegue identificar duas regiões para os ângulos de direção de chegada, relacionadas às posições reais das fontes.

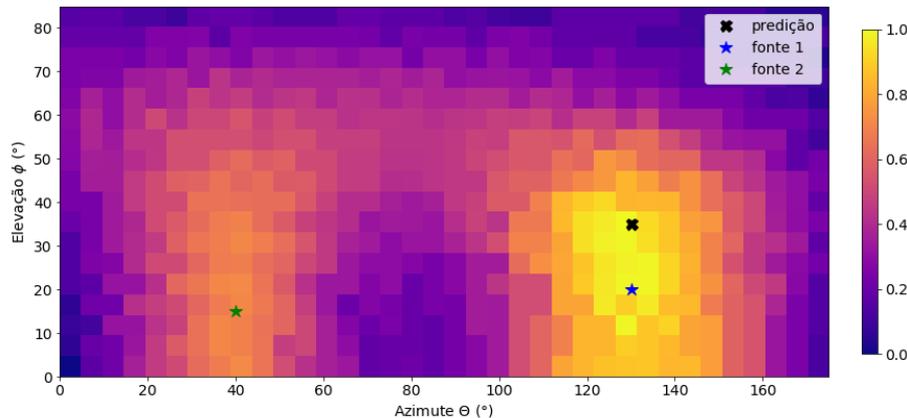


Figura 5.11: Painel de Semblance para a localização de múltiplas fontes.

Um dos problemas é o volume sonoro que cada fonte exerce sobre o arranjo de microfones. Como a amplitude do sinal gravado pelos microfones é normalizada, o sinal de maior volume sonoro apresentará uma energia relativa maior do que o outro sinal, de forma que o método de Semblance favorece as correções de atraso (sincronização dos canais de áudio) para a configuração de maior energia. Dessa forma, a fonte de maior volume apresenta maior valor de Semblance no painel. O volume sonoro pode estar associado a potência de emissão da fonte e a distância que ela se encontra do arranjo. Dependendo do tipo de fonte, isso pode ser interpretado como um ruído, um sinal não desejado que traz dificuldades para o discriminador acertar a fonte desejada (sinal de interesse, SOI). Quando se sabe que o cenário de aplicação possui fontes de ruído, é possível utilizar técnicas para minimizar a

sua influência na detecção.

No nosso caso, o discriminador de pico está configurado para detectar apenas um pico, de maneira que o algoritmo prediz apenas a localização de uma das fontes. Isso pode ser aperfeiçoado para adequar o algoritmo para a predição de mais fontes. Um dos desafios, neste caso, é estabelecer um critério de clusterização dos elementos do painel de Semblance que possibilite isolar os pontos que representam uma única fonte. Além disso, fontes muito próximas podem não ser separáveis, entre outras coisas, devido ao nível sonoro que pode ser detectado de cada uma delas.

## Capítulo 6

# Considerações Finais

### 6.1 Perspectivas Futuras

A execução deste trabalho possibilitou o aprofundamento de conceitos de detecção de fontes sonoras, especialmente os aspectos práticos da programação e construção de um sistema capaz de fazer isto. Uma série de desafios e aprimoramentos podem ser elencados:

- **Apontamento do laser:** pode-se utilizar um sistema mecânico comercial de *pan-tilt* para realizar o apontamento do laser de maneira mais precisa e acurada, corrigindo a defasagem entre os eixos de rotação. Servo-motores mais robustos podem ser utilizados para evitar o *jitter* e possibilitar uma varredura em  $360^\circ$  para o ângulo de azimute.
- **Bufferização do áudio:** o áudio captado pelo sistema de microfones pode ser bufferizado para permitir um funcionamento mais próximo de uma aplicação de tempo real, executando o algoritmo de *Semblance* para frames menores que 1 segundo de duração. Isto pode ser feito de maneira paralela, uma vez que enquanto o sistema captura o próximo áudio (preenche o buffer), o buffer anterior alimenta o algoritmo de *Semblance*.
- **Taxa de amostragem:** pode-se verificar o desempenho da aplicação para taxas de amostragem diferentes. Apesar do *Semblance* funcionar melhor para taxas de amostragens grandes, pode-se encontrar um equilíbrio entre uma taxa de amostragem que traz bons resultados e uma que aumenta o desempenho. Isso ocorre, uma vez que o cálculo de atraso (sincronização dos canais) opera em cima de mais elementos quando a taxa de amostragem é maior. Uma diminuição da taxa de amostragem pode diminuir o número de cálculos efetuados durante essa etapa (que repete-se para cada elemento do grid de busca).
- **Processamento paralelo:** além do processamento paralelo de bufferização, pode-se imaginar formas de paralelizar as etapas do próprio algoritmo de *Semblance* para se ganhar desempenho.
- **Programação em baixo nível:** outra linha de ataque para tornar a aplicação em tempo real é migrar a programação do Python para uma linguagem de mais baixo

nível, inclusive eliminando-se processos de sistema operacional do Raspberry Pi e aproximando-se de um sistema *bare metal*.

- **Aperfeiçoamento da detecção de múltiplas fontes:** novos discriminadores de pico podem ser estabelecidos para a detecção de múltiplas fontes. Pode-se estabelecer critérios para determinar regiões do grid de forma mais compacta. O algoritmo de Beamforming pode ser usado para aplicar ganho em determinadas regiões preditas e estabelecer uma forma de seleção do áudio de cada fonte.
- **Robustez à ruídos:** aqui não tratamos os efeitos de ruído sobre o algoritmo de Semblance. Estes efeitos podem ter alto impacto no desempenho da solução. Filtros de Wiener e algoritmos de aprendizado de máquina podem ser usados dentro do pipeline do Semblance para limpar o ruído dos áudios captados. Estes aperfeiçoamentos podem ser importantes, ou até imprescindíveis, para alguns cenários de aplicação.
- **Detecção de fontes em movimento e *Tracking*:** a utilização de frames (janelamento) dos áudios permite encontrar ângulos para cada frame. Se estes ângulos estiverem variando de uma determinada forma, pode ser um forte indicativo de movimento da fonte sonora. Isso permite estabelecer um critério para fazer o rastreamento (*tracking*) da fonte, utilizando um filtro de predição ou um modelo cinemático.

## 6.2 Conclusão

Neste trabalho fizemos uma revisão dos principais conceitos de Localização de Fontes Sonoras, especialmente no quesito de Detecção de Direção de Chegada. Aqui trabalhamos com um algoritmo baseado em TDOA por medida de coerência entre sinais captados por um arranjo de microfones (sinal multicanal). O algoritmo foi implementado e testado em um computador de placa única, mostrando-se eficaz na predição da direção das fontes sonoras. Além disso, apresentamos resultados do seu desempenho de tempo de execução, assim como os parâmetros associados à esse tempo. Evidentemente, uma resolução maior para a predição consome mais tempo de execução, mas, não tão evidentemente, nem sempre retorna uma melhor estimativa. Isso pode estar associado à uma série de problemas, entre eles, a amostragem espacial do experimento e do algoritmo.

Apesar do algoritmo usado ser uma adaptação de algoritmos utilizados em sísmica de reflexão, os resultados obtidos demonstram que sua utilização é válida. Aplicando-se o critério de um erro de até  $10^\circ$ , o algoritmo mostrou-se bastante assertivo. Há contudo detalhes de ambiguidade, como aquelas causadas por composição de ângulo de azimute e elevação diferentes, que referenciam à mesma posição no espaço. Ao considerar-se o baixo custo dos itens utilizados e as possibilidades de melhorias elencadas, podemos concluir que a solução implementada aqui, alcançou os objetivos propostos.

De maneira geral, este trabalho serve como introdução a área de localização de fontes e processamento de áudio, instruindo para a construção e programação de sistemas capazes de realizar esta tarefa. Diversas linhas de aperfeiçoamento e continuidade de estudos são apresentadas, demonstrando que o assunto aqui tratado é bastante vasto e promissor.

# Referências Bibliográficas

- [1] M. Lindholm, M. Eronen, M. Timonen, J. Meriläinen, *A ring-width chronology of Scots pine from northern Lapland covering the last two millennia*. Ann. Bot. Fennici 36:119-126, ISSN 0003-3847, (1999).
- [2] O. Kasum, A. Perović, A. Jovanović, *Measures and Metrics of Biological Signals*. Front. Physiology 9:1707. doi: 10.3389/fphys.2018.01707, (2018).
- [3] X. S. Zhang, F. Wang, *Signal Processing for Finance, Economics and Marketing: concepts, framework, and big data applications*. IEEE Signal Processing Magazine, doi: 10.1109/MSP.2017.2663138, (2017).
- [4] I. Palmieri, *Modelagem de sinais neuronais utilizando filtros lineares de tempo discreto*. Escola Politécnica, Universidade de São Paulo, USP (2015).
- [5] B. P. Lathi, *Sinais e Sistemas Lineares*. Bookman, 2 edição (2007).
- [6] M. Strauss, P. Mordel, V. Miguet, A. Deleforge, *DREGON: Dataset and Methods for UAV-Embedded Sound Source Localization*. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Madrid, Spain. IEEE, pp.5735-5742, doi: 10.1109/IROS.2018.8593581, hal:01854878, (2018).
- [7] D. Iglesias, A. Matriss, Akin Tatoglu, E. A. King, *Improving autonomous vehicle safety—Using acoustic source localization to influence the decision making capabilities of an autonomous vehicle*. The Journal of the Acoustical Society of America 145, 1733; doi: 10.1121/1.5101365, (2019).
- [8] K. Nose Filho, *Desconvolução e separação cega de sinais esparsos e aplicações em sísmica de reflexão*. Universidade Estadual de Campinas, UNICAMP (2015).
- [9] T. Barros, R. Lopes, M. Tygel *Implementation aspects of eigendecomposition-based high-resolution velocity spectra*. Geophysical Prospecting, 63, 99–115 doi: 10.1111/1365-2478.12164, (2015).
- [10] D. Morin, *Waves*. Department of Physics, Harvard University, (2010). Disponível em: <http://www.people.fas.harvard.edu/~djmorin/book.html>. Acesso em: 18/08/2019.
- [11] S. W. Rienstra e A. Hirschberg, *An Introduction to Acoustics*. Eindhoven University of Technology (2019).

- [12] K. E. Gilbert e H. E. Bass, *Acoustic Waves*. Encyclopedia of Atmospheric Sciences, (2002).
- [13] D. A. Russell, *Graduate Program in Acoustics*. Pennsylvania State University, (2001). Disponível em: <https://www.acs.psu.edu/drussell/Demos/rad2/mdq.html>. Acesso em 20/08/2019.
- [14] Representação tridimensional de uma onda acústica. Imagem adaptada de: [https://www.cymascope.com/cyma\\_research/physics.html](https://www.cymascope.com/cyma_research/physics.html). Acesso em 20/08/2019.
- [15] A. Plumb, *Dynamics of the Atmosphere: dispersive and nondispersive waves*. Disponível em: [http://www-eaps.mit.edu/~rap/courses/12333\\_notes/A3%20waves.pdf](http://www-eaps.mit.edu/~rap/courses/12333_notes/A3%20waves.pdf). Acesso em 20/08/2019.
- [16] B. S. Finn, *Laplace and the Speed of Sound*, ISIS, VOL. 55, No. 179, (1964).
- [17] M. J. Blandamer e J. C. R. Reis, *Newton-Laplace Equation*. A Notebook for Topics in Thermodynamics of Solutions and Liquid Mixtures, University of Leicester, (2004). Disponível em <https://www.le.ac.uk/chemistry/thermodynamics/pdfs/3000/Topic2540.pdf>. Acesso em 25/08/2019.
- [18] Moduli Adiabatic Bulk. Disponível em <http://web.mit.edu/8.03-esg/watkins/8.03/bmod.pdf>. Acesso em 30/11/2019.
- [19] S. W. Rienstra, A. Hirschberg *An Introduction to Acoustics*. Eindhoven University of Technology, (2019).
- [20] D. A. Russell, J. P. Titlow, Y. Bommen, *Acoustic monopoles, dipoles, and quadrupoles: An experiment revisited*. Am. J. Phys. 67 (8), (1999).
- [21] C. R. Nave, *Inverse Square Law, Sound*. Georgia State University, (2016). Disponível em: <http://hyperphysics.phy-astr.gsu.edu/hbase/Acoustic/invsqs.html>. Acesso em 26/08/2019.
- [22] M. Carley, *Some notes on Acoustic*. Narcissus Marsh, 1683/4, Phil. Trans. Roy. Soc. Lond., 156:472–486.
- [23] Microphone Handbook, National Instruments, (2019). Disponível em <https://www.ni.com/pt-br/innovations/white-papers/08/microphone-handbook.html>. Acesso em 10/11/2019.
- [24] H. Robjohns, *A Brief History of Microphones*, (201). Disponível em: <http://www.microphone-data.com/media/filestore/articles/History-10.pdf>. Acesso em 10/11/2019.
- [25] J. Lewis, *Analog and Digital MEMS Microphone Design Considerations*, Analog Devices (2013).

- [26] Propriedades direcionais de microfones. Disponível em: <https://www.mediacollege.com/audio/microphones/directional-characteristics.html>. Acesso em 10/11/2019.
- [27] J. M. Moses, K. P. Thout, *A Simple Laser Microphone for Classroom Demonstration*, The Physics Teacher, Vol. 44, (2006).
- [28] D. Garthe, *A Fiber-optic Microphone*. Sensors and Actuators A, (1991).
- [29] AES5-2018, *AES recommended practice for professional digital audio — Preferred sampling frequencies for applications employing pulse-code modulation*. Audio Engineering Society (2018).
- [30] J. Benesty, J. Chen, Y. Huang, *Microphone Array Signal Processing*. Springer, Vol 1, (2008).
- [31] V. Ormenesse, *Construção de um sistema para estimação de direção de chegada de sinais acústicos*. Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas, Universidade Federal do ABC, UFABC (2015).
- [32] L. C. F. Nogueira, *Sistema de Localização de Múltiplas Fontes Sonoras em Recintos Fechados Baseados em Algoritmos de Separação Cega de Fontes*. Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa em Engenharia, Universidade Federal do Rio de Janeiro, UFRJ (2015).
- [33] A. Horneffer, *Beamforming (and LOFAR)*, Max-Planck-Institut für Radioastronomie, (2012). Disponível em [https://www2.physik.uni-bielefeld.de/fileadmin/user\\_upload/workshops/glow/Talks/Horneffer-Beamforming\\_neu\\_.pdf](https://www2.physik.uni-bielefeld.de/fileadmin/user_upload/workshops/glow/Talks/Horneffer-Beamforming_neu_.pdf). Acesso em 25/08/2019.
- [34] T. Benedek, P. Tóth, *Beamforming measurements of an axial fan in an industrial environment*. Peridica Polytechnica Mechanical Engineering, 57(2), pp. 37-46, doi: 10.3311/PPme.7043, (2013).
- [35] E. G. dos Santos, *Identificação de Fontes Sonoras em Ambientes Reverberantes Utilizando Beamforming*. Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa em Engenharia, Universidade Federal do Rio de Janeiro, UFRJ (2015).
- [36] L. A. P. F. de Brito, *A utilização de mapas acústicos como ferramenta de identificação do excesso de ruído em áreas urbanas*. Eng. Sanit. Ambient, vol.22, n.6 (2017).
- [37] I. C. M. Guedes, S. R. Bertoli, *Mapa Acústico como Ferramenta de Avaliação de Ruído de Tráfego Veicular em Aracaju - Brasil*. PARC Research in Architecture and Construction, vol. 5, n.2 (2015).
- [38] R. R. Santin, *Localização de Fontes Sonoras em Recintos Fechados Aplicada a Televigilância Médica*. Universidade Federal de Sergipe, UFS (2012).

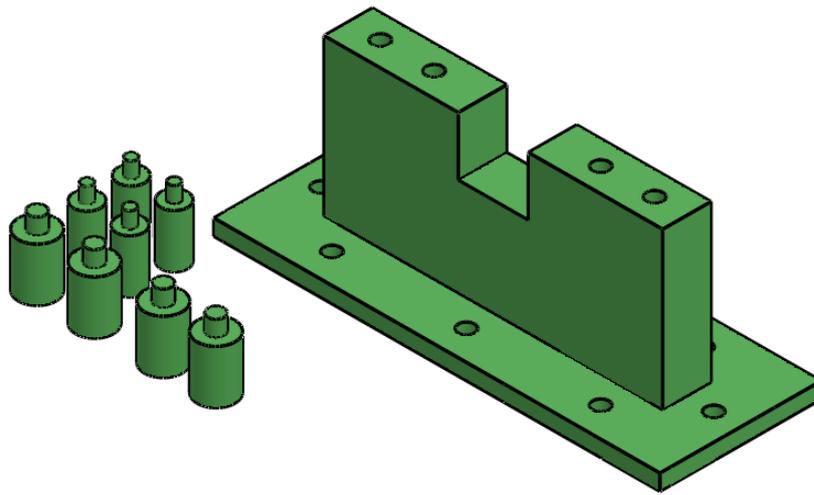
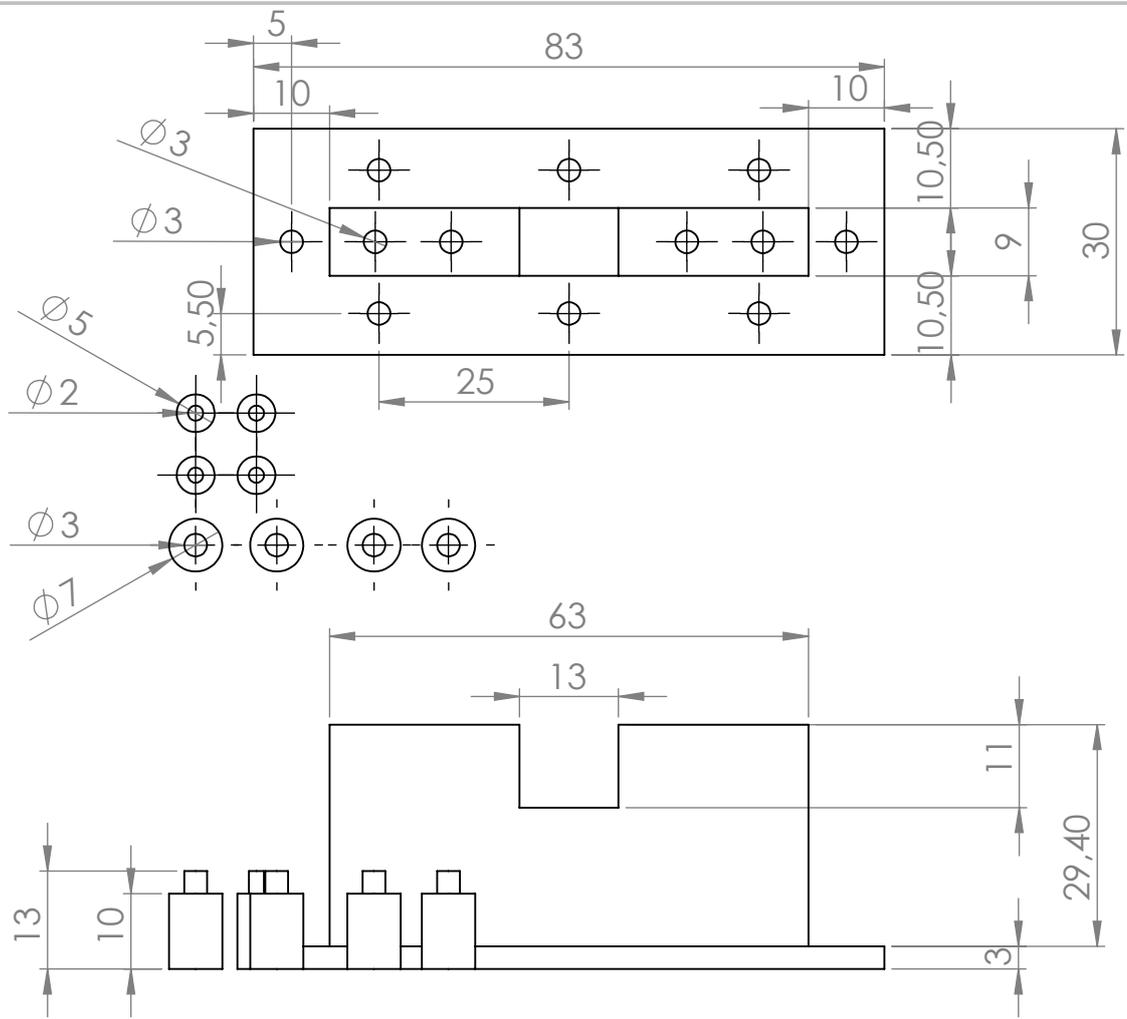
- [39] S. V. Bueno, R. J. L. Netto, E. Z. Nadalin, W. Pavan, R. Attux, J. M. T. Romano, *Alguns Aspectos do Problema de Localização de Fontes Acústicas*. IV EADCA - Quarto Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial, Universidade Estadual de Campinas, UNICAMP (2011).
- [40] S. Argentieri, P. Danes, P. Soueres, *A Survey on Sound Source Localization in Robotics: from Binaural to Array Processing Methods*. Computer Speech and Language, Elsevier, 34 (1), pp. 87-112. hal-01058575, (2015).
- [41] N. S. Neidell, M. T. Taner, *Semblance and other Coherency Measures for Multichannel Data*. Geophysics, Vol. 36, No. 3 (1971).
- [42] C. Knapp, G. Carter, *The generalized correlation method for estimation of time delay*. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, no. 4, pp. 320-327, (1976).
- [43] G. S. I. Aldeia, A. E. Crispim, G. Barreto, K. Alves, H. Ferreira, K. Nose-Filho, *textitA Semblance based TDOA algorithm for Sound Source Localization*. XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, (2019).
- [44] T. Spadini, G. S. I. Aldeia, G. Barreto, K. Alves, H. Ferreira, R. Suyama, K. Nose-Filho, *On the application of SEGAN for the attenuation of the ego-noise in the speech sound source localization problem*. 4th Workshop on Communication Networks and Power Systems, WCNPS (2019).
- [45] R. O. Schmidt, *Multiple Emitter Location and Signal Parameter Estimation*. IEEE Transaction on Antennas and Propagation, vol 34, No 3, (1986).
- [46] Especificações Raspberry Pi 3, Modelo B. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em 20/08/2019.
- [47] Arranjo de 6 microfones ReSpeaker para Raspberry Pi 3. Disponível em: [https://respeaker.io/6\\_mic\\_array/](https://respeaker.io/6_mic_array/). Acesso em 20/08/2019.
- [48] Datasheet Conversor Analógico/Digital AC108. Disponível em: [https://cdn.sparkfun.com/assets/6/0/d/c/a/AC108\\_Datasheet\\_V1.1.pdf](https://cdn.sparkfun.com/assets/6/0/d/c/a/AC108_Datasheet_V1.1.pdf). Acesso em 30/10/2019.
- [49] Datasheet Microfones MSM321A3729H9CP. Disponível em: <http://www.memsensing.com/webupfile/1531103042534785279.0-ENG.pdf>. Acesso em 11/11/2019.
- [50] Especificações Arduino Uno. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>. Acesso em 11/11/2019.
- [51] Linguagem de programação Python. Disponível em: <https://www.python.org/>. Acesso em 20/08/2019.

- [52] Sistema operacional Rasbian. Disponível em: <https://www.raspberrypi.org/downloads/raspbian/>. Acesso em 12/11/2019.
- [53] Tutorial de utilização ReSpeaker. Disponível em: [http://wiki.seeedstudio.com/ReSpeaker\\_6-Mic\\_Circular\\_Array\\_kit\\_for\\_Raspberry\\_Pi/](http://wiki.seeedstudio.com/ReSpeaker_6-Mic_Circular_Array_kit_for_Raspberry_Pi/). Acesso em 20/08/2019.



## Apêndice A:

Desenhos de projeto dos suportes



SE NÃO ESPECIFICADO:  
DIMENSÕES EM MILÍMETROS  
ACABAM. SUPERFÍCIE:  
TOLERÂNCIAS:  
LINEAR:  
ANGULAR:

ACABAMENTO:

DEBUR AND  
BREAK SHARP  
EDGES

NÃO MUDAR A ESCALA DO DESENHO

REVISÃO

	NOME	ASSINATURA	DATA		
DES.					
VERIF.					
APROV.					
MANUF					
QUALID					
				MATERIAL:	
				PESO:	

TÍTULO:

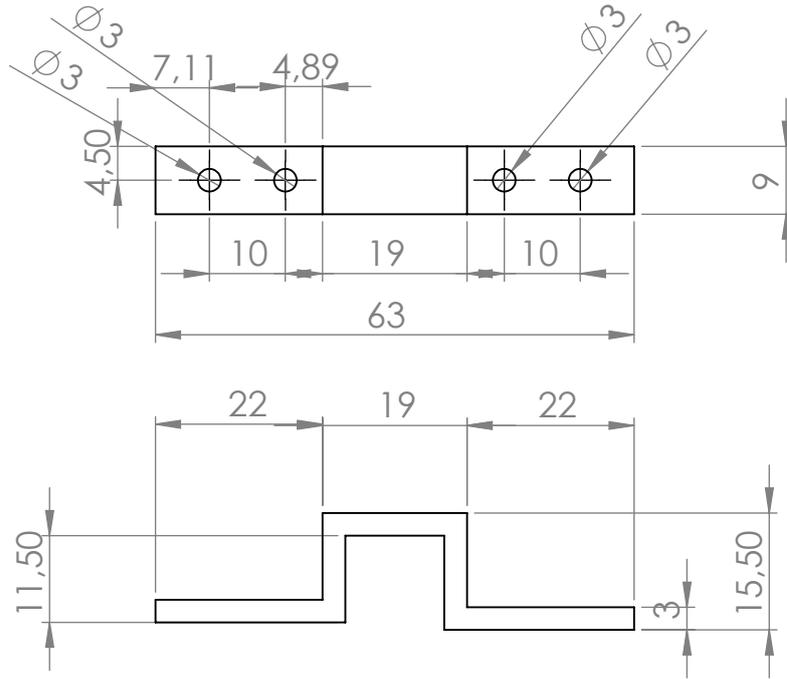
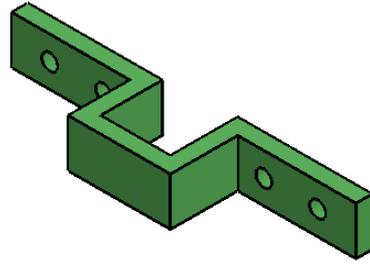
DES. Nº

suportes

A4

ESCALA: 1:2

FOLHA 1 DE 1



SE NÃO ESPECIFICADO:  
DIMENSÕES EM MILÍMETROS  
ACABAM. SUPERFÍCIE:  
TOLERÂNCIAS:  
LINEAR:  
ANGULAR:

ACABAMENTO:

DEBUR AND  
BREAK SHARP  
EDGES

NÃO MUDAR A ESCALA DO DESENHO

REVISÃO

	NOME	ASSINATURA	DATA		
DES.					
VERIF.					
APROV.					
MANUF.					
QUALID.					
				MATERIAL:	
				PESO:	

TÍTULO:

DES. Nº

bracadeira

A4

ESCALA: 1:2

FOLHA 1 DE 1

Apêndice B:

Códigos desenvolvidos

```

1 # H. Ferreira, 2019
2 # Algoritmo baseado na implementacao
3 # de G. Aldeia e K. Nose-Filho, referencia [42]
4
5 # -----
6 # ----- Algoritmo de DOA - Semblance -----
7 # -----
8
9 # Bibliotecas
10 import numpy as np
11
12 # Recebe os SINAIS (x) e as tentativas de estimacao dos atrasos (lags) de cada canal
13 def delay(x,lag):
14
15     N, M = len(x), len(x[0])
16     y = np.zeros((N,M))
17
18     for i in range(0,M): #Percorre o vetor de lag
19         if lag[i] > 0:
20             y[:, i] = np.concatenate((np.zeros(lag[i]), x[:-lag[i],i]), axis=0)
21         else:
22             y[:, i] = np.concatenate((x[-1*lag[i]:,i], np.zeros(-1*lag[i])), axis=0)
23
24     return(y)
25
26 # Calcula o valor de Semblance para cada posicao do grid de varredura angular
27 # Compondo um painel (matriz) com os valores para cada elemento de varrido
28 def z_painel(delta, micPos, speedOfSound, Fs, x):
29
30     theta = np.arange(0, np.pi, delta*np.pi/180.0)
31     phi = np.arange(0, np.pi/2, delta*np.pi/180.0)
32     numMic = len(micPos)
33
34     kd = np.zeros( (len(theta), len(phi), 3) ) # Vetor de chegada
35     tau = np.zeros((numMic)) # Atraso de chegada
36     z = np.zeros( (len(phi), len(theta)) ) # Medida de Semblance
37
38     for i in range(0,len(theta)):
39         for j in range(0,len(phi)):
40             # Mudanca de coordenadas para sistema cartesiano
41             kd = [np.cos(theta[i])*np.cos(phi[j]), np.sin(theta[i])*np.cos(phi[j]),
42                 np.sin(phi[j])]
43             for k in range(0,numMic):
44                 tau[k] = round(Fs*np.inner(kd, micPos[k,:])/speedOfSound) #
45                 # Determinacao dos atrasos (Eq 3.10, pag 34)
46             y = delay(x, tau.astype(int)) # Aplicacao dos atrasos em cada sinal
47             # (canal)
48             zs = z_semlance(y) # Medida de Semblance
49             z[j,i] = zs # Composicao do painel de Semblance
50             # (grid de varredura angular)
51
52     return(z)
53
54 # Determina o pico do valor de Semblance na matriz de varredura angular (painel de
55 # semblance)
56 def doa_prediction(semblance_painel, delta):
57     theta = np.arange(0, np.pi, delta*np.pi/180.0)
58     phi = np.arange(0, np.pi/2, delta*np.pi/180.0)
59
60     final_semblance_painel = semblance_painel.copy()
61     x, y = 0, 0
62     for i, _ in enumerate(phi):
63         for j, _ in enumerate(theta):
64             if (final_semblance_painel[i][j] > final_semblance_painel[x][y]):
65                 x, y = i, j
66
67     pred_theta, pred_phi = np.degrees( (theta[y], phi[x]) )
68
69     return([pred_theta,pred_phi])
70
71 # Normalizacao as amplitudes dos sinais
72 def normalize(x):

```

```
68
69     N, M = len(x), len(x[0])
70     y = np.zeros((N,M))
71
72     for i in range(0, M):
73         y[:,i] = x[:,i]/np.linalg.norm(x[:,i], ord=2)
74
75     return(y)
76
77 # Calcular valor de Semblance (Eq 3.9, pag34)
78 def z_semlance(x):
79
80     num = np.sum(np.square(np.sum(x,axis=1)))
81     den = np.sum(np.square(x))
82     z = num/den
83
84     return(z)
```

```

1 # H. Ferreira, 2019
2
3 # -----
4 # -----Estimacao do DOA-----
5 # -----
6
7 # Bibliotecas
8 import semblance
9 import wavio
10 import numpy as np
11 import sys
12 import time
13
14 inicio = time.time() # Macarcao de tempo de execucao
15
16 # Entrada via argumento
17 delta = float(sys.argv[1]) # Passo do gride de angulos
18 file = sys.argv[2] # Entrada do arquivo de audio
19
20 # Leitura do arquivo de audio
21 wav = wavio.read(file) # Leitura do arquivo de audio
22 Fs = wav.rate # Taxa de amostragem
23
24 x = semblance.normalize(wav.data[:,1:6]) # Normaliza os canais
25
26 # Parametros do DOA - Semblance
27 #delta = 10 # Passo do gride de angulos [graus]
28 speed_of_sound = 342 # Velocidade do som [m/s]
29
30 # Posicao dos microfones
31 micPos = np.array([
32     #(-0.175, 0.035, 0.015), # mic 1
33     (-0.175, 0.085, 0.015), # mic 2
34     (-0.135, 0.112, 0.015), # mic 3
35     (-0.095, 0.085, 0.015), # mic 4
36     (-0.095, 0.035, 0.015), # mic 5
37     (-0.135, 0.012, 0.015) # mic 6
38 ])
39
40 # Realizacao do DOA
41 painel = semblance.z_painel(delta,micPos, speed_of_sound, Fs, x)
42 predict_doa = semblance.doa_prediction(painel, delta)
43
44 # Composicao da resposta de saida
45 tempo = str(time.time()-inicio)+';' # Tempo de execucao
46 canais = str(len(x[1]))+';' # Numero de canais
47 amostragem = str(Fs)+';' # Taxa de amostragem
48 azimuth = str(predict_doa[0])+';' # Azimute predito
49 elevacao = str(predict_doa[1])+';' # Elevacao predita
50 delta2 = str(delta)+';' # Passo utilizado
51
52 # Saida via print
53 out = amostragem+canais+delta2+azimuth+elevacao+tempo # Saida
54 print(out)

```

```

1 # H. Ferreira, 2019
2
3 # -----
4 # -----Controle dos Servos-----
5 # -----
6
7 # Bibliotecas
8 from nanpy import (ArduinoApi, SerialManager, Servo)
9 from time import sleep
10 import numpy as np
11 import sys
12
13 # Raspberry tenta se conectar ao Arduino
14 try:
15     connection = SerialManager()
16     ard = ArduinoApi(connection = connection)
17 except:
18     print("Falha ao se conectar ao Arduino")
19
20 # Definicao dos pinos
21 pinLaser = 7
22 pinServo_elevation = 8
23 pinServo_azimuth = 9
24
25 # Variaveis (Entrada via argumento)
26 azimuth = sys.argv[1]
27 elevation = sys.argv[2]
28 delay = 5 # Tempo que o laser fica aceso
29
30 # Setup dos Servos
31 servo_elevation = Servo(pinServo_elevation)
32 servo_azimuth = Servo(pinServo_azimuth )
33
34 # Setup do Laser
35 ard.pinMode(pinLaser, ard.OUTPUT)
36
37 # Atuacao Servo
38 servo_azimuth.write(azimuth)
39 print("Azimute", azimuth)
40 servo_elevation.write(elevation)
41 print("Elevacao", elevation)
42 sleep(1)
43
44 # Liga o Laser e depois de delay segundos, desliga
45 ard.digitalWrite(pinLaser, ard.HIGH)
46 sleep(delay)
47 ard.digitalWrite(pinLaser, ard.LOW)

```

```
1  #!/bin/bash
2  # H. Ferreira, 2019
3  # Script complementar para realizacao das medidas de desempenho angular
4
5  read angulo
6  delta=10
7
8  # Coleta dos arquivos
9  for i in {1..30};
10 do
11 arecord -D ac108 -f S32_LE -r 48000 -c 6 -d 1 $i.wav
12 done
13
14 # Processamento
15 for i in {1..30};
16 do
17
18 python global.py $delta $i.wav >> resultados_${angulo}
19 rm $i.wav
20
21 done
22
```

```
1  #!/bin/bash
2  # H. Ferreira, 2019
3  # Script complementar para realizacao do loop do pipeline
4  # DOA + apontamento do Laser
5
6  # Loop de repeticao
7  while true
8  do
9  # Captura o audio
10 arecord -D ac108 -f S32_LE -r 48000 -c 6 -d 1 temp.wav
11 # Encontra a direcao e aponta o laser
12 python global_servo.py temp.wav
13
14 # Exclui o audio
15 rm temp.wav
16 done
17
```

```

1 # H. Ferreira, 2019
2
3 # -----
4 # ----- DOA e Apontamento do Laser -----
5 # -----
6
7 # Bibliotecas
8 import semblance
9 import wavio
10 import numpy as np
11 import sys
12 import time
13 from nanpy import (ArduinoApi, SerialManager, Servo)
14 from time import sleep
15 import sys
16
17 def apontar_laser(azimuth,elevation,delay):
18
19     # Raspberry tenta se conectar ao Arduino
20     try:
21         connection = SerialManager()
22         ard = ArduinoApi(connection = connection)
23     except:
24         print("Falha ao se conectar ao Arduino")
25
26     # Definicao dos pinos
27     pinLaser = 7
28     pinServo_elevation = 8
29     pinServo_azimuth = 9
30
31     servo_elevation = Servo(pinServo_elevation) # Setup Servos
32     servo_azimuth = Servo(pinServo_azimuth )    # Setup Servos
33     ard.pinMode(pinLaser, ard.OUTPUT)          # Setup do Laser
34
35     # Atuacao Servo
36     servo_azimuth.write(azimuth)
37     print("Azimute",azimuth)
38     servo_elevation.write(elevation)
39     print("Elevacao", elevation)
40     sleep(1)
41
42     # Liga o Laser e depois de delay segundos, desliga
43     ard.digitalWrite(pinLaser, ard.HIGH)
44     sleep(delay)
45     ard.digitalWrite(pinLaser, ard.LOW)
46
47     pass
48
49     # Arquivo de audio
50     file = sys.argv[1]          # Entrada do arquivo de audio
51     wav = wavio.read(file)     # Leitura do arquivo de audio
52     Fs = wav.rate              # Taxa de amostragem
53     x = semblance.normalize(wav.data[:,1:6])    # Normaliza os canais
54
55     # Parametros do DOA - Semblance
56     delta = 10.0               # Passo do gride de angulos [graus]
57     speed_of_sound = 342      # Velocidade do som [m/s]
58
59     # Posicao dos microfones
60     micPos = np.array([
61         #(-0.175, 0.035, 0.015), # mic 1
62         (-0.175, 0.085, 0.015), # mic 2
63         (-0.135, 0.112, 0.015), # mic 3
64         (-0.095, 0.085, 0.015), # mic 4
65         (-0.095, 0.035, 0.015), # mic 5
66         (-0.135, 0.012, 0.015)  # mic 6
67     ])
68
69     painel = semblance.z_painel(delta,micPos, speed_of_sound, Fs, x)
70     predict_doa = semblance.doa_prediction(painel, delta)
71     apontar_laser(predict_doa[0],predict_doa[1],5)

```