

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
PROJETO DE GRADUAÇÃO**



**LEONI MOTA LORIS**

**UM ESTUDO DE TÉCNICAS DE DETERMINAÇÃO DE  
DIREÇÃO DE CHEGADA EM ARRANJO DE SENSORES.**

VITÓRIA – ES  
Julho/2017

LEONI MOTA LORIS

**UM ESTUDO DE TÉCNICAS DE DETERMINAÇÃO DE  
DIREÇÃO DE CHEGADA EM ARRANJO DE SENSORES.**

Parte manuscrita do Projeto de Graduação do aluno Leoni Mota Loris, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

---

Dr. Evandro Ottoni Teatini Salles  
Orientador

---

Leoni Mota Loris  
Aluno

LEONI MOTA LORIS

## **UM ESTUDO DE TÉCNICAS DE DETERMINAÇÃO DE DIREÇÃO DE CHEGADA EM ARRANJO DE SENSORES.**

Parte manuscrita do Projeto de Graduação do aluno Leoni Mota Loris, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovada em \_\_\_\_, de \_\_\_\_ de \_\_\_\_.

### **COMISSÃO EXAMINADORA:**

---

**Dr. Evandro Ottoni Teatini Salles**  
**UFES**  
**Orientador**

---

**Dr. Klaus Fabian Côco**  
**UFES**  
**Examinador**

---

**Dr. Marcelo Eduardo Vieira Segatto**  
**UFES**  
**Examinador**

## RESUMO

Neste trabalho, arranjos de sensores são abordados assim como as principais implementações de algoritmos para o processamento de sinais em alta resolução, em especial para sensores acústicos. Algoritmos de detecção da direção de chegada foram implementados e avaliados em um ambiente simulado (Algoritmo MUSIC – *Multiple Signal Classification*, correlação cruzada e por detecção de fase) e um algoritmo para a reconstrução do sinal final na formação do feixe (algoritmo GSC – *Generalized Sidelobe Canceller*). Foi em seguida avaliada sua implementação em um sistema microcontrolado (microcontrolador TM4C123GH6PM), onde restrições de processamento e memória direcionaram o modo de implementação. Para embarcar um algoritmo no microcontrolador, foi necessário avaliar o custo computacional e os requisitos de memória das opções testadas nesse trabalho, optando-se pela abordagem LMS - *Least Mean Square*. De modo a garantir resposta em tempo real avaliou-se duas arquiteturas de *software* embarcado: Sistema operacional de tempo real (*FreeRTOS – Free Real Time Operating System*) e *Protothreads*. Por questões de implementação, optou-se pela *Protothread*. O sistema de *hardware* desenvolvido consistiu, além do TM4C123GH6PM, de dois microfones e um *display* gráfico. Foi desenvolvida uma bancada de testes e os resultados alcançados foram satisfatórios, mesmo para um algoritmo simples como LMS.

Palavras chave: Arranjo de sensores, direção de chegada, processamento de sinais, *Beamforming*, Sistemas embarcados.

## LISTA DE FIGURAS

Figura 1: Separação de fontes em uma imagem com reflectância.....	13
Figura 2: Sistema de reconstrução vocal LPC .....	16
Figura 3: Estrutura para filtragem espacial em um arranjo de antenas.....	17
Figura 4: Exemplo de um trecho estacionário em um sinal do fonema /a/ gravado.....	18
Figura 5: Arranjo linear de sensores recebendo uma frente de ondas em um ângulo $\theta$ . .....	19
Figura 6: Magnitude da saída normalizada em função do ângulo de chegada onde a linha preta representa uma atenuação máxima em um lóbulo secundário de 14dB ( $d = \lambda/2$ m, $\lambda =$ (340 m/s/ 100 Hz), $N = 5$ ).....	21
Figura 7: Módulo da saída no arranjo de sensores com o <i>Delay-sum beamformer</i> ao variar o número de microfones. ....	23
Figura 8: Potência normalizada no <i>Delay-sum beamformer</i> ao variar a distância entre sensores.....	24
Figura 9: Espectro em frequência de uma senoide amostrada a uma taxa de 1000 amostras/segundo.....	25
Figura 10: Arranjo de sensores recebendo um sinal em um cenário em que $\lambda/2 < l < \lambda$ .....	26
Figura 11: Deslocamentos discrepantes $\tau$ (real) e $\tau'$ (detectado). ....	26
Figura 12: Resposta em frequência dos filtros passa faixa.....	28
Figura 13: Potência da saída do Delay-Sum Beamformer.....	32
Figura 14: Potência da saída do Beamformer MVDR.....	33
Figura 15: Potência da saída do Salay-Sum Beamformer .....	33
Figura 16: Potência da saída do MCDR Beamformer .....	34
Figura 17: Correlação cruzada de um quadro de amostras (ângulo de chegada=50°).....	37
Figura 18: Correlação cruzada de um quadro de amostras (ângulo de chegada =90°).....	37
Figura 19: Correlação cruzada de um quadro de amostras (ângulo de chegada =30°).....	38
Figura 20: Direções de chegada encontradas com voz masculina.....	39
Figura 21: Direções de chegada encontradas com voz feminina.....	39
Figura 22: Histograma dos erros na identificação do ângulo de chegada. a) voz masculina. b) voz feminina. ....	40
Figura 23: Sinais teste para o filtro LMS e o filtro obtido.....	43
Figura 24: Filtro adaptativo obtido de um quadro de amostras (ângulo de chegada=90°, voz masculina).....	44

Figura 25: Filtro adaptativo obtido de um quadro de amostras (ângulo de chegada=90°, voz feminina).....	44
Figura 26: Filtro adaptativo obtido de um quadro de amostras (ângulo de chegada=30°, voz masculina).....	45
Figura 27: Filtro adaptativo obtido de um quadro de amostras (ângulo de chegada=30°, voz feminina).....	45
Figura 28: Direções de chegada encontradas (voz masculina).....	46
Figura 29: Direções de chegada encontradas (voz feminina).....	47
Figura 30: Estimção da direção de chegada com o MUSIC.....	50
Figura 31: Algoritmo MUSIC com ângulos de chegada = 0° (voz masculina) e -60°(voz feminina).....	51
Figura 32: Algoritmo MUSIC com ângulos de chegada = 0° (voz masculina) e -32° (voz feminina).....	51
Figura 33: Algoritmo MUSIC com ângulos de chegada = 0° (voz masculina) e -10° (voz feminina).....	52
Figura 34: Estrutura do GSC .....	54
Figura 35: Módulo da transformada de Fourier da saída.....	56
Figura 36: Módulo da transformada de Fourier da saída.....	57
Figura 37: Potência normalizada na saída do GSC mantendo uma das fontes em 90° (voz masculina) e a segunda fonte variando de 0° até 180° (voz feminina). .....	58
Figura 38: Microfone eletreto pré-amplificador .....	60
Figura 39: Erro na compilação do software com o compilador da Texas Instruments .....	62
Figura 40: Aplicativo PitchLab Pro para estimção da frequência fundamental <b>Erro! Indicador não defini</b> .....	
Figura 41: Arranjo de microfones com dois microfones e LCD .....	66
Figura 42: Fluxograma do Software embarcado .....	67
Figura 43: Locutor masculino falando mais perto do microfone da direita (a metade da direita do sinal corresponde ao microfone da direita e respectivamente, a metade da esquerda, ao microfone da esquerda).....	67
Figura 44: Locutor masculino em 60° (identificou uma fonte em 77°). .....	68
Figura 45: Locutor masculino em 30° (identificou uma fonte em 31°). .....	69
Figura 46: Locutor feminino em 60° (identificou uma fonte em 64°). .....	69
Figura 47: Locutor feminino em 30° (identificou uma fonte em 31°). .....	70
Figura 48: Ângulo de chegada encontrado em função do ângulo de chegada real.....	71

Figura 49: Histograma do erro para os ângulos encontrados. a) voz masculina. b) voz feminina .....	71
Figura 50: Placa com o microcontrolador STM32L476.....	73

## **LISTA DE TABELAS**

Tabela 1: Sinais chegando ao arranjo .....	55
Tabela 2: Complexidade dos algoritmos para encontrar a direção de chegada.....	60
Tabela 3: Especificações do microcontrolador.....	61

## LISTA DE ABREVIATURAS E SIGLAS

DTI	Diferença de Tempo Interaural
DII	Diferença de Intensidade Interaural
LSA	<i>Linear Sensor Array</i> (Arranjo linear de sensores)
LPC	<i>Linear Predictive Coding</i> (Codificação preditiva linear)
LMS	<i>Least Mean Squares</i> (Mínimo Erro Médio Quadrático)
PCM	<i>Pulse Code Modulation</i> (Modulação por Largura de Pulso)
C-LMS	<i>Constrained LMS</i> (LMS Restringido)
RLS	<i>Recursive least squares</i> (Mínimos Quadrados Recursivos)
RTOS	<i>Real Time Operating System</i> (Sistema Operacional de Tempo Real)
BSS	<i>Blind Source Separation</i> (Separação cega de Fontes)
SPI	<i>Serial Peripheral Interface</i> (Interface Periférica Serial)
$\mu$ DMA	<i>Micro Direct Memory Access</i> (Micro Acesso Direto à Memória)

## LISTA DE SÍMBOLOS

$x_m[t]$	Sinal amostrado no tempo $t$ do sensor $m$
$y_m[t]$	Saída filtrada do sensor $m$
$y[t]$	Soma de todas as saídas no arranjo de sensores
$M$	Número de sensores
$N$	Número de amostras em cada quadro
$d$	Número de fontes
$\mathbf{x}[t]$	Vetor contendo sinais de todos os $M$ sensores no tempo $t$
$\lambda$	Comprimento de onda do sinal $x_m[t]$
$\theta$	Ângulo de chegada da frente de ondas
$l$	Distância entre os sensores
$f_o$	Frequência fundamental do sinal $x_m[t]$
$O(\cdot)$	Ordem da complexidade do algoritmo
$\mathbf{x}^T$	Transposição do vetor $\mathbf{x}$
$\mathbf{w}_m$	Vetor de coeficientes do filtro no sensor $m$
$F(f(t))$	Transformada de Fourier da função $f(t)$
$\tau(\theta)$	Tempo que a frente de ondas leva para chegar de um sensor ao próximo sensor

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>12</b>
1.1	Contextualização .....	12
1.2	Aplicações .....	12
1.3	Objetivos.....	14
1.4	Distribuição do trabalho .....	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>16</b>
2.1	Modelo do sinal de voz.....	16
2.2	Processamento no Arranjo Linear de Sensores .....	19
2.2.1	Modelo do Arranjo .....	19
2.2.2	Delay-Sum Beamformer.....	20
2.3	Amostragem .....	24
2.4	Considerações sobre as simulações .....	27
<b>3</b>	<b>MVDR – (MINIMUM VARIANCE DISTORTIONLESS RESPONSE).....</b>	<b>29</b>
<b>4</b>	<b>DIREÇÃO DE CHEGADA .....</b>	<b>34</b>
4.1	Correlação cruzada .....	35
4.1.1	Fundamentos .....	35
4.1.2	Resultados .....	36
4.2	Algoritmo adaptativo.....	40
4.2.1	Fundamentos .....	40
4.2.2	Resultados .....	43
4.3	MUSIC (Multiple Signal Classification).....	47
4.3.1	Fundamentos .....	47
4.3.2	Resultados .....	50
<b>5</b>	<b>BEAMFORMING .....</b>	<b>53</b>
5.1	Generalized Sidelobe Canceller .....	53
5.2	Resultados.....	55
<b>6</b>	<b>SISTEMA EMBARCADO .....</b>	<b>58</b>
6.1	Software embarcado .....	60
6.2	Protothreads .....	62
6.3	Implementação .....	65
6.4	Resultados.....	67

<b>7 CONCLUSÕES .....</b>	<b>72</b>
<b>8 REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>75</b>

# 1 INTRODUÇÃO

## 1.1 Contextualização

Desde cedo no ciclo de vida humano é observado um grande desenvolvimento, principalmente a partir dos seis meses, de sua capacidade de localizar sons em sua volta através de um processo de localização acústica. A localização acústica é baseada em diferenças sutis na intensidade sonora e na diferença de tempo da chegada do som nos dutos auditivos. Este é um processo fundamental que está relacionado à sobrevivência de diversos seres vivos que empregam este sistema quando precisam inferir sobre a localização de presas ou predadores. (FURST, *et al.*, 2004).

Em um contexto diferente, a localização acústica motiva o estudo de métodos de Separação Cega de Fontes (*Blind Source Separation* ou *BSS* do inglês), cujas aplicações alcançam as áreas de processamento de sinais, aprendizado de máquinas e visão computacional. A Separação é dita ‘cega’ pois não há informação alguma (ou pouca informação) sobre as fontes de sinal.

O problema da separação cega de fontes consiste em recuperar o conjunto de sinais da Equação (1) oriundos de  $d$  fontes diferentes:

$$\mathbf{s}(t) = [s_1(t), s_2(t) \dots s_d(t)]^T. \quad (1)$$

Os sinais da Equação (1) são misturados pela matriz  $A$  e medidos por  $M$  sensores:

$$\mathbf{x}(t) = [x_1(t), x_2(t) \dots x_M(t)]^T.$$

$\mathbf{x}(t)$  e  $\mathbf{s}(t)$  se relacionam com  $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ , sendo  $A$  a matriz de mistura com  $M$  linhas e  $d$  colunas. Quando  $M > d$  o sistema é chamado de sobredeterminado, com solução em métodos lineares simples, ao passo que se  $M < d$ , o sistema é dito subdeterminado, necessitando de métodos mais sofisticados para se recuperar o sinal. (CARDOSO, 1998).

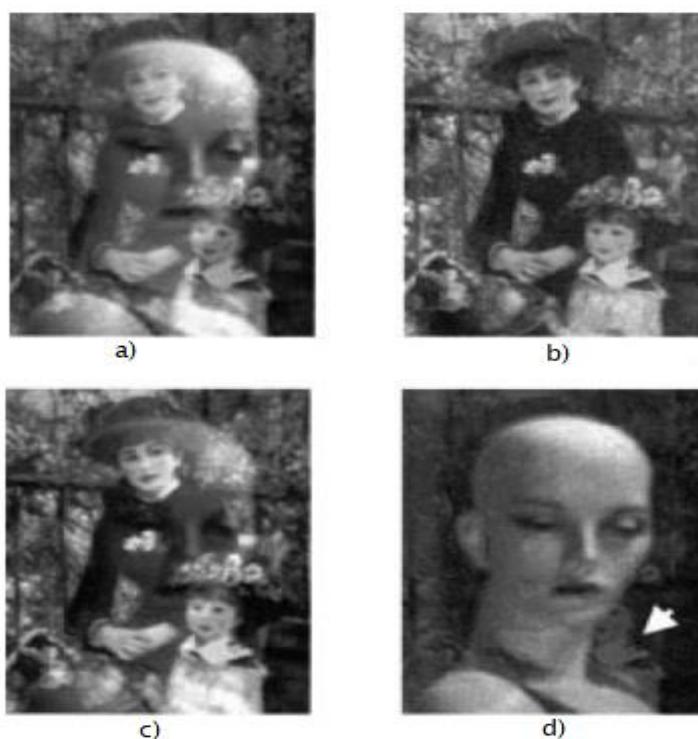
## 1.2 Aplicações

Como mencionado, as aplicações dos métodos de BSS alcançam não só processamento de sinais em alta resolução como também visão computacional e aprendizado de máquinas. É entendido como processamento de sinais em alta resolução o tratamento dos sinais recebidos quando são dispostos de arranjos de sensores (acústicos ou eletromagnéticos) para separar,

amplificar ou atenuar fontes de sinais (por ex.: Locutores, para o caso de sensores acústicos ou usuários de uma rede de telecomunicações).

Para aplicações em visão computacional e aprendizado de máquinas, é possível citar misturas instantâneas de fotos onde há refletância através de uma superfície transparente como na Figura 1. Nas imagens a) e c), as fontes estão misturadas e na segunda coluna observamos as duas fontes separadas utilizando um dos algoritmos de separação de fontes (algoritmo ICA, do inglês *Independent Component Analysis*).

Figura 1: Separação de fontes em uma imagem com reflectância



Fonte: BRONSTEIN, 2005.

Sendo assim, serão investigados métodos de separação de fontes para sinais sonoros (sendo estas fontes, o preletor, o professor ou o palestrante, por exemplo).

As principais dificuldades encontradas ao trabalhar com sinais de voz são referentes à sua largura de banda. Algoritmos de separação de fontes em arranjos de sensores foram desenvolvidos voltados para sistemas de comunicação, no qual o sinal recebido no arranjo de sensores é modelado como uma exponencial complexa de frequência constante, (portanto,

banda estreita), obrigando que os algoritmos sejam adaptados para tratar sinais de som (agora, banda larga). É considerada também a não estacionariedade do sinal de voz (discutido na seção 2.1) (DE OLIVEIRA, A. *et al.* 2008).

Desta maneira, é disposto um arranjo de sensores para estimar as informações sobre a localização de uma fonte sonora (e conseqüentemente separar as fontes). O arranjo utilizado é chamado de Arranjo Linear de Sensores ou LSA (*Linear Sensor Array*), pois seus sensores estão dispostos linearmente com distância fixa entre eles (retratado na Figura 6).

### **1.3 Objetivos**

Neste trabalho, é proposta a investigação e implementação de alguns principais algoritmos em arranjos de sensores assim como o estudo de uma técnica de *Beamforming*. Em seguida, será utilizada em um sistema embarcado uma dessas técnicas, avaliando-se os resultados de se projetar um arranjo de sensores em um sistema com limitação de capacidade de processamento e memória.

*Beamforming* ou *Beamformer* (traduzido em algumas referências de língua portuguesa como ‘Formação De Feixe’) tem como objetivo ‘direcionar’ o lóbulo principal do arranjo de sensores para a fonte de interesse, assim como, atenuar seus lóbulos secundários, impedindo que sinais interferentes apareçam no sinal de saída do arranjo. O processo de *Beamforming* depende por vezes do conhecimento prévio da direção de chegada, seja para alinhar os sinais de chegada ou para encontrar os filtros do *beamformer*. Foi trabalhado desta forma, com métodos que encontram a direção de chegada de um sinal.

Neste trabalho foi simulado um arranjo de sensores no qual três métodos para a descoberta do ângulo de chegada de um sinal foram implementados. Estes métodos são: MUSIC (*Multiple signal Classification*), a correlação cruzada entre os sensores e uma técnica baseada em um algoritmo adaptativo. Em seguida, é implementado dois *beamformers*, o *Delay-Sum Beamformer* e o GSC (*Generalized Sidelobe Canceller*). Sinais de teste para as simulações foram obtidos com gravações de vozes masculinas, vozes femininas e uma música, através do microfone interno do computador. Ao avaliar os resultados das simulações, foi escolhida a

melhor abordagem para a implementação do sistema em um microcontrolador TM4C123GH6PM, da *Texas Instruments*.

#### ***1.4 Distribuição do trabalho***

O trabalho é dividido em duas partes, a primeira refere-se às simulações computacionais e a segunda compreende na implementação com um sistema embarcado. Por isto, este texto segue a seguinte divisão:

Foi abordado no capítulo 2 os fundamentos teóricos básicos do sinal com o qual será trabalhado, assim como as técnicas utilizadas para seu processamento.

A primeira parte do trabalho, representado pelas simulações, é abordada nos capítulos 3 até o capítulo 5. No capítulo 3 é mostrada uma técnica geral de *Beamforming* que serve também para encontrar direções dos ângulos de chegada. No desenvolver do capítulo são percebidas as necessidades de técnicas separadas para encontrar o ângulo de chegada e técnicas de *Beamforming* que são abordadas nos capítulos que o seguem.

No capítulo 4 é discorrido sobre as três técnicas para a descoberta dos ângulos de chegada e resultados de simulações em vários cenários são discutidos para cada uma das técnicas.

Em seguida, no capítulo 5 é discutido sobre a principal técnica de *Beamforming* assim como suas vantagens sobre as precedentes. Simulações em cenários diferentes foram feitas e seus resultados foram expostos e discutidos.

No capítulo 6 é discutida a implementação do algoritmo escolhido em um sistema embarcado assim como seus critérios de escolha. É abordado neste mesmo capítulo as *Protothreads*, que foram utilizadas neste sistema embarcado.

A conclusão deste texto é feita no capítulo 7, sumarizando os resultados obtidos e definindo passos futuros para uma continuação do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

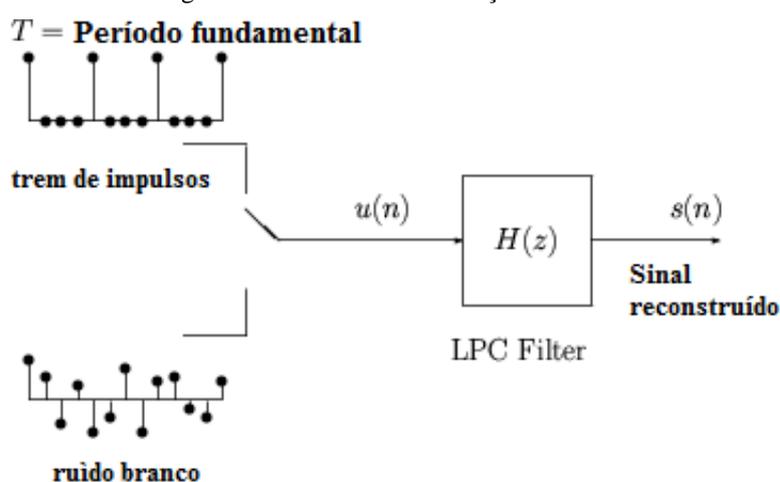
### 2.1 Modelo do sinal de voz

O sinal envolvido neste trabalho se trata de um sinal de voz, e é de sua natureza física variar temporalmente e espacialmente.

A variação temporal é mais bem observada quando é analisado um fonema emitido pelo locutor. Este fonema pode ser aproximado por um modelo Fonte-Filtro, no qual a fonte é vinculada à excitação variante no tempo fornecida às cordas vocais na forma de variação da pressão do ar. Já o filtro faz referência à filtragem sonora variante no tempo realizada pela dinâmica do canal vocal (laringe, faringe, cavidade oral e a cavidade nasal). Neste Modelo, sons vocálicos (p. ex.: /a/, /o/) compreenderão sinais cuja fonte é devida a uma excitação periódica enquanto que fonemas fricativos (p. ex.: /z/, /v/) têm em sua fonte, um ruído turbulento vinculado a uma constrição no canal vocal (FANT G. 1971).

Existe uma codificação do sinal de voz que se baseia no modelo Fonte-Filtro do canal vocal chamada Codificação Linear Preditiva (LPC – *Linear Predictive Coding*) que consiste em utilizar um trem de impulsos ou um ruído branco (ruído contendo todas as componentes de frequências) como sinal fonte (para fonemas vocálicos e para fonemas fricativos, respectivamente). Esta codificação é retratada na Figura 2.

Figura 3: Sistema de reconstrução vocal LPC

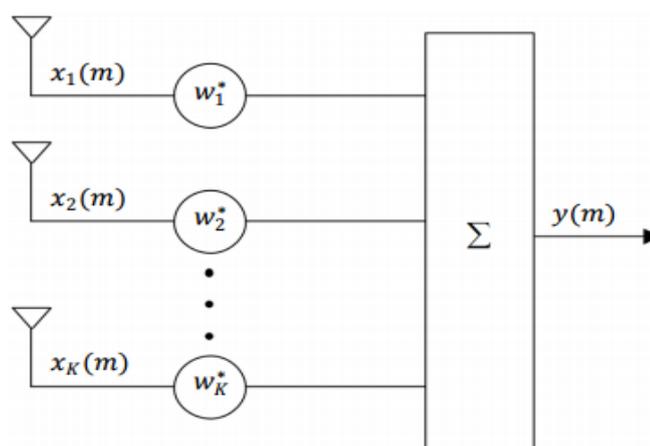


Fonte: ELECTRONIC FRONTIER FOUNDATION, 2000.

O sinal da fonte  $u(n)$  é selecionado e em seguida filtrado usando a função de transferência  $H(z)$ . Os parâmetros do filtro LPC podem ser estimados usando o método descrito por Aarts M. *et al.*, (2012), que exige uma inversão da matriz de autocorrelação estimada do sinal original.

A codificação e filtragem LPC é um exemplo de tratamento temporal do sinal enquanto que a filtragem espacial pode ser compreendida como o tratamento feito em um sinal quando suas amostras estão dispostas espacialmente, como em uma imagem ou em um instante de tempo no arranjo de antenas. A estrutura convencional da filtragem espacial em arranjos de antenas é indicada na Figura 4.

Figura 4: Estrutura para filtragem espacial em um arranjo de antenas



Fonte: CHEUNG, B. 2002.

De um ponto de vista estatístico, quando diz-se que um sinal é um processo estocástico estacionário (sentido amplo) é dito que seu valor esperado (operador esperança) é constante e independente do tempo e sua correlação depende somente da diferença de instantes em que ela é avaliada. Sendo assim, usufruindo da estacionariedade de um sinal é possível utilizar abordagens tal como a modelagem A.R.M.A. (*Autoregressive–Moving–Average*) para o tratamento do sinal.

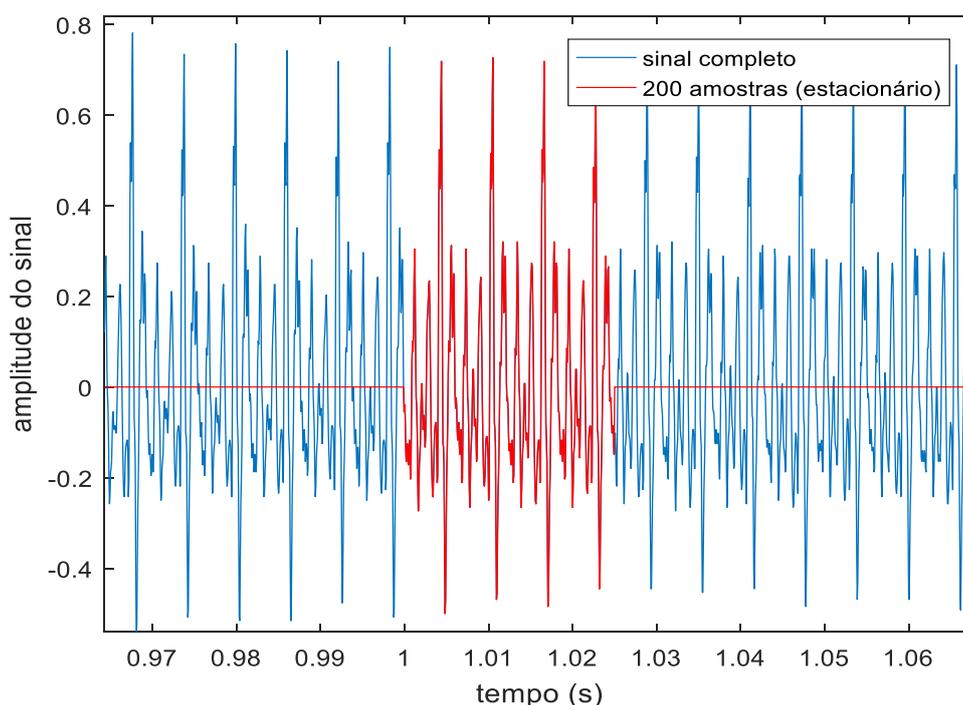
Como já mencionado, o sinal de voz apresenta a característica de ser não estacionário. Entretanto avaliando-o em uma pequena vizinhança de um instante de tempo em um fonema vocálico, este trecho pode ser modelado como um sinal estacionário contendo componentes de frequência que são chamados de formantes. Estas dão características ao timbre, frequência

fundamental e a dinâmica do sinal (primeiro ao terceiro formante caracterizam a vogal, os próximos, o locutor).

Para a captura de um trecho estacionário do sinal de voz, um intervalo de tempo de 25ms (tempo médio de um sinal de voz quase-estacionário, que pode variar de 15-25ms de acordo com Eringis D. e Tamulevičius G. 2015) será amostrado a uma taxa de 8000 amostras por segundo, contendo 200 amostras (podendo ser estendido para 256 amostras, pela facilidade com o processamento de vetores cujos tamanhos são potências de 2) para formar um quadro (do inglês *frame*) de amostras.

A Figura 5 ilustra uma seleção de 200 amostras (em vermelho) retiradas do sinal do fonema /a/ gravado por um locutor masculino.

Figura 5: Exemplo de um trecho estacionário em um sinal do fonema /a/ gravado.



Fonte: Script MATLAB do autor.

Nota-se que este, sendo um som vocálico, apresenta uma periodicidade bem definida, mesmo observando um período maior do que 25ms podemos notar esta periodicidade referente à frequência fundamental.

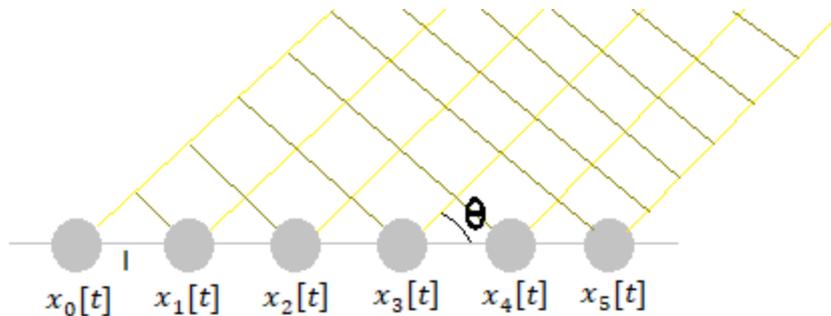
## 2.2 Processamento no Arranjo Linear de Sensores

### 2.2.1 Modelo do Arranjo

A estrutura do LSA é composta de  $M$  sensores igualmente distanciados. A distância entre as fontes de sinal e os sensores deve ser maior do que o tamanho da LSA e igualmente, a distância entre os sensores deve ser menor do que o comprimento de onda inerente à frequência fundamental do som em avaliação (*aliasing* espacial, discutido na seção 2.3), mas não tão menor, evitando que torne os sinais nos diversos sensores indistinguíveis (AARTS M. *et al.* 2012).

No arranjo de sensores exemplificado na Figura 6, está os sinais amostrados são representados no 1°, 2° ... 6° sensor como  $x_0[t], x_1[t] \dots x_5[t]$ , onde  $x_m[t]$  representa o sinal amostrado no tempo  $t$  do sensor  $m$ . A distância entre os sensores é  $l$  e o ângulo de chegada da frente de ondas do sinal desejado é  $\theta$ . Cada sensor  $m$  irá filtrar temporalmente o seu sinal usando um filtro cujos coeficientes são dados pelo vetor  $\mathbf{w}_m[t]$ .

Figura 6: Arranjo linear de sensores recebendo uma frente de ondas em um ângulo  $\theta$ .



Fonte: KRIM H. VIBERG M. (1996).

Na Equação (2) é dado o sinal na saída do sensor  $m$ .

$$y_m[t] = \sum_{k=0}^{N-1} \mathbf{w}_m[k] x_m[t - k] \quad (2)$$

Na Equação (2),  $N$  é a quantidade de amostras em um pequeno trecho estacionário (*frame* do sinal, contendo 256 amostras) e  $t$ , a variável temporal que será avaliada nos tempos de amostragem  $t \in \{0, T, 2T, 3T, \dots\}$  com  $T$  igual ao período de amostragem.

É desenvolvida a Equação (3) da saída em um sensor. Em que  $\delta[t]$  é um impulso unitário discreto centrado na origem e  $\Delta t_m$  é o tempo que deslocará o impulso para o sensor  $m$  e  $w_m[t]$  é o filtro do sensor  $m$ .

$$\begin{aligned} w_m[t] &= \delta[t + \Delta t_m]. \\ y_m[t] &= x[t + \Delta t_m]. \end{aligned} \quad (3)$$

### 2.2.2 Delay-Sum Beamformer

Considerando agora que a mesma frente de ondas chegará a todos os sensores em instantes de tempos diferentes, a variável temporal  $t$  na equação (3) deverá ser deslocada em função da posição do sensor e do ângulo de chegada da frente de ondas, uma vez que uma amostra do sinal chegará mais atrasada quanto maior for a distância do sensor à fonte e quanto menor for o ângulo de chegada  $\theta$ . Pela Figura 6, infere-se usando relações trigonométricas que o tempo em que a frente de ondas demorará a chegar ao sensor  $m$  tendo alcançado o sensor zero é dado pela Equação (4):

$$tempo = \frac{m l \cos(\theta)}{v} = \frac{m l \cos(\theta)}{\lambda f}, \quad (4)$$

onde  $v$  é a velocidade do som,  $\lambda$  é o comprimento de onda do sinal  $x$  e  $f$  é a frequência deste sinal.

Assim, a Equação (5) da saída será obtida ao somar o sinal advindo de todos os  $M$  sensores no qual a variável  $t$  estará devidamente deslocada pela Equação (4).

$$y[t] = \sum_{m=0}^{M-1} x_m \left[ t - \frac{m l \cos(\theta)}{\lambda f} + \Delta t_m \right]. \quad (5)$$

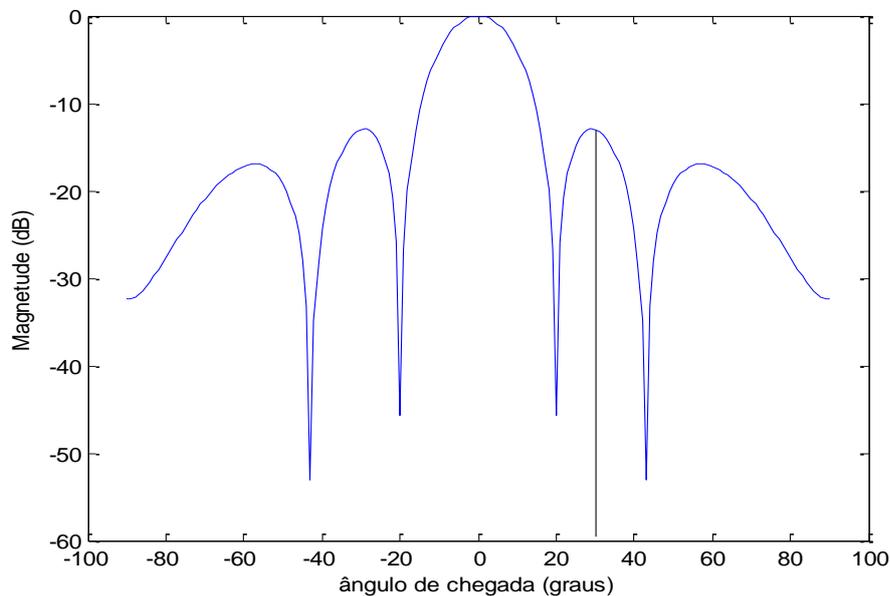
É possível notar a partir da Equação (5) que quando o atraso  $\Delta t_m$  gerado pelo filtro do sensor for igual ao tempo definido na Equação (4), o sinal total será simplesmente a soma do mesmo sinal  $x_m[t]$ . Este *Beamformer* onde os filtros dos sensores são deslocadores do sinal é chamado de *Delay-and-Sum Beamformer*. Para obter a resposta do *Delay-and-Sum Beamformer*, é escolhido um sinal fonte complexo  $s[t] = e^{2\pi i f_o t}$  com uma única componente

de frequência, representada por  $f_0$ , comprimento de onda  $\lambda$  (sendo  $i$  a unidade complexa). É obtido a partir da Equação (5):

$$\begin{aligned}
 y[t] &= \sum_{m=0}^{M-1} e^{2\pi i f_0 (t - \frac{m l \cos(\theta)}{\lambda f_0} + \Delta t)}, \\
 y[t] &= \sum_{m=0}^{M-1} e^{i 2\pi f_0 t} \cdot e^{-\frac{i m l \cos(\theta) 2\pi f_0}{\lambda f_0}} \cdot e^{i 2\pi f_0 \Delta t}, \\
 y[t] &= e^{i 2\pi f_0 (t + \Delta t)} \sum_{m=0}^{M-1} e^{-\frac{2\pi i m l \cos(\theta)}{\lambda}}. \tag{6}
 \end{aligned}$$

Como experiência,  $\Delta t = \frac{l}{v}$  é tomado, indicando que os filtros nos sensores privilegiarão o ângulo  $0^\circ$  (derivado da Equação (4)), uma vez que o sistema somará as saídas considerando a frente de onda ortogonal ao arranjo de sensores. Um *script* em MATLAB foi executado variando o ângulo  $\theta$  para traçar a curva do módulo de  $y[t]$ , e esta, é mostrada na Figura 7.

Figura 8: Magnitude da saída normalizada em função do ângulo de chegada onde a linha preta representa uma atenuação máxima em um lóbulo secundário de 14dB ( $l = \lambda/2$  m,  $\lambda = (340 \text{ m/s} / 100 \text{ Hz})$ ,  $M = 5$ ).



Fonte: script MATLAB do autor.

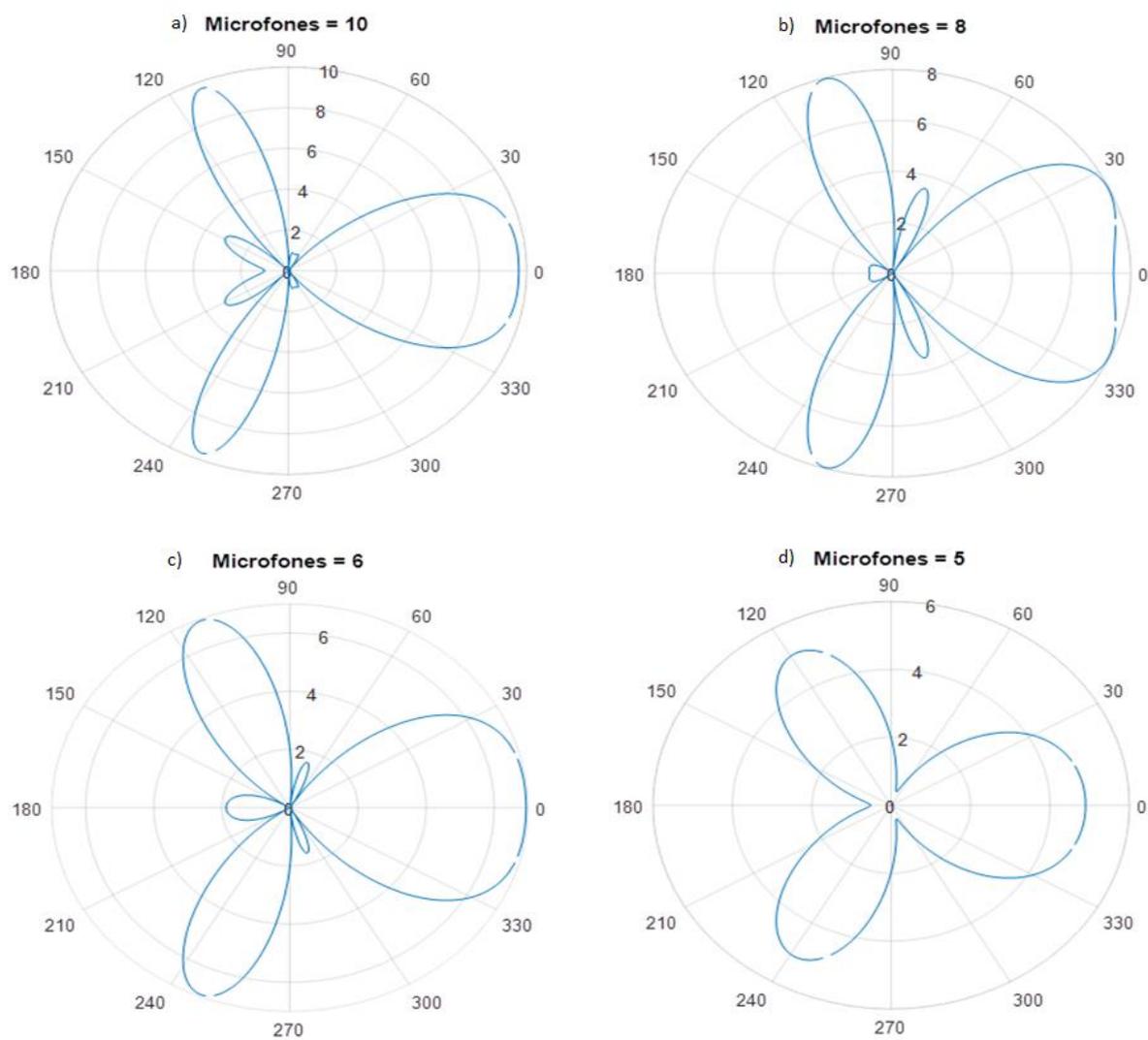
Nota-se que o arranjo terá ganho unitário quando ângulo de chegada iguala a  $0^\circ$  (e múltiplos de  $180^\circ$ , devido a periodicidade de  $y[t]$  em relação à  $\theta$ ) pois esta foi normalizada. Uma atenuação em outros ângulos, como em  $20^\circ$  é também observado. Observa-se ainda a

existência de um lóbulo secundário em  $30^\circ$  (indicado pela linha preta na Figura 8) fornecendo uma atenuação de aproximadamente 14dB. Considerando que este não deveria ser um ângulo de preferência, qualquer fonte interferente que está na direção de um lóbulo secundário aparecerá no sinal final, fazendo necessário o uso de um segundo tratamento.

É possível observar a partir da Figura 9 que quantidade de lóbulos secundários aumenta com quantidade de sensores. Isto é um fenômeno típico em arranjos de sensores. O número dos lóbulos secundários aumentará com o número de elementos do arranjo e com o espaçamento entre os sensores (Figura 10). E é importante notar também que devido à natureza periódica da Equação (6), para cada fonte detectada, existirão dois lóbulos principais, salvo quando a distância entre os sensores é maior do que  $\frac{\lambda}{2}$ , gerando uma maior quantidade de lóbulos secundários. Este caso é representado nas Figuras 10.c e 10.d (*espacial aliasing*, tema que será discutido na próxima seção).

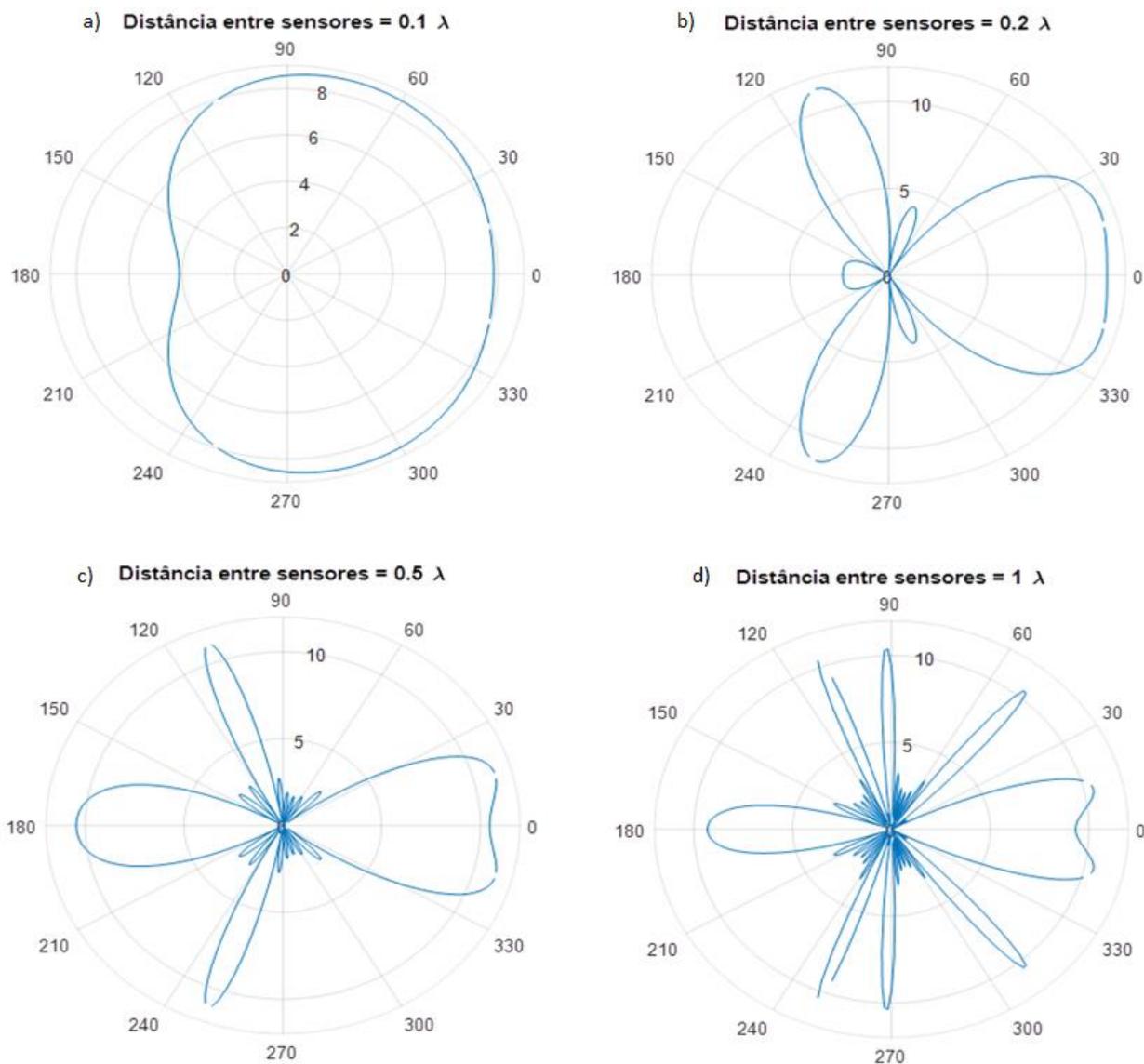
Figura 9: Módulo da saída no arranjo de sensores com o *Delay-sum beamformer* ao variar o número de microfones.

(ângulos de chegada =  $115^\circ$  e  $0^\circ$ , distância entre os sensores = 0.2 metros)



Fonte: Script MATLAB do autor.

Figura 10: Potência normalizada no *Delay-sum beamformer* ao variar a distância entre sensores  
(ângulos de chegada =  $115^\circ$  e  $0^\circ$ , número de sensores = 10)



Fonte: Script MATLAB do autor.

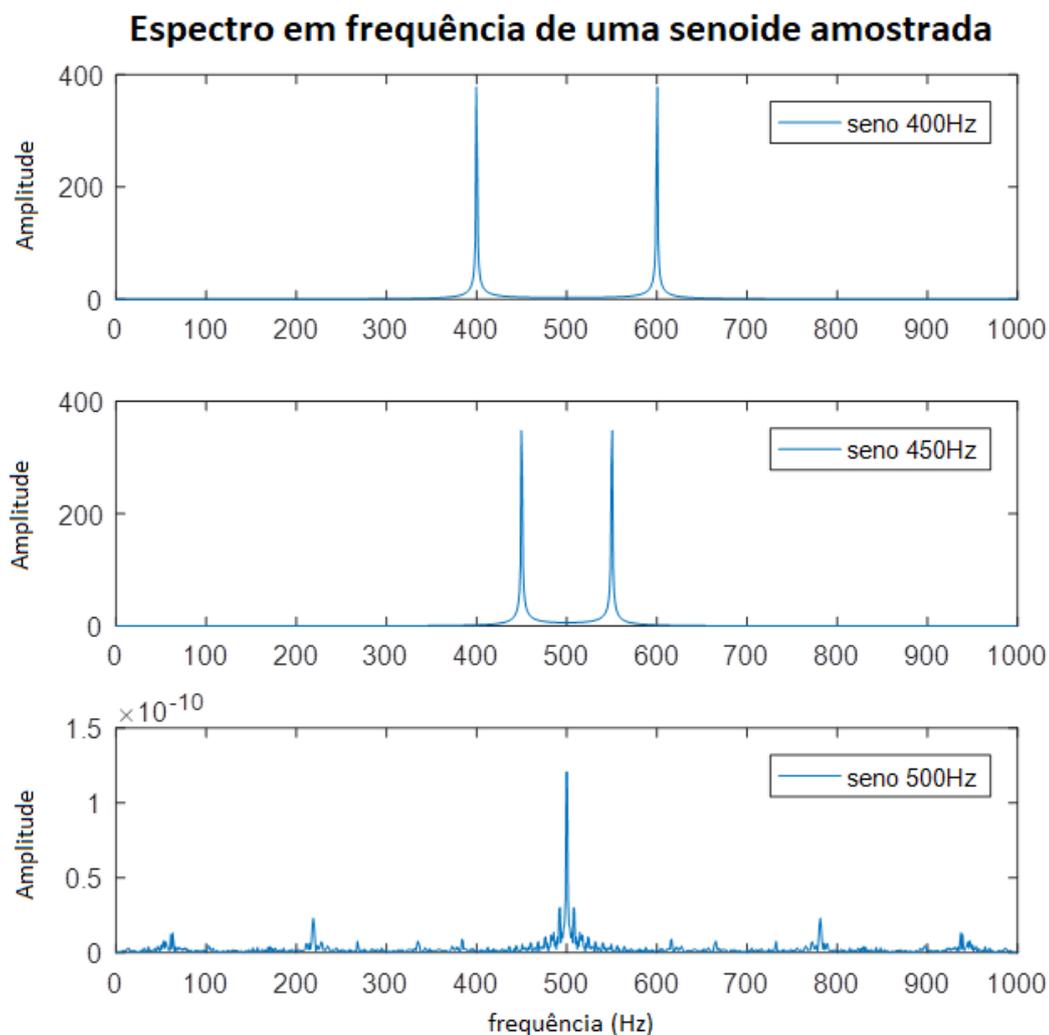
### 2.3 Amostragem

Ao amostrar um sinal, é preciso atentar a dois aspectos. São estes o *aliasing* espectral e espacial.

O *aliasing* espectral acontece quando a taxa de amostragem é menor do que duas vezes a maior componente frequência do sinal em questão. O efeito espectral da amostragem é repetir o espectro do sinal em múltiplos inteiros da frequência de amostragem. Desta forma o

*aliasing* espectral ocorre quando a versão repetida do espectro sobrepõe o espectro centrado do sinal, como se observa na Figura 11.

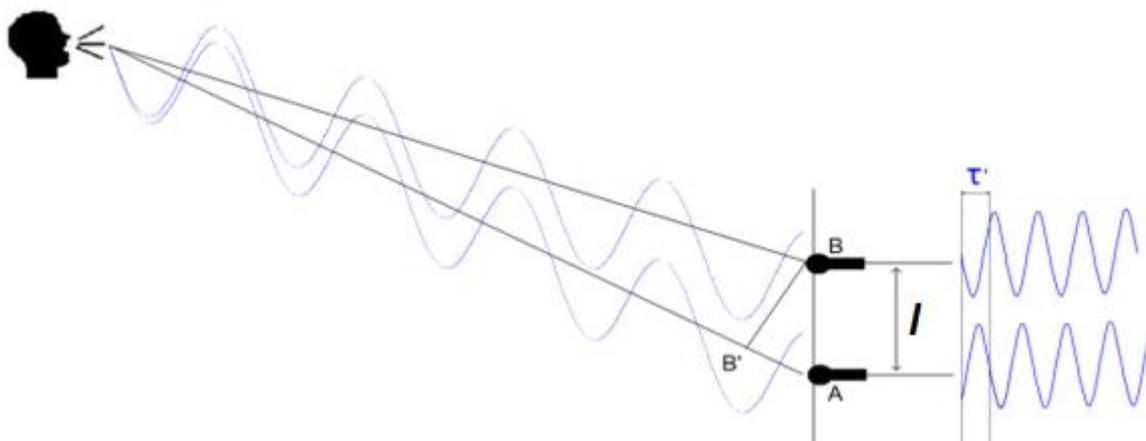
Figura 11: Espectro em frequência de uma senoide amostrada a uma taxa de 1000 amostras/segundo



Fonte: Próprio autor

Um segundo aspecto que deve ser considerado é o *aliasing* espacial, que ocorrerá quando a distância entre os sensores for menor do que a metade do comprimento de onda da onda amostrada. A Figura 12 exemplifica esta condição.

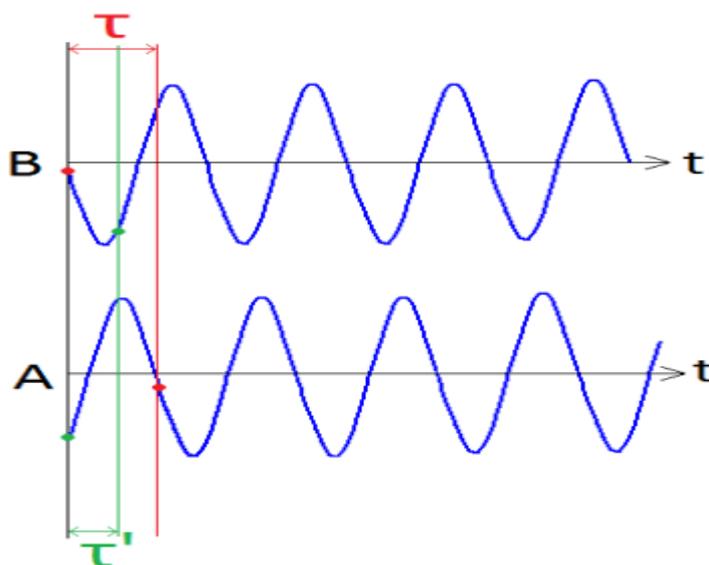
Figura 12: Arranjo de sensores recebendo um sinal em um cenário em que  $\lambda/2 < l < \lambda$



Fonte: SCOLA, C. ORTEGA, M, (2010).

Ao contabilizar os sinais amostrados pelo microfone, é possível mostrar que o deslocamento detectado não será o deslocamento real (SCOLA C. e ORTEGA M., 2010). Na Figura 13 é retratado o deslocamento detectado (errôneo) e o deslocamento real.

Figura 13: Deslocamentos discrepantes  $\tau$  (real) e  $\tau'$  (detectado).



Fonte: SCOLA, C. ORTEGA, M, (2010).

De acordo com a Figura 13, por inspeção visual, verifica-se que o deslocamento real é o marcado em vermelho. O orador estará mais próximo do microfone B, logo o sinal chegará primeiramente em B. Entretanto, o sistema irá detectar o deslocamento indicado pela cor

verde, representando o primeiro deslocamento detectável (menor do que  $\lambda/2$ ), e inferindo erroneamente. Portanto, para o caso da Figura 13, ao observar sinais em um arranjo cuja distância entre os sensores é maior do que  $\lambda/2$ , conclui-se que o orador está mais próximo do microfone A, em relação ao microfone B, o que é falso. Assim, é preciso garantir que  $l \leq \frac{\lambda}{2}$ , para que não ocorra o fenômeno de *aliasing* espacial.

#### 2.4 Considerações sobre as simulações

A primeira consideração refere-se à largura de banda do sinal. Apesar do sinal trabalhado ter uma largura de banda que se estende de 300 a 3400 Hz, o sinal será definido em uma única frequência (premissa da banda estreita), sendo necessária a adaptação dos algoritmos posteriormente, da forma como é feita nas referências encontradas. (SILVEIRA, M. *et al.*, 2013). Na Equação (7) o sinal que chega aos  $M$  sensores é escrito considerando que este poderá ser composto de uma parcela devido à fonte de sinal  $d$   $s_d[n] = e^{jn\omega}$  (no índice de amostragem  $n$  e com frequência angular  $\omega$ ) e uma segunda parcela, ao ruído.

$$\mathbf{x}[n] = \begin{bmatrix} 1 \\ e^{j\omega\tau} \\ e^{j\omega 2\tau} \\ e^{j\omega 3\tau} \\ \vdots \\ e^{j\omega M\tau} \end{bmatrix} s_d[n] + \mathbf{ruído}[n], \quad (7)$$

em que  $\mathbf{x}[n]$  é um vetor  $M \times 1$  contendo as amostras recebidas dos  $M$  sensores no instante de amostragem  $n$ . O vetor que multiplica  $s_d[n]$  representa os deslocamentos de um mesmo sinal fonte ao percorrer o arranjo de sensores e  $\tau$  é o deslocamento ao qual o sinal  $s_d[n]$  é submetido entre um sensor e o seu consecutivo (podemos obter a forma analítica do sinal amostrado  $A \cos(\omega t + \theta) = \text{Re}\{Ae^{i\omega t + \theta}\}$ , com a transformada de Hilbert do sinal banda estreita para obter uma exponencial complexa da frequência em questão). Como serão adotadas estas definições para este trabalho, o sinal de som poderá ser escrito como uma exponencial complexa com uma das seguintes abordagens:

- a) Identificação da frequência fundamental:

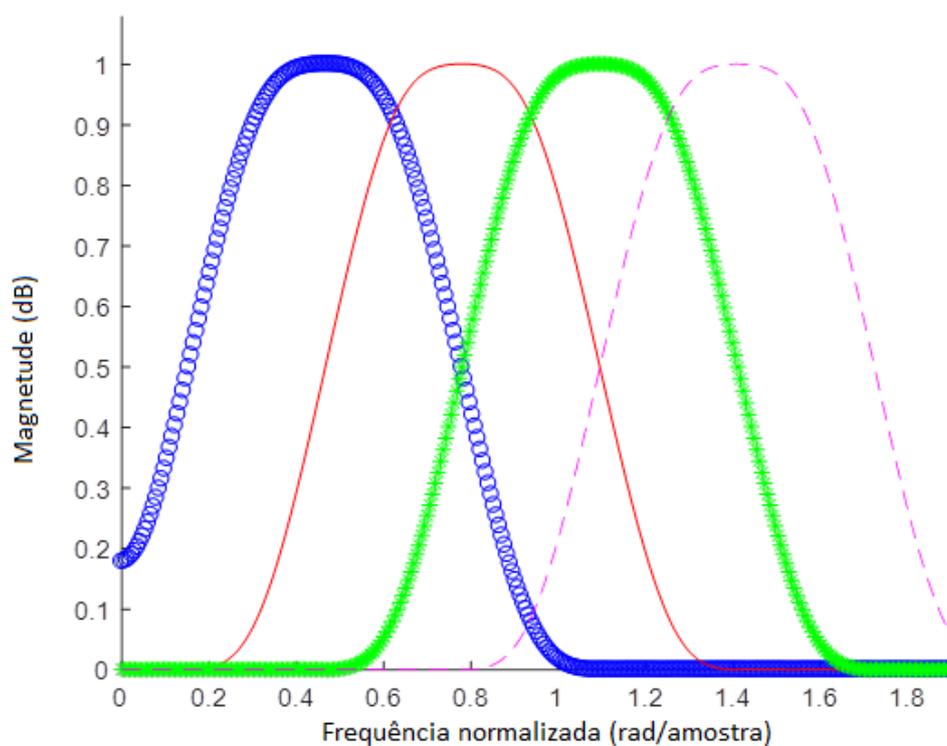
É possível identificar a frequência fundamental do quadro de amostras usando técnicas como a auto correlação do sinal ou a transformada *cepstral* do sinal (a transformada inversa de

Fourier do logaritmo do quadrado da transformada de Fourier do sinal, ou seja,  $F^{-1}\{\log(|F\{f(t)\}|)\}$ , onde  $F^{-1}(f)$  é a transformada inversa de Fourier de  $f$ ) para em seguida filtrar o sinal do quadro com um filtro passa banda. Sua saída será constituída pela frequência fundamental em seu espectro, facilitando a sua escrita na forma de uma exponencial complexa como feito no início da seção 2.4. Esta abordagem se mostra problemática quando a fonte do sinal de interesse tem períodos de silêncio, dificultando a identificação da frequência fundamental.

b) Divisão do sinal em várias bandas

De forma mais geral, é possível sintetizar diversos filtros passa-banda com resposta em frequência como indicado na Figura 14. Cada um dos filtros fornecerá um sinal o qual satisfará a premissa da banda estreita, permitindo que sejam feitos os desenvolvimentos dos algoritmos com cada uma das saídas referentes aos filtros passa banda.

Figura 14: Resposta em frequência dos filtros passa faixa



Fonte: script MATLAB do autor

Para concluir, as simulações foram feitas com áudios gravados a partir do microfone interno do computador. Em seguida, foram artificialmente deslocados para formar o sinal com o qual o arranjo de sensores deve trabalhar. Foi adicionado um ruído branco de SNR = 40 dB a todos os dois sinais (uma música, uma texto falado por um homem e por uma mulher) que permitiu simular um ambiente apropriado. Um arranjo com três sensores foi utilizado nas simulações.

### 3 MVDR – (MINIMUM VARIANCE DISTORTIONLESS RESPONSE)

Apesar da simplicidade, o *Beamformer Delay-Sum* apresenta resultados não satisfatórios em um ambiente ruidoso assim como em um ambiente cujas fontes interferentes chegam nas direções dos lóbulos secundários (linha preta na Figura 8). Para reduzir a parcela da potência proveniente do ruído e sinal interferente, é necessária uma técnica que minimize a potência do sinal de saída mantendo um ganho unitário na direção de interesse. Uma das primeiras tentativas de implementar esta técnica foi apresentada por Capon J., em 1969. Este método é conhecido como MVDR (*Minimum Variance Distortionless Response*).

A abordagem realizada por Capon relaciona-se com a estrutura estatística do sinal levando-se em conta a aproximação da matriz de autocorrelação do sinal de entrada no arranjo de sensores.

Primeiramente, é feita a aproximação da matriz de autocorrelação do sinal  $\mathbf{x}[n]$

$$\begin{aligned}\mathbf{R}_{xx} &= E[\mathbf{x}[n] \cdot \mathbf{x}^H[n]], \\ \hat{\mathbf{R}}_{xx} &= \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}[n] \cdot \mathbf{x}^H[n].\end{aligned}\tag{8}$$

Sendo  $\mathbf{x}[n]$  um vetor  $M \times 1$  contendo as amostras recebidas dos  $M$  sensores no instante de amostragem  $n$ . A letra  $\mathbf{H}$  sobrescrita representa o operador hermitiano (transposto do conjugado). Foi feita esta aproximação, pois para ser aplicado o operador esperança à variável aleatória, é preciso conhecer as propriedades estatísticas desta variável e como estas propriedades não são conhecidas, aproximamos a matriz de autocorrelação com um conjunto finito de amostras.

Em seguida, a saída  $y[n]$  do arranjo de sensores é calculada com o produto interno entre o filtro e o sinal, como na Equação (9):

$$y[n] = \mathbf{w} \cdot \mathbf{x}[n], \quad (9)$$

sendo  $\mathbf{w}$  um vetor complexo  $1 \times M$  representando os pesos do filtro aplicado (a exponencial complexa representará a versão espectral do filtro cujos coeficientes são modelados como o impulso deslocado visto na seção 2.2.2 (*delay-sum beamformer*)) ao sinal chegando ao arranjo.

Nota-se que a potência de saída consiste em aplicar o operador esperança ao quadrado da Equação (9) e obter a Equação (10).

$$E[|y[n]|^2] = \mathbf{w}^H \cdot E[\mathbf{x}[n]\mathbf{x}^H[n]] \cdot \mathbf{w}. \quad (10)$$

Pela aproximação da Equação (8) referente à matriz de autocorrelação, é obtida a Equação (11):

$$P(\mathbf{w}) = E[|y[n]|^2] = \mathbf{w}^H \cdot \widehat{\mathbf{R}}_{xx} \cdot \mathbf{w}. \quad (11)$$

É desejado encontrar um  $\mathbf{w}$  que maximize a potência do sinal de saída, mantendo a restrição das Equações (12) e (13) :

$$\mathbf{w} \cdot \begin{bmatrix} 1 \\ e^{jw\tau(\theta)} \\ e^{jw2\tau(\theta)} \\ e^{jw3\tau(\theta)} \\ \vdots \\ e^{jw(M-1)\tau(\theta)} \end{bmatrix} = \mathbf{1}, \quad (12)$$

$$\mathbf{w} \cdot \mathbf{a}(\theta) = \mathbf{1}. \quad (13)$$

Sendo  $\tau(\theta) = \frac{m l \cos(\theta)}{v}$ , o tempo que a frente de ondas leva para chegar de um sensor ao próximo sensor, quando o ângulo de chegada da frente de ondas é  $\theta$ . A Equação (13) é a restrição à otimização, que indica que os pesos  $\mathbf{w}$  devem ‘inverter’ o efeito do deslocamento submetido pelas amostras no arranjo de sensores, mantendo o ganho unitário na direção  $\theta$ .

Os multiplicadores de Lagrange serão usados para obter o  $\mathbf{w}$  que maximiza a potência média. Será desenvolvido na Equação (14):

$$\Lambda(\mathbf{w}, \lambda) = P(\mathbf{w}) + \lambda_L(g(\mathbf{w}) - k), \quad (14)$$

sendo  $\lambda_L$  o multiplicador de Lagrange,  $g(\mathbf{w}) = k$  a restrição e  $P(\mathbf{w})$  a função a ser minimizada. O  $\mathbf{w}$  que nos fornece um extremo de  $P(\mathbf{w})$  é obtida a Equação (15) ao igualar o gradiente da função  $\Lambda(\mathbf{w}, \lambda)$  a zero:

$$\begin{aligned}\nabla\Lambda &= 0, \\ \nabla P(\mathbf{w}) + \lambda_L \nabla g(\mathbf{w}) &= 0, \\ 2\mathbf{R}_{xx}\mathbf{w} + \lambda_L 2\mathbf{a}(\theta) &= 0, \\ \mathbf{w} &= -\lambda_L \mathbf{R}_{xx}^{-1} \mathbf{a}(\theta).\end{aligned}\tag{15}$$

Ao substituir este  $\mathbf{w}$  na Equação (13) de restrição, será obtida a Equação (16);

$$-\lambda_L = \frac{1}{\mathbf{a}^H(\theta) \mathbf{R}_{xx}^{-1} \mathbf{a}(\theta)},\tag{16}$$

substituindo novamente na Equação (15):

$$\mathbf{w} = \frac{\mathbf{a}^H(\theta) \mathbf{R}_{xx}^{-1}}{\mathbf{a}^H(\theta) \mathbf{R}_{xx}^{-1} \mathbf{a}(\theta)}.\tag{17}$$

Serão dados então, na Equação (17), os pesos de cada um dos sensores que maximizam a potência da saída na direção  $\theta$ . É possível encontrar a potência em função do ângulo de chegada  $\theta$  substituindo estes pesos da Equação (17) em  $P(\mathbf{w})$ , é obtida a Equação (18):

$$P(\mathbf{w}) = \frac{1}{\mathbf{a}^H(\theta) \mathbf{R}_{xx}^{-1} \mathbf{a}(\theta)}.\tag{18}$$

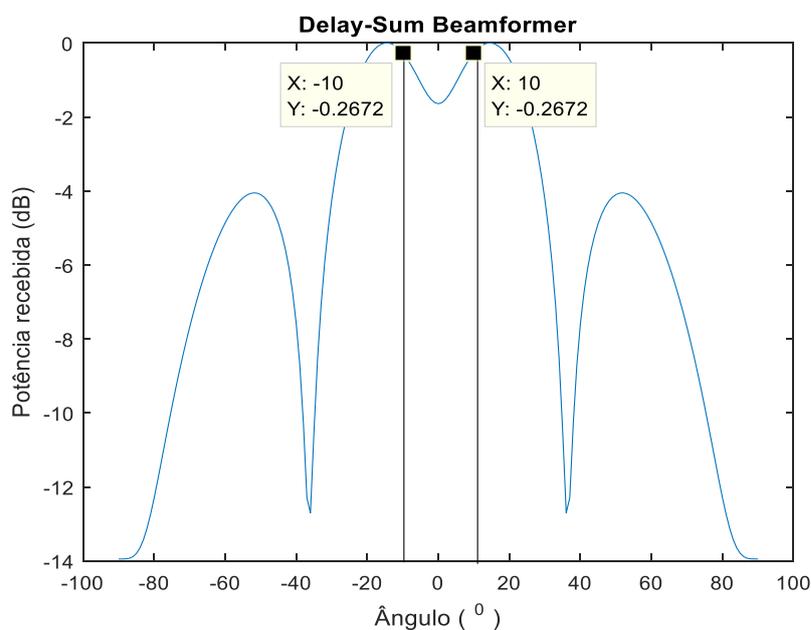
Ao utilizar esta escolha de pesos é possível mapear a potência do sinal em função da direção da chegada do sinal. Da mesma forma, é possível mapear a potência recebida em função da chegada quando os pesos escolhidos são deslocamentos (*Delay-sum Beamformer*) e compará-lo com o MVDR para encontrar a direção de chegada dos sinais.

Será simulado um cenário em que duas fontes são detectadas pelo arranjo de sensores. Será mapeada a potência em função do ângulo de chegada calculando a potência ao variar o ângulo e calculando os pesos. Os pesos do *Delay-sum Beamformer* podem ser escritos como na Equação (19).

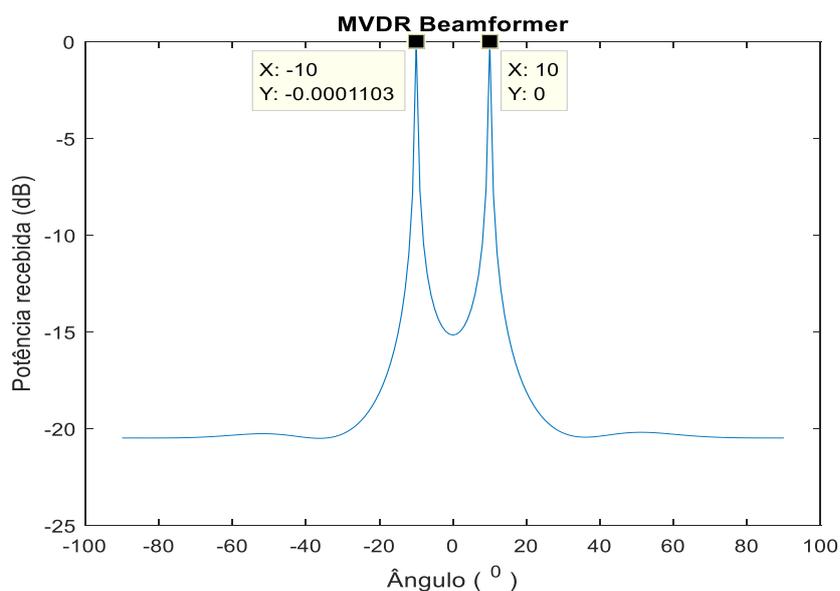
$$\mathbf{w} = \begin{bmatrix} 1 \\ e^{-jw_0 \tau(\theta)} \\ e^{-jw_0 2 \tau(\theta)} \\ e^{-jw_0 3 \tau(\theta)} \\ \vdots \\ e^{-jw_0(M-1) \tau(\theta)} \end{bmatrix}. \quad (19)$$

Estes pesos representam deslocamentos no tempo quando a frequência angular é  $w_0$ . A potência do *Delay-sum Beamformer* é calculada com  $E[|y[n]|^2]$ . No arranjo é fixado  $l = 0,2$  metros,  $M = 5$  sensores,  $w_0 = 2\pi 500$  rad/seg,  $c$  (velocidade do som) = 340,29 m/s e um SNR de 40 dB. Os resultados comparativos são mostrados nas Figuras 15 e 16.

Figura 15: Potência da saída do *Delay-Sum Beamformer*  
(ângulos de chegada =  $[-10^\circ$  e  $10^\circ$ ]).



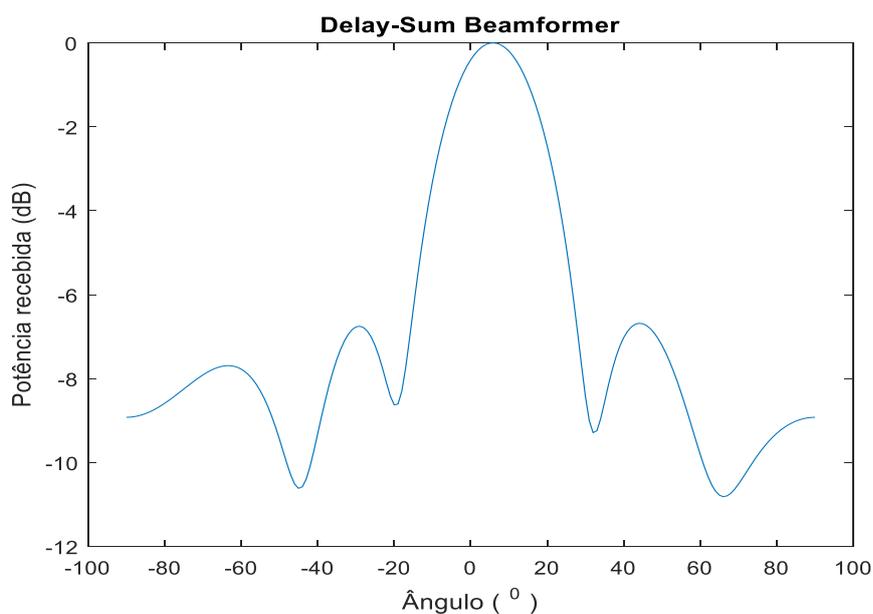
Fonte: script MATLAB do autor

Figura 16: Potência da saída do *Beamformer* MVDR

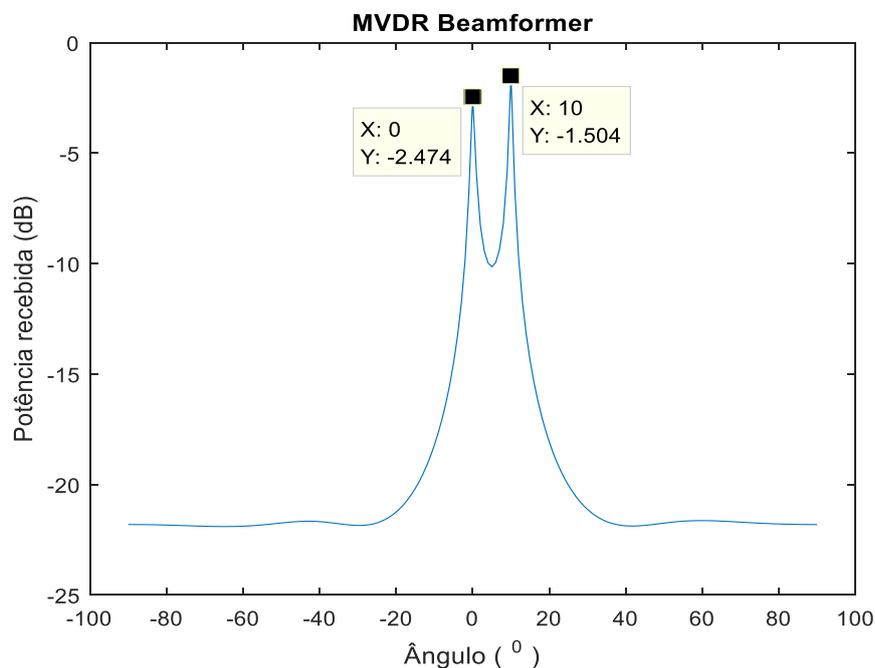
Fonte: script MATLAB do autor

Observa-se que o *Delay-Sum* apresenta baixa resolução para fontes cujos ângulos de chegada são próximos, ao passo que o MVDR se mostra bem melhor na resolução. O MVDR se mostra útil em sistemas de comunicações no qual a frequência da portadora permanece conhecida e constante. Um caso mais crítico aparece quando temos ângulos de chegada relativamente próximos como na Figura 17.

Figura 17: Potência da saída do *Delay-Sum Beamformer*  
(ângulos de chegada= [0° e 10°])



Fonte: script MATLAB do autor

Figura 18: Potência da saída do MCDR *Beamformer*

Fonte: script MATLAB do autor

Se estes *Beamformers* fossem avaliadas em sistemas radares de aviação, por exemplo, seria fácil perceber a ineficácia do *Delay-sum Beamformer* ao tentar detectar o número de fontes e suas direções, como nas Figuras 18 e 17. Nesta primeira, não seriam detectadas duas fontes (0° e 10°) enquanto que no segundo caso, é observável a presença de ambas as fontes.

Apesar de uma melhor resolução com o MVDR, não será possível restringir o sinal de chegada a sinais contendo uma só frequência (premissa da banda estreita, seção 2.4), pois nos sinais de voz é encontrada uma maior largura de banda, (seção 2.1), sendo necessárias técnicas que possam manter a alta resolução, adaptando o arranjo de antenas em um ambiente cuja banda do sinal é larga.

## 4 DIREÇÃO DE CHEGADA

Os principais algoritmos responsáveis pela adaptação do arranjo de sensores são o MVDR, o Frost e o GSC (este último será também abordado neste trabalho). Em todos estes casos é necessário que a informação sobre a direção de chegada seja conhecida de antemão para a criação do(s) filtro(s) envolvidos no processo de *Beamforming*. É possível encontrar a direção

de chegada como feito na seção precedente, comparando os *Beamformers MVDR* e o *Delay-Sum*, procurando os picos na potência do sinal ao mapeá-lo para todos os ângulos de chegada. Entretanto, esta é a forma mais rudimentar, pois exigirá muito recurso computacional, inviabilizando a criação do sistema.

Serão apresentados três métodos usados na literatura para encontrar a direção de chegada. A primeira, mais simples, é útil para sistemas com poucos recursos, baseando-se no cálculo da correlação cruzada entre os sensores. A segunda, baseada na utilização de um algoritmo adaptativo (RLS, LMS) e pode ser também utilizada como *Beamformer*. A terceira técnica consiste em utilizar a decomposição em auto-vetores dos sinais que chegam para formar o ‘espaço dos sinais’ e o ‘espaço do ruído’ e inferir sobre as direções de chegada utilizando o algoritmo MUSIC (*Multiple Signal Classification*). Este último é o mais usado em sistemas com amplos recursos computacionais, e requer um só quadro com amostras para estimar uma curva que se aproxima das curvas de potência encontradas na seção 3.

## 4.1 Correlação cruzada

### 4.1.1 Fundamentos

Neste método, dois sensores consecutivos são escolhidos e é calculada a estimação da correlação cruzada como na Equação (20) (sensores  $m$  e  $m+1$ ):

$$(x_m \otimes x_{m+1})[n] := \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} x_m[k] x_{m+1}[k+n]. \quad (20)$$

Para que esta correlação cruzada não varie rapidamente, é possível adaptá-la para que considere versões passadas da correlação cruzada como feito na Equação (21):

$$(x_m \otimes x_{m+1})[n] := \alpha \cdot (x_m \otimes x_{m+1})[n-1] + \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} x_m[k] \cdot x_{m+1}[k+n]. \quad (21)$$

Sendo  $\alpha$ , uma constante entre  $[0, 1[$ . Quando  $\alpha = 0$ , a recursão não considerará nada do passado da correlação cruzada. E se  $\alpha = \lim_{\alpha \rightarrow 1}$ , a recursão considerará o instante  $n - 1$  tanto quanto o atual.

Por este método, o ângulo de chegada será encontrado procurando pelo primeiro pico na função de correlação cruzada. Este pico representa o número de amostras em que ao deslocar o sinal do microfone  $m$ , será obtido um sinal com a maior correlação em relação ao sinal do microfone  $m+l$ .

Com o número de amostras em mãos, basta subtrair  $\frac{N}{2}$  a este número, para que seja possível a detecção de ângulos maiores do que  $90^\circ$ . O ângulo de chegada pode ser encontrado pela Equação (22) desenvolvendo a Equação (22)

$$\begin{aligned}\tau(\theta) &= \frac{m l \cos(\theta)}{v}, \\ \cos(\theta) &= \frac{\tau v}{l}, \\ \theta &= \cos^{-1} \frac{n_{amostras} F_s v}{l}\end{aligned}\tag{22}$$

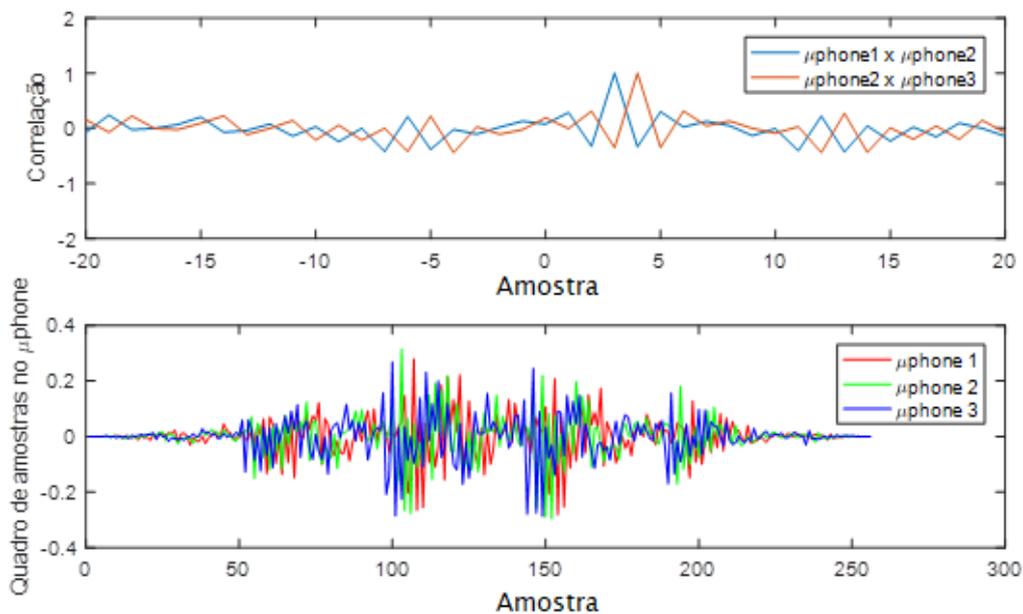
Na Equação (22),  $n_{AMOSTRAS}$  é o número de amostras após a subtração de  $\frac{N}{2}$  e  $v$  é a velocidade do som.

#### 4.1.2 Resultados

Como foram utilizados três sensores, o ângulo de chegada final foi encontrado a partir de uma média ponderada entre o máximo da correlação entre os sensores um e dois e entre os sensores dois e três. O peso da média ponderada será o valor da correlação naquele máximo. Isto é feito para aumentar a resolução, uma vez que o número de amostras deslocadas varia de acordo com a Equação (22).

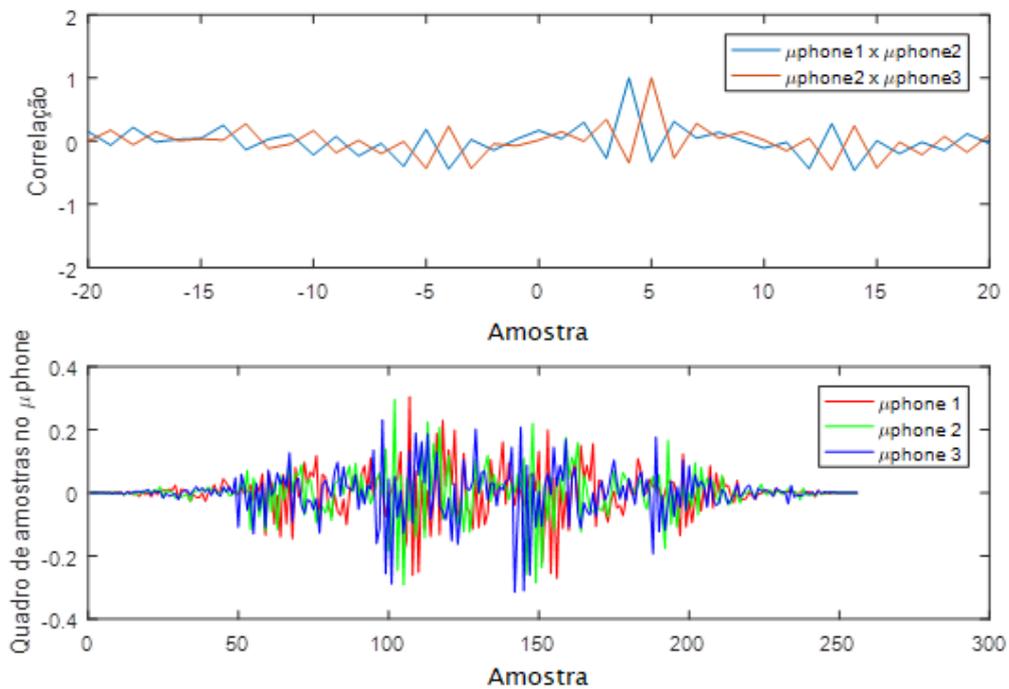
Quando  $l = 0,2$  metros,  $F_s = 8000$  amostras/segundo e  $v = 340,29$  metros/ segundo observa-se que esta relação variará de  $-5$  amostras até  $5$  amostras (número inteiro mais próximo). Portanto, se apenas dois sensores estivesse disponíveis, a resolução ficaria prejudicada, pois com um alcance de  $11$  amostras ( $-5$  até  $5$  amostras de deslocamento) seria necessário mapear os ângulos de chegada de  $0$  até  $180^\circ$ . Desta forma, mais sensores e uma média ponderada aumentará precisão. As Figuras 19, 20 e 21 mostram resultados das simulações aonde uma voz masculina é usada como teste.

Figura 19: Correlação cruzada de um quadro de amostras (ângulo de chegada=50°)



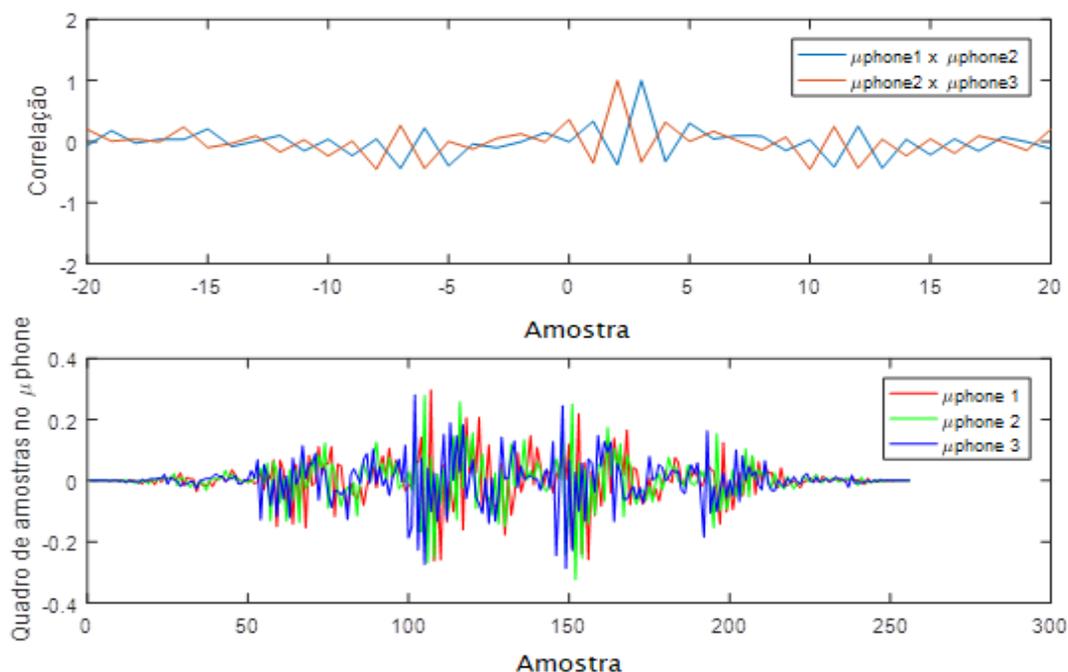
Fonte: Script MATLAB do autor.

Figura 20: Correlação cruzada de um quadro de amostras (ângulo de chegada=90°)



Fonte: Script MATLAB do autor

Figura 21: Correlação cruzada de um quadro de amostras (ângulo de chegada = 30°)

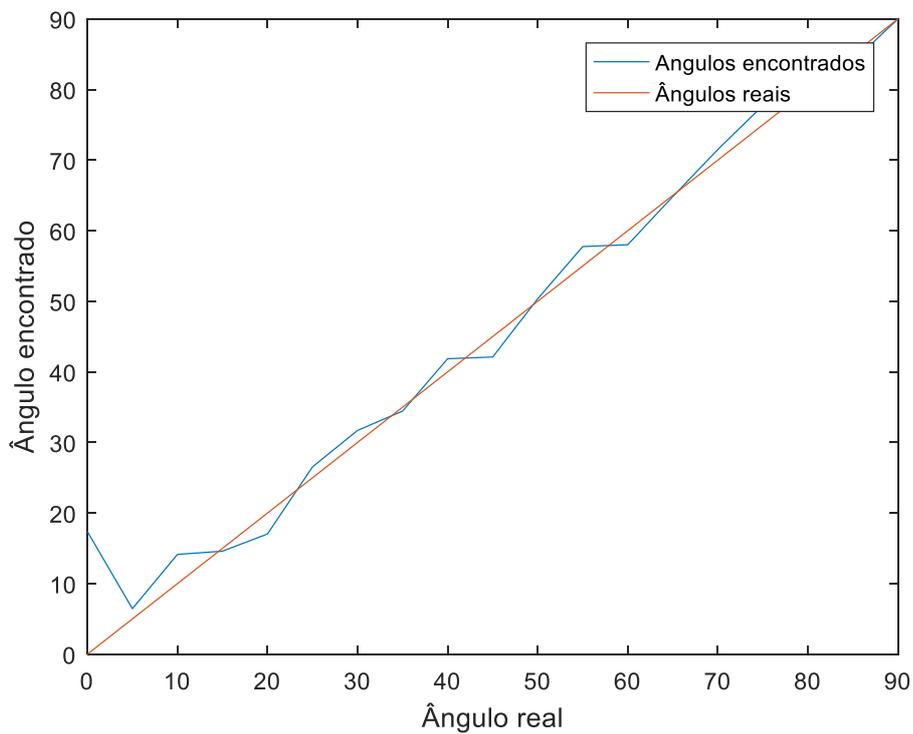


Fonte: Script MATLAB do autor

É interessante notar nas Figuras 19, 20 e 21 o efeito na correlação quando os ângulos de chegada têm valores que não seriam detectáveis com apenas dois microfones. Através das duas funções de correlação foi possível aumentar a precisão das medidas de ângulo de chegada usando o valor da correlação dos dois picos. Percebe-se nos gráficos representados nas Figuras 22 e 23, o resultado de diversas simulações para a descoberta do ângulo de chegada quando é trabalhado com vozes masculina e feminina.

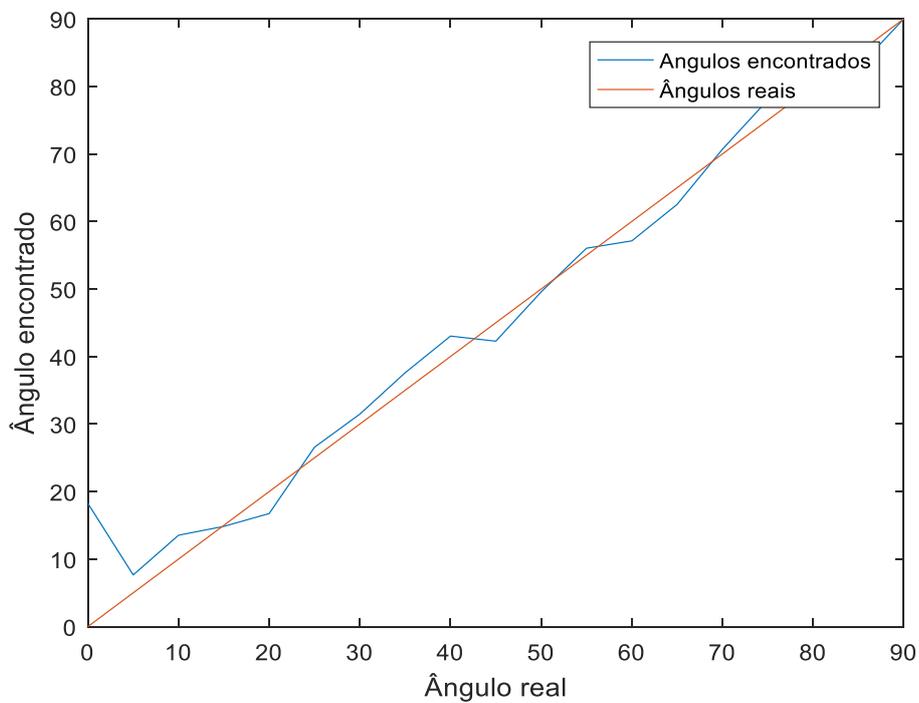
Ao avaliar a forma das curvas nas Figuras 22 e 23 é possível notar que suas formas tendem a se assemelhar com degraus. Embora a estimação do ângulo de chegada com duas funções de correlação apresente resultados satisfatórios, não é possível desvincular estes resultados aos deslocamentos discretos nos sinais entre os microfones, o que confere o aspecto de vários degraus às curvas.

Figura 22: Direções de chegada encontradas com voz masculina.



Fonte: Script MATLAB do autor

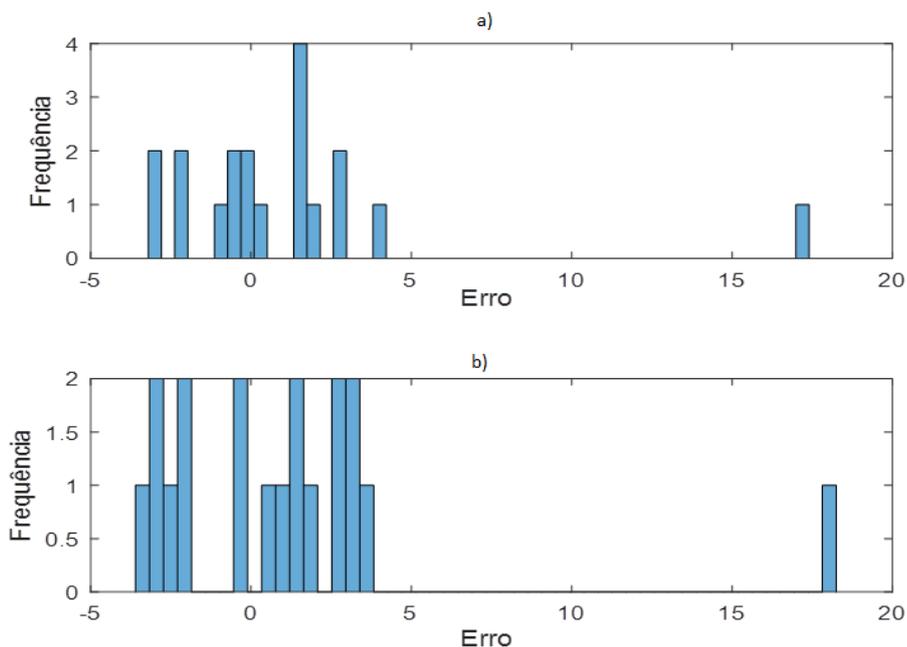
Figura 23: Direções de chegada encontradas com voz feminina



Fonte: Script MATLAB do autor

Erros em relação ao ângulo real foram medidos e seus histogramas estão na Figura 24:

Figura 24: Histograma dos erros na identificação do ângulo de chegada. a) voz masculina. b) voz feminina.



Fonte: Script MATLAB do autor

Os ângulos próximos de  $0^\circ$  graus se mostraram difíceis de serem identificados, pois a quantidade de amostras deslocadas não varia suficientemente, consequência da Equação (4), cuja derivada disporá de um módulo mínimo quando  $\theta = 0^\circ$ .

## 4.2 Algoritmo adaptativo

### 4.2.1 Fundamentos

Esta implementação baseia-se na publicação de Scola E Ortega (2010), na qual é utilizado o algoritmo LMS em um arranjo com dois microfones.

A parte adaptativa do sistema consiste em um filtro a cada um de dois microfones adjacentes.

O algoritmo do mínimo erro quadrático médio (LMS) foi inventado pelo professor de Stanford Bernard Widrow e o estudante de PHD Ted Hoff em 1960. Ele é usado para calcular os coeficientes do filtro que permite obter o valor mínimo do sinal do erro quadrático. Este

signal de erro é definido como a diferença entre o sinal desejado e o sinal de saída. O LMS pertence à família de algoritmos de gradiente descendente estocástico que busca, de forma iterativa, encontrar o filtro ótimo com base no erro calculado no instante presente. Este algoritmo não exige inversões de matriz, por isso é relativamente simples.

No algoritmo LMS, são utilizados dois vetores que representam o sinal  $\mathbf{x}'[n]$  e um filtro  $\mathbf{w}$  (de tamanho  $P$ , sendo este a ordem do filtro). O valor referência é dado por  $d[n]$  e sua saída  $y[n] = \mathbf{x}'[n]\mathbf{w}$ . As etapas do algoritmo LMS são descritas a seguir:

a) Processo de filtragem

$\mathbf{w}$  é iniciado com  $P$  valores arbitrários. A cada instante de tempo  $n$  será organizada as  $P$  últimas amostras do sinal no vetor  $\mathbf{x}$  da como na Equação (23):

$$\mathbf{x}'[n] = [x[n], x[n-1], x[n-2], \dots, x[n-P+1]]. \quad (23)$$

A saída do sistema ( $y'[n]$ ) será calculada na Equação (24):

$$y'[n] = \mathbf{w}^H \cdot \mathbf{x}'[n], \quad (24)$$

no qual  $\mathbf{w}^H$  é a hermitiana (transposto do conjugado) de  $\mathbf{w}$ . Em seguida, o erro em relação ao sinal destino ( $d[n]$ ) é calculado na Equação (25):

$$e = d[n] - y'[n]. \quad (25)$$

b) Processo adaptativo

Os novos pesos são em seguida calculados na Equação (26), baseados nos anteriores e no valor do erro.

$$\mathbf{w} = \mathbf{w} + \mu \cdot \mathbf{x}'[n] \cdot e. \quad (26)$$

As correções sucessivas do peso levarão eventualmente o erro ao valor mínimo.

$\mu$  é chamado de passo do algoritmo e é um parâmetro de grande importância, pois este controla a convergência do algoritmo, se mostrando fundamental em um sistema de tempo real.

Kwok E Buchholz (2005) mostraram que ao obter o maior autovalor  $\lambda_{max}$  da matriz de autocorrelação  $\mathbf{R}_{xx}[n] = E[\mathbf{x}'[n]\mathbf{x}'^H[n]]$ , podem ser encontrados valores para  $\mu$  com o qual a convergência é garantida. Assim,  $\mu$  deve satisfazer a Equação (27):

$$0 < \mu < \frac{1}{\lambda_{max}}. \quad (27)$$

Para aplicar o algoritmo ao nosso sistema, basta considerar a Equação (28):

$$\mathbf{x}'^{[n]} = [x_{m+1}[n], x_{m+1}[n-1], x_{m+1}[n-2], \dots, x_{m+1}[n-P+1]]. \quad (28)$$

Foi preciso adaptar os sinais advindos dos sensores  $x_m[n]$  e  $x_{m+1}[n]$  para que não houvesse o problema do ‘deslocamento negativo’. Este problema aparece, pois ao utilizar o filtro em um só microfone, seria suposto de antemão que a fonte de interesse emite som de um ângulo  $\theta$  alcançando primeiramente o sensor  $m$  e em seguida o sensor  $m+1$ . Esta suposição é falsa, uma vez que não é conhecido de onde vem o sinal.

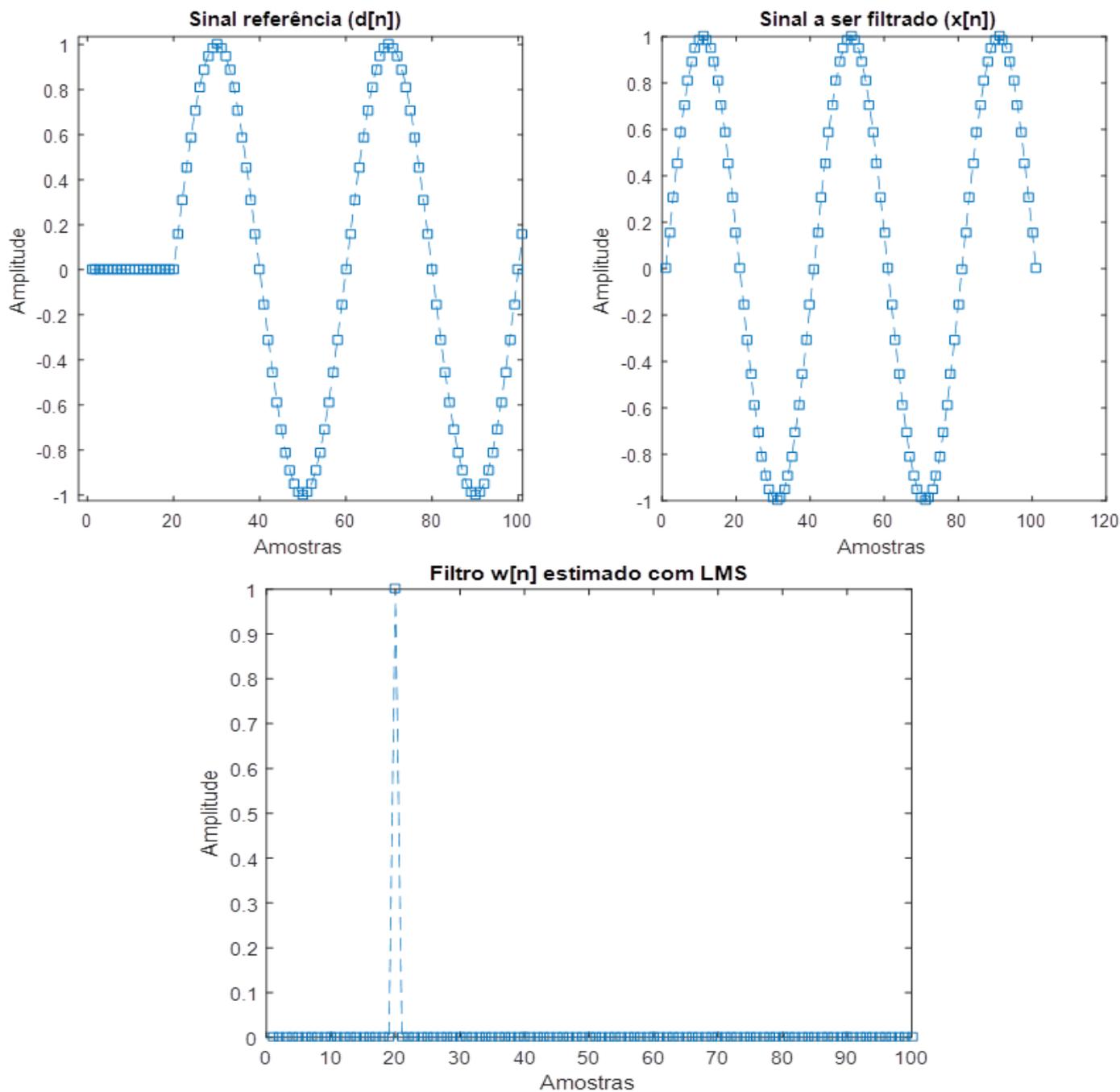
Para evitar esta situação,  $x_m[n]$  será deslocado antes da filtragem adaptativa, de forma que nunca haja deslocamentos negativos. Aplicando a convolução à  $x_m[n]$ , é obtido a Equação (29)

$$x''_m[n] = x_m[n] * [0,0, \dots, 0,1,0, \dots, 0,0]. \quad (29)$$

A quantidade a deslocar o sinal  $x_m[n]$  foi definida encontrando o valor máximo da equação  $\tau(\theta).Fs = \frac{m l \cos(\theta)}{v}.Fs$ , para  $m=l$ . Novamente,  $\tau(\theta)$  é o tempo em que a frente de ondas leva para chegar ao sensor  $m$  ao atingir o sensor zero.  $Fs$  é a taxa de amostragem. O máximo da equação ocorrerá quando  $\cos(\theta) = 1$  e seu mínimo, quando  $\cos(\theta) = -1$ .

Após o processo adaptativo, será encontrado um vetor  $\mathbf{w}$  cuja forma deverá idealmente se assemelhar a um impulso unitário deslocado, conforme se observa na Figura 25.

Figura 25: Sinais teste para o filtro LMS e o filtro obtido



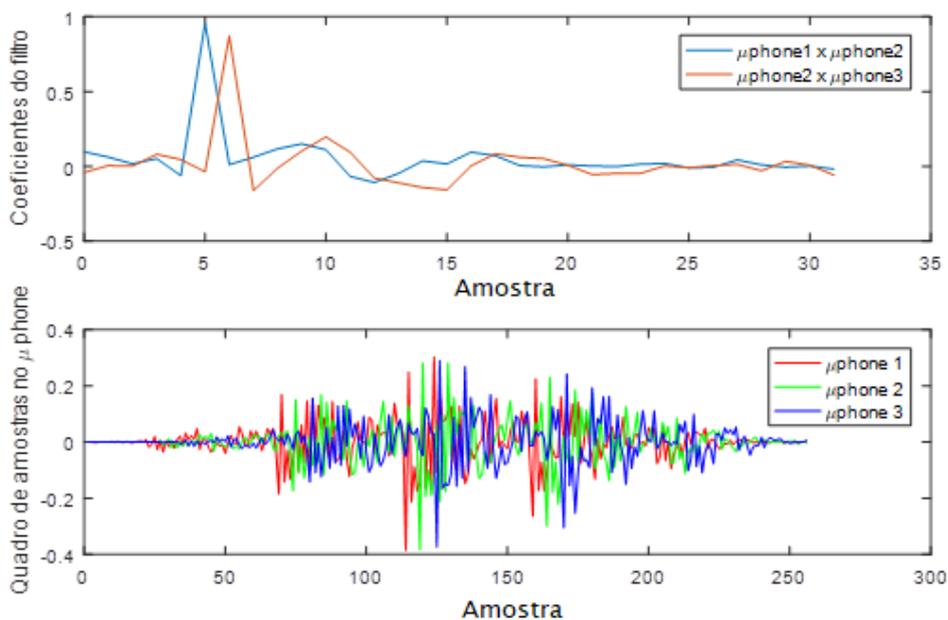
Fonte: script MATLAB do autor

#### 4.2.2 Resultados

Novamente, três sensores serão utilizados, e o ângulo de chegada final será encontrado a partir de uma média ponderada entre o máximo dos dois filtros encontrados entre os sensores um e dois e entre os sensores dois e três, respectivamente. O peso da média ponderada será o valor do peso do filtro naquele máximo. As Figuras 26, 27, 28 e 29 mostram resultados de

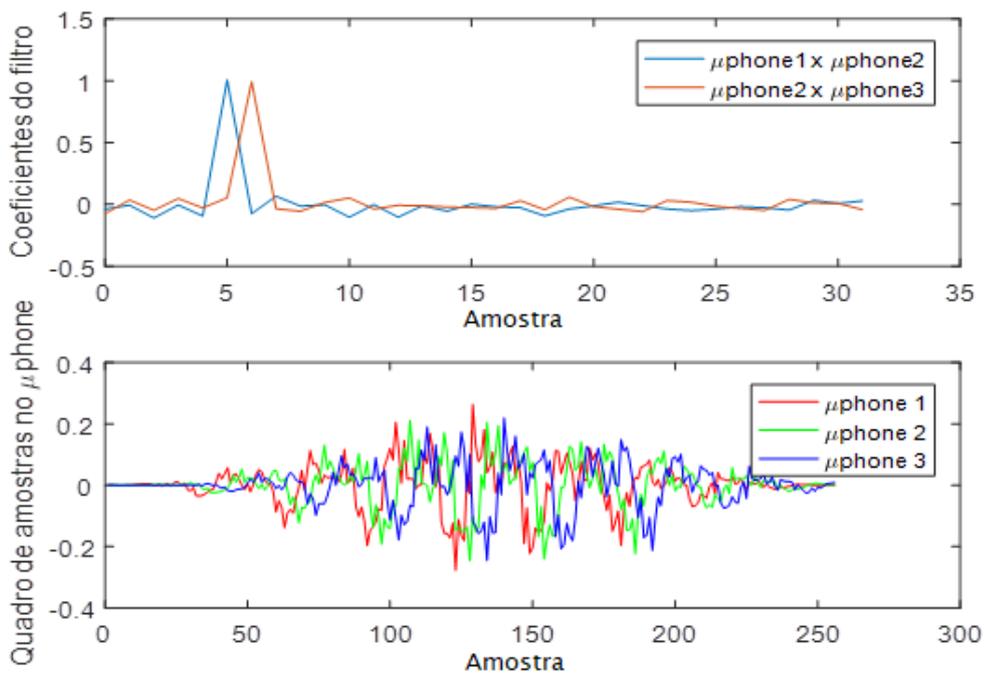
simulações com uma voz masculina e uma voz feminina. Em todos os casos está presente uma música como ruído cuja SNR é de 40dB. O passo do algoritmo  $\mu$  foi fixado em 0.1.

Figura 26: Filtro adaptativo obtido de um quadro de amostras (ângulo de chegada=90°, voz masculina).



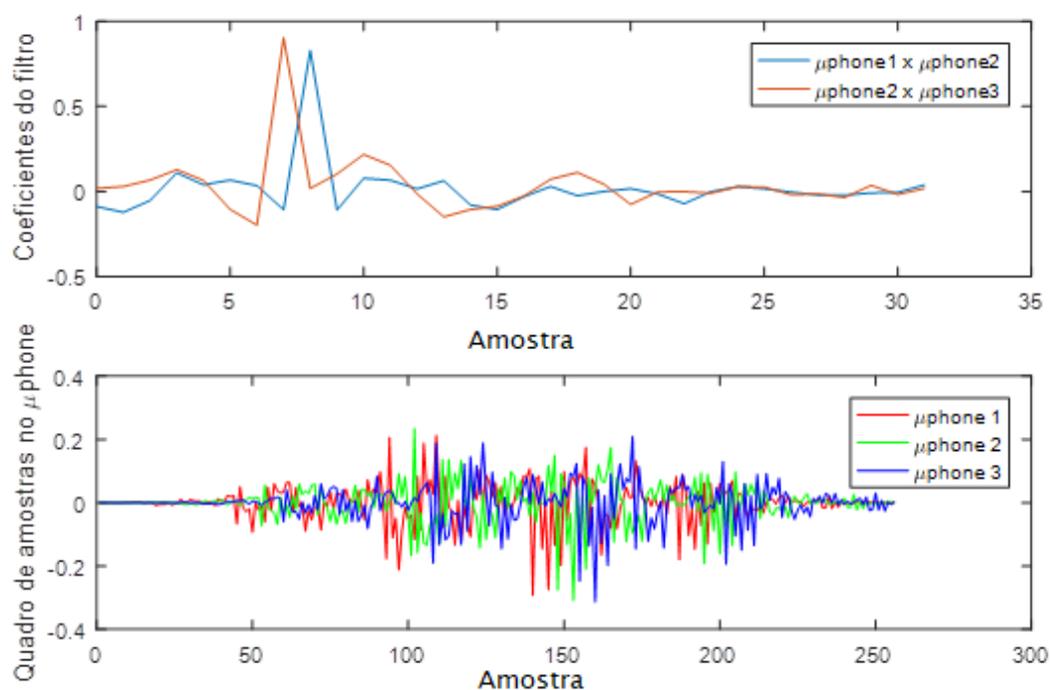
Fonte: script MATLAB do autor

Figura 27: Filtro adaptativo obtido de um quadro de amostras (ângulo de chegada=90°, voz feminina).



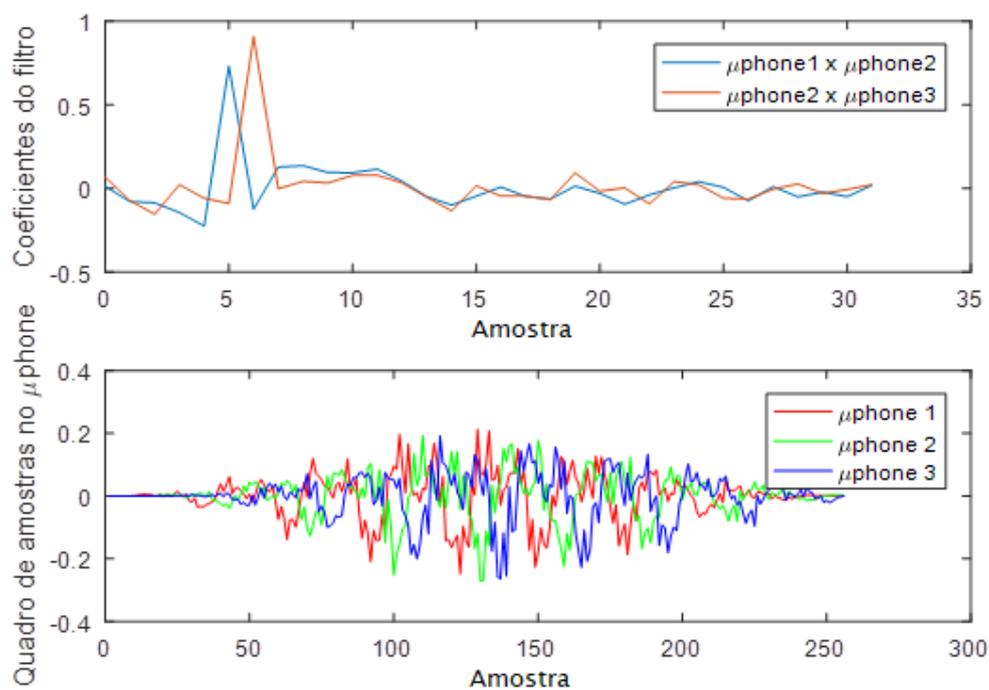
Fonte: script MATLAB do autor

Figura 28: Filtro adaptativo obtido de um quadro de amostras (ângulo de chegada=30°, voz masculina).



Fonte: script MATLAB do autor

Figura 29: Filtro adaptativo obtido de um quadro de amostras (ângulo de chegada=30°, voz feminina).

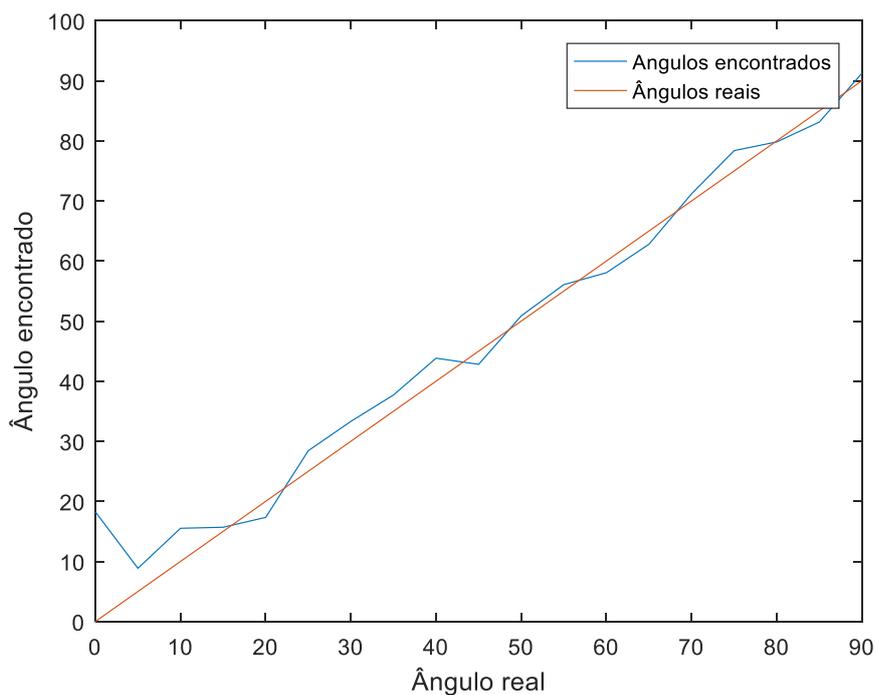


Fonte: script MATLAB do autor

Constata-se a partir das Figuras 26, 27, 28 e 29, que a forma com a qual o deslocamento entre o sinal de um microfone e outro se comporta é bastante parecida com os resultados da seção precedente (Seção 4.1.2). A diferença, entretanto, se mostra ao perceber que na medida em que o ângulo aumenta, o deslocamento identificado pelo máximo na função dos pesos do filtro diminui. Isto ocorre porque é deslocado os sinais a serem usados no algoritmo. Desta forma o deslocamento deverá ser subtraído ao máximo da função dos coeficientes do filtro deslocada de  $2 \cdot \frac{m l \cos(\theta)}{v} \cdot Fs$ , quando  $\cos(\theta) = 1$ , o que forneceria para este caso, 10 amostras.

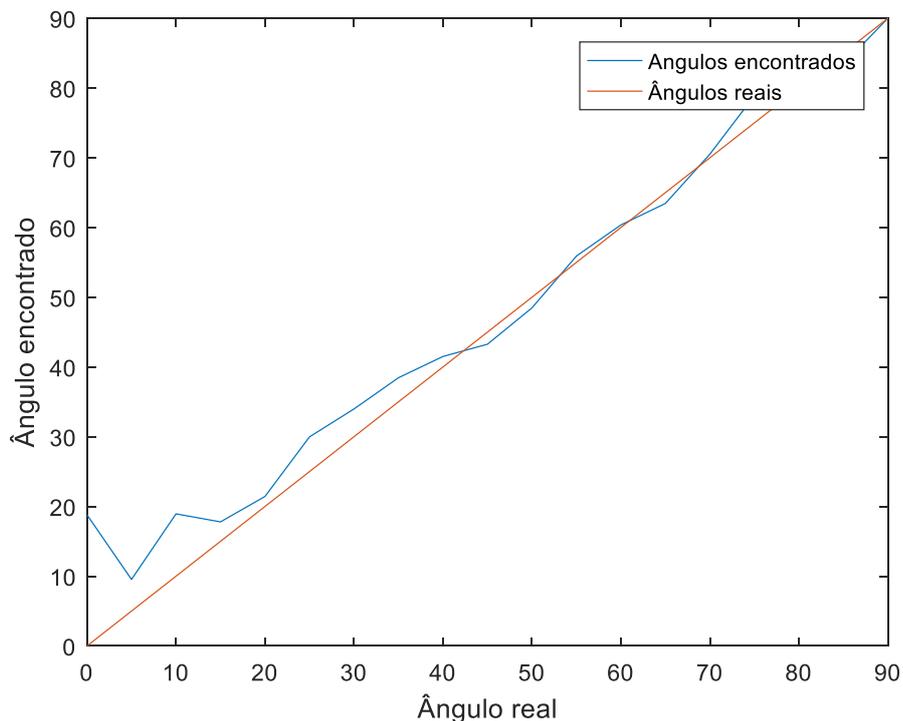
Nos gráficos das Figuras 30 e 31, os resultados de diversas simulações para a descoberta do ângulo de chegada são mostrados em função do ângulo de chegada real.

Figura 30: Direções de chegada encontradas (voz masculina).



Fonte: script MATLAB do autor

Figura 31: Direções de chegada encontradas (voz feminina).



Fonte: script MATLAB do autor

Como na seção precedente, os ângulos próximos de  $0^\circ$  graus se mostraram difíceis de serem identificados, devido a não variação do deslocamento das amostras em ângulos próximos de  $0^\circ$ .

### 4.3 MUSIC (*Multiple Signal Classification*)

#### 4.3.1 Fundamentos

Este algoritmo usa uma decomposição da matriz de covariância em duas matrizes ortogonais, sendo uma respectiva ao subespaço dos sinais e outra do subespaço dos ruídos. Utilizando a priori um sinal contendo uma faixa de frequência, a partir das considerações da seção 2.4 e desconsiderando a ação do ruído, é possível definir a matriz  $\mathbf{A}_s$  na Equação (30):

$$\mathbf{A}_s = [\mathbf{a}(\theta_{sinal\ 1}), \mathbf{a}(\theta_{sinal\ 2}), \dots, \mathbf{a}(\theta_{sinal\ d-1})], \quad (30)$$

sendo  $\mathbf{a}(\theta)$ , definido na Equação (10), o deslocamento pelo qual o sinal da fonte de interesse é submetido ao chegar aos  $M$  microfones.

O sinal desejado recebido no sensor é dado pela Equação (31):

$$\mathbf{X}_s[n] = \mathbf{A}_s \mathbf{S}_s[n], \quad (31)$$

sendo  $\mathbf{X}_s[n]$  uma matriz  $d \times M$  contendo o sinal recebido nos  $M$  microfones advindo das  $d$  fontes. Será definido também os sinais interferentes na Equação (32):

$$\mathbf{X}_I[n] = \mathbf{A}_I \mathbf{S}_I[n]. \quad (32)$$

O sinal total que chega ao arranjo de sensores é composto pelo sinal desejado somado às interferências e o ruído:

$$\mathbf{X}[n] = \mathbf{X}_s[n] + \mathbf{X}_I[n] + \text{ruído}[n]. \quad (33)$$

Será assumido agora que os sinais interferentes, os sinais das fontes e do ruído são descorrelacionados, Como indicado nas Equações (34), (35) e (36):

$$E[\mathbf{X}_s[n] \mathbf{X}_I^H[n]] = 0, \quad (34)$$

$$E[\mathbf{X}_s[n] \text{ruído}^H[n]] = 0, \quad (35)$$

$$E[\mathbf{X}_I[n] \text{ruído}^H[n]] = 0. \quad (36)$$

Desta forma, a matriz de correlação do sinal  $\mathbf{X}[n]$  será encontrada usando as equações (32) e (31) e desenvolvendo a Equação (37).

$$\begin{aligned} \mathbf{R}_{xx} &= E[\mathbf{X}[n] \mathbf{X}^H[n]] \\ \mathbf{R}_{xx} &= \mathbf{A}_s \mathbf{R}_{ss} \mathbf{A}_s^H + \mathbf{A}_I \mathbf{R}_{II} \mathbf{A}_I^H + \sigma_n^2 \mathbf{I}, \end{aligned} \quad (37)$$

na qual,  $\mathbf{R}_{ss} = E[\mathbf{S}_I[n] \mathbf{S}_I^H[n]]$  e  $\sigma_n^2$  é a variância do ruído, representado por um ruído branco. É possível assim encontrar sua decomposição em autovalores e autovetores:

$$\mathbf{R}_{xx} \mathbf{v}_i = \lambda_i \mathbf{v}_i \quad i = 1, 2, \dots, M. \quad (38)$$

Organizando os autovalores de  $\mathbf{R}_{xx}$  de ordem decrescente teremos  $M-d$  menores autovalores correspondentes ao ruído, e  $d$  maiores autovalores respectivos aos sinais.

A Equação (37) é em seguida simplificada para que contenha apenas termos de ruído e sinal, sendo os sinais interferentes, parte do sinal. Utilizando assim, os autovalores  $\lambda_i$  da Equação (38) e inserindo a Equação (37), será obtida a Equação (39):

$$\mathbf{R}_{xx} \mathbf{v}_i = \lambda_i \mathbf{v}_i = (\mathbf{A}_s \mathbf{R}_{ss} \mathbf{A}_s^H + \sigma^2 \mathbf{I}) \mathbf{v}_i. \quad (39)$$

Restringindo  $i = d + 1, d + 2, \dots, M$ , a equação de  $\mathbf{R}_{xx}\mathbf{v}_i$  é reduzida a conter os autovetores/autovalores referentes ao ruído, de forma que a obtemos a Equação (40):

$$\begin{aligned}\sigma^2\mathbf{v}_i &= (\mathbf{A}_s\mathbf{R}_{ss}\mathbf{A}_s^H + \sigma^2\mathbf{I})\mathbf{v}_i, \\ \mathbf{A}_s\mathbf{R}_{ss}\mathbf{A}_s^H\mathbf{v}_i &= \mathbf{0}.\end{aligned}\quad (40)$$

Multiplicando ambos os lados da equação por  $\mathbf{R}_{ss}^{-1}(\mathbf{A}_s\mathbf{A}_s^H)^{-1}\mathbf{A}_s^H$ :

$$\begin{aligned}\mathbf{R}_{ss}^{-1}(\mathbf{A}_s\mathbf{A}_s^H)^{-1}(\mathbf{A}_s^H\mathbf{A}_s)\mathbf{R}_{ss}\mathbf{A}_s^H\mathbf{v}_i &= \mathbf{0}, \\ \mathbf{A}_s^H\mathbf{v}_i &= [\mathbf{a}(\theta_{\text{sin}al\ 1}), \mathbf{a}(\theta_{\text{sin}al\ 2}), \dots, \mathbf{a}(\theta_{\text{sin}al\ d-1})]^H\mathbf{v}_i = \mathbf{0}.\end{aligned}\quad (41)$$

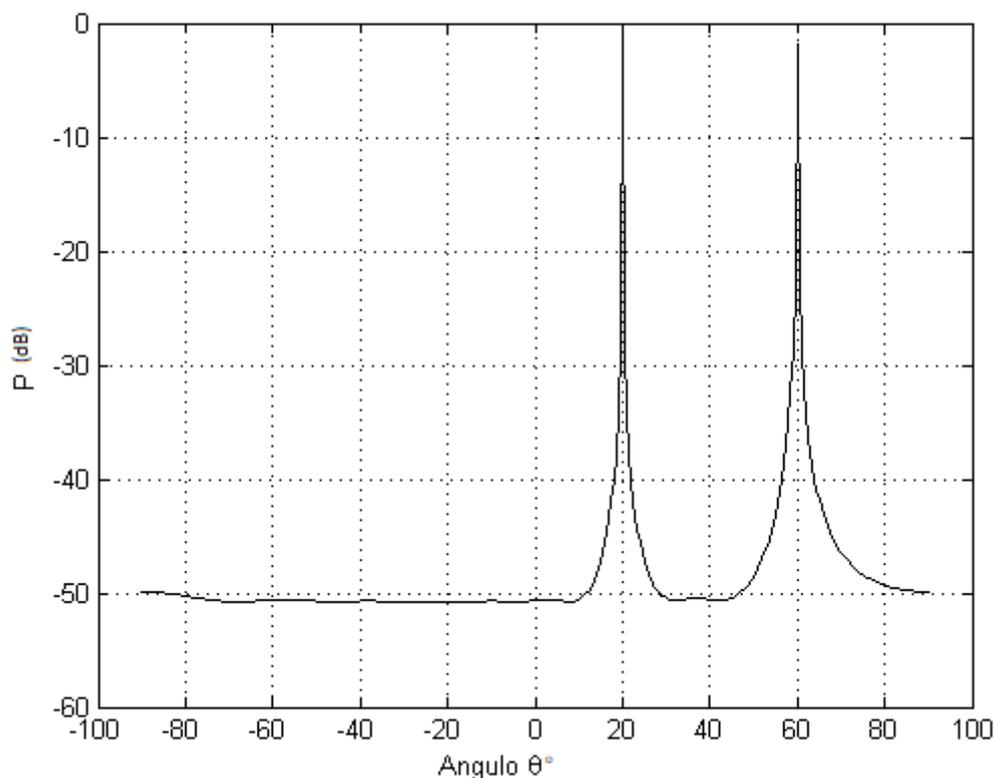
A equação (41) indica que o autovetor dos ruídos é ortogonal às colunas da matriz  $\mathbf{A}_s^H$  (o espaço dos sinais é ortogonal ao espaço dos ruídos). Sendo assim, é possível definir a Equação (42), chamada de *pseudoespectro*.

$$\mathbf{P}(\theta) = \frac{1}{\|\mathbf{v}_i^H\mathbf{A}_s(\theta)\|^2}.\quad (42)$$

Quando  $\mathbf{A}_s(\theta)$  é ortogonal à  $\mathbf{v}_i$ , o resultado do produto interno é zero, gerando um mínimo (não será zero, pela existência do ruído), criando assim, um pico em  $\mathbf{P}(\theta)$ . É vista a aplicação desta equação na Figura 32. Os parâmetros são descritos na legenda da Figura 32.

Figura 32: Estimação da direção de chegada com o MUSIC no qual duas fontes estão presentes (20° e 60°).

M=4, N=256, Frequência=100 Hz, L=0,2, SNR=40



Fonte: script MATLAB do autor

A Figura 32 mostra o *pseudospectro* de um arranjo de sensores encontrando duas fontes emanando potência nos ângulos 20° e 60° em um ambiente ruidoso (ruído branco) com uma precisão bem maior do que qualquer método abordado até agora.

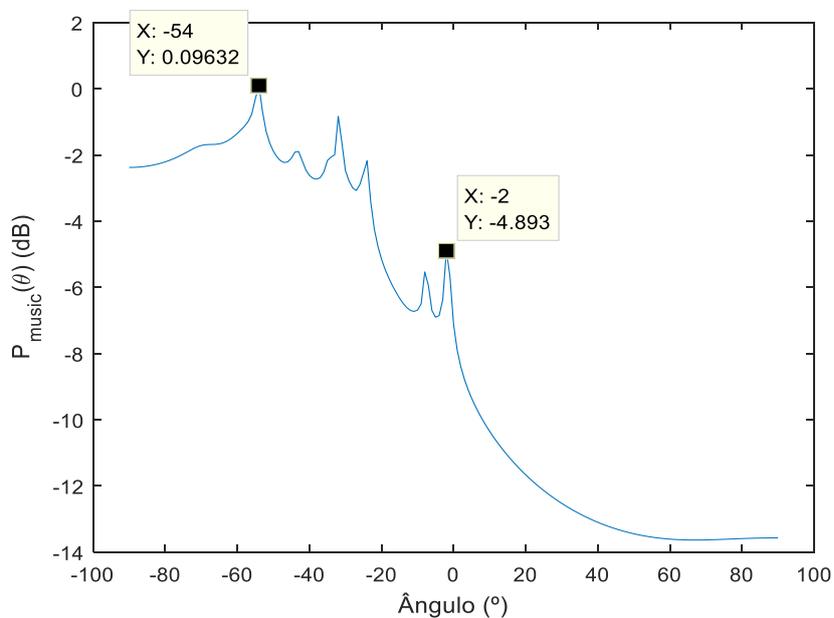
#### 4.3.2 Resultados

Seguindo a abordagem da seção 2.4, o sinal gravado é filtrado com seis filtros FIR, passa-banda, de ordem 64. O MATLAB fornece funções que auxiliam na construção destes filtros, bastando dizer a amplitude desejada em quais frequências. As bandas de interesse são fixadas em [100, 200, 300, 400, 500, 600, 700] Hz, pois a frequência fundamental do sinal de voz se encontrará nesta faixa de frequências (é mostrado na seção 6.3 obtenção da frequência fundamental).

Foram feitas simulações com duas fontes. É desejada a visualização da resolução do algoritmo à ângulos de chegada próximos. Será observado também o efeito da divisão do espectro nas sete bandas (advindas dos filtros passa faixa) de interesse. As Figuras 33, 34 e 35

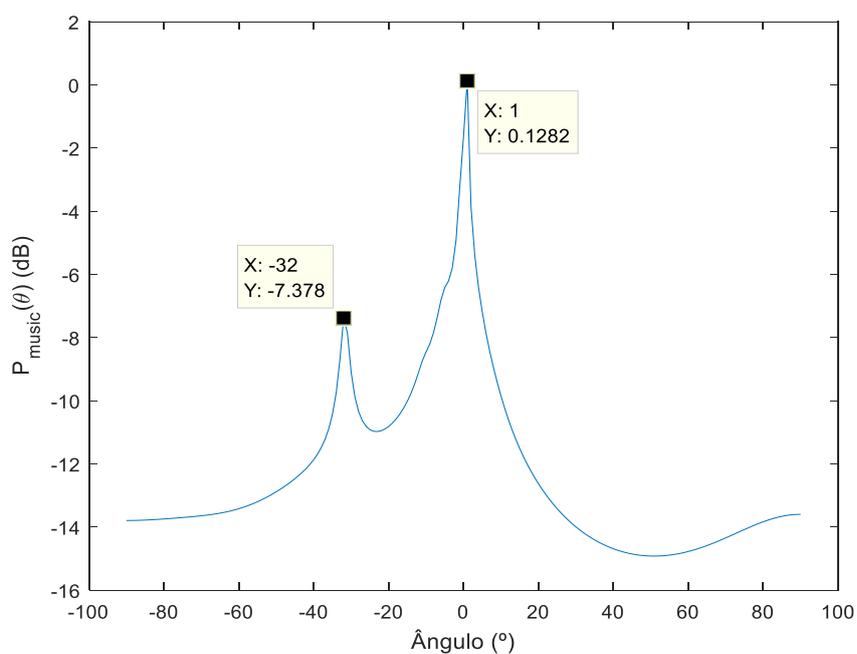
retratam uma maior resolução, quando comparado aos métodos anteriores, mesmo quando os ângulos de chegada bem próximos tem no máximo 10° de diferença.

Figura 33: Algoritmo MUSIC com ângulos de chegada = 0° (voz masculina) e -60°(voz feminina)



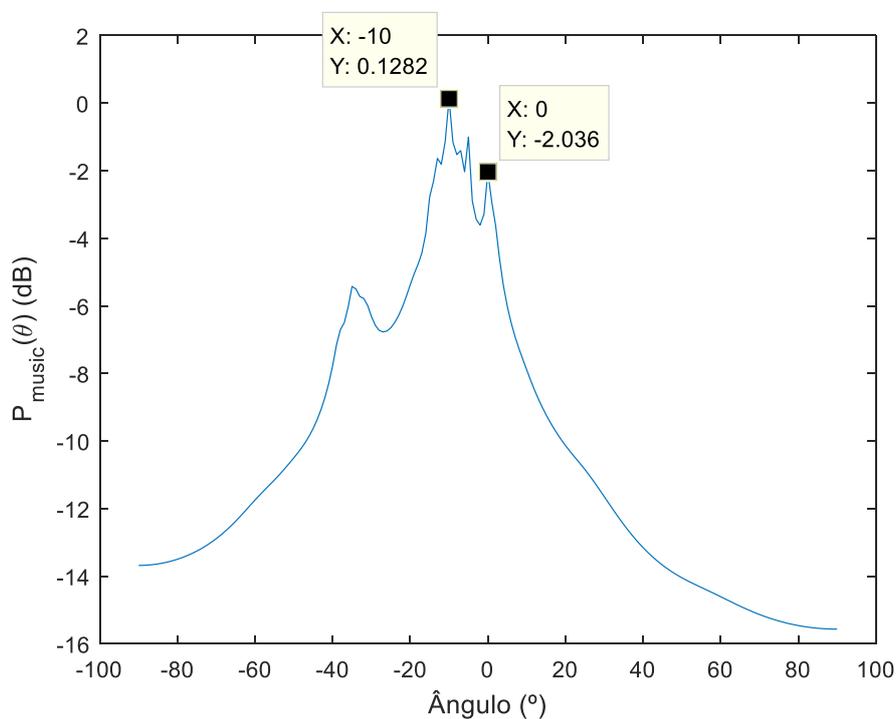
Fonte: script MATLAB do autor

Figura 34: Algoritmo MUSIC com ângulos de chegada = 0° (voz masculina) e -32° (voz feminina)



Fonte: script MATLAB do autor

Figura 35: Algoritmo MUSIC com ângulos de chegada =  $0^\circ$  (voz masculina) e  $-10^\circ$  (voz feminina)



Fonte: script MATLAB do autor

São detectadas, entretanto, duas desvantagens desta abordagem. O primeiro, derivado da divisão do sinal em bandas se mostra nas Figuras 33 e 35. Como o período fundamental do sinal não é conhecido de antemão, não é conhecido em qual banda de frequências estará a frequência fundamental, permitindo que o MUSIC identifique em cada banda, fontes baseadas nos autovetores do ruído.

O segundo aspecto negativo refere-se à complexidade computacional deste algoritmo. Para cada banda de frequências será executado o algoritmo cuja complexidade é  $O(M^2D + M^2N)$ , demandando muito mais custo computacional quando comparado ao LMS ( $O(N)$ ), inviabilizando sua implementação em sistemas embarcados simples (DHIMAN, 2013 e STOCKLE, 2015).

## 5 BEAMFORMING

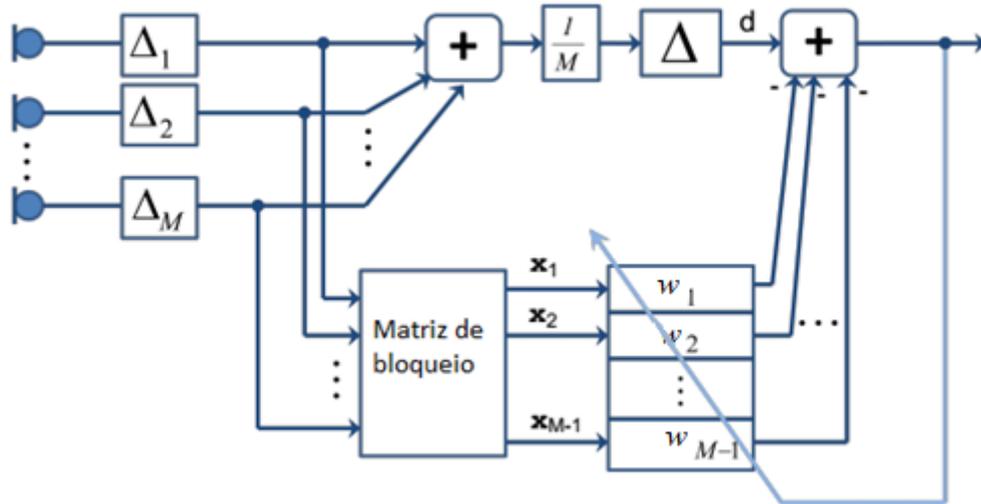
Após a obtenção do ângulo de chegada do sinal de interesse, o sinal será temporalmente alinhado com deslocamentos respectivos aos ângulos, para em seguida, reconstruir nada além do sinal desejado, eliminando os sinais interferentes. Este é o processo chamado *Beamformer*, pois tem como efeito, reduzir ao máximo a potência absorvida dos lóbulos laterais permanecendo com o ganho unitário na direção desejada. Será usado então um sistema composto por um filtro digital para cada saída dos sensores, onde os parâmetros destes filtros serão estimados com um algoritmo adaptativo.

### 5.1 *Generalized Sidelobe Canceller*

O GSC (*Generalized Sidelobe Canceller*) baseia-se no artigo publicado por Griffiths E Jim (1982) que representou um salto para o desenvolvimento em arranjos adaptativos. Comparativamente, o algoritmo antecessor, o algoritmo de Frost O. L. (1972), exigia muito processamento ao utilizar a filtragem adaptativa C-LMS (*Constrained Least Mean Squares*). Com o Frost, a solução adaptativa é restrita a uma equação matricial que permitia um ganho unitário na direção de chegada enquanto tentava minimizar a potência do sinal através de filtros adaptativos. Quando comparado ao MVDR, o GSC ou o Frost se mostram mais eficientes, pois trabalham totalmente do domínio do tempo, não sendo necessário que os passos do algoritmo sejam refeitos para outras frequências. (TOWNSEND, P. 2009)

Como se observa na Figura 36, sua topologia compreende em dois ramos, a superior consistindo em um filtro para obter uma equalização desejada, assim como um atraso temporal suficiente para alinhar os sinais que serão subtraídos da parte inferior. E seu ramo inferior consiste em uma matriz que irá bloquear o sinal de interesse (permitindo apenas ruído e sinais interferentes) e um conjunto de filtros adaptativos cujos pesos serão adaptados através do algoritmo LMS.

Figura 36: Estrutura do GSC



Fonte: BERTRAND A. BERNARDI G. (2014)

Será chamada de  $\mathbf{X}'[n] = [\mathbf{x}_1[n], \mathbf{x}_2[n] \dots \mathbf{x}_{M-1}[n]]^T$ , a matriz  $M \times N$  contendo as  $N$  amostras dos  $M$  sensores após serem alinhadas temporalmente (com o resultado obtido do ângulo de chegada). A matriz de bloqueio recomendada por Griffiths e Jim(1982) é mostrada na Equação (43):

$$\mathbf{B} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}. \quad (43)$$

A matriz de bloqueio exemplificada na Equação (43) tem  $M = 4$ .

Nota-se que a matriz apresentará  $M - 1$  linhas, o que fará com que o resultado de multiplicar  $\mathbf{X}'[n]$  pela matriz de bloqueio contenha em suas linhas as diferenças  $\mathbf{x}_1[n] - \mathbf{x}_2[n]$  até  $\mathbf{x}_{M-1}[n] - \mathbf{x}_M[n]$ .

$$\mathbf{X}''[n] = \mathbf{B}\mathbf{X}'[n].$$

Sabendo que as entradas estarão temporalmente alinhados (arranjo de microfones já 'direcionada' para a fonte alvo), a matriz com as  $M - 1$  saídas ( $\mathbf{X}''[n]$ ) conterà só o sinal ruidoso assim como os sinais interferentes.

O sinal da saída do sistema será então dado pela Equação (44):

$$y[n] = d[n] - \sum_{m=1}^{M-1} \mathbf{w}_m^H \mathbf{X}_m''[n]. \quad (44)$$

Os filtros  $\mathbf{w}_m$  serão adaptados de forma a minimizar a potência do sinal de saída. O algoritmo adaptativo usado será o NLMS (versão normalizada do LMS, evitando problemas com a escala da entrada).

Os filtros serão adaptados utilizando matriz dos coeficientes dos filtros concatenados:

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{M-2}, \mathbf{w}_{M-1}]$$

Sendo  $\mathbf{W}$  a concatenação dos pesos referentes aos  $M-1$  filtros alocados após a matriz de bloqueio. Cada vez que recebemos quadros de amostras novos iremos atualizar os pesos do filtro de acordo com a Equação (45):

$$\mathbf{W} = \mathbf{W} + \frac{\mu}{|\mathbf{X}_m''[n]|_F^2} \mathbf{X}_m''[n] y[n] \quad (45)$$

Onde  $|\mathbf{X}_m''[n]|_F^2$  é a norma de Frobenius de  $\mathbf{X}_m''[n]$ , que contém às  $N$  amostras temporais nas colunas, suas  $M-1$  amostras espaciais nas linhas.

## 5.2 Resultados

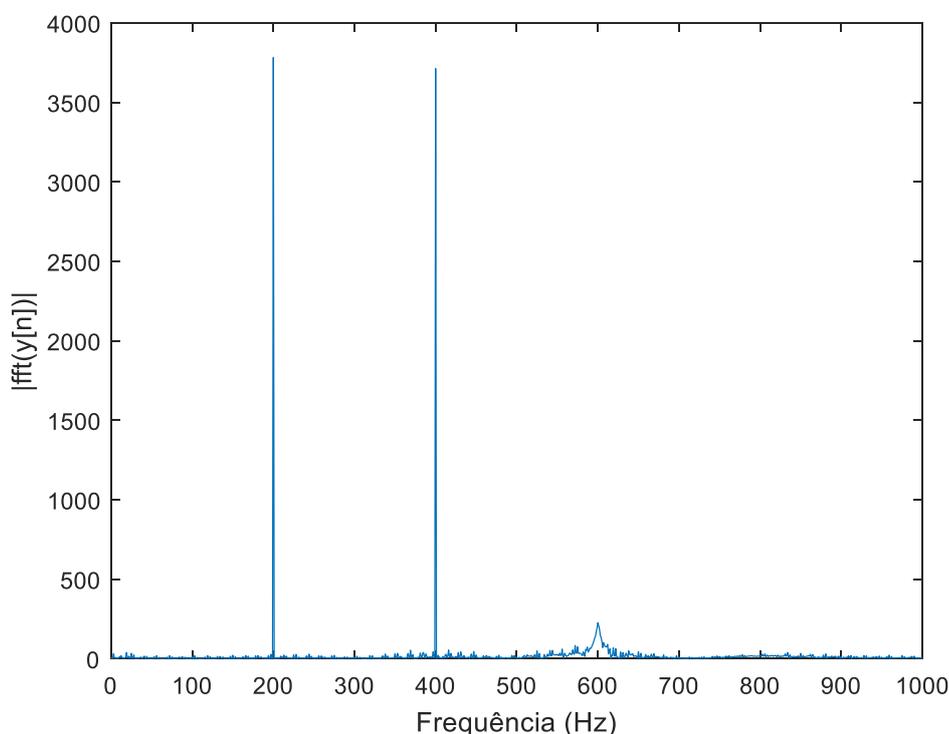
Os resultados do GSC podem ser avaliados a partir da inspeção auditiva do sinal assim como a partir da verificação do seu espectro, observando se as componentes de frequências das fontes interferentes não estão presentes. Como a inspeção auditiva do sinal não é documentável aqui, os espectros na saída do GSC foram obtidos com quatro fontes de sinais (senoidais) de entrada conhecidos em um arranjo com três microfones. A Tabela 1 detalha estes sinais de entrada.

Tabela 1: Sinais chegando ao arranjo

	Frequência (Hz)	Ângulo de chegada
Sinal 1 (interferente)	200	90°
Sinal 2 (Sinal de interesse)	400	Ângulo variante
Sinal 3 (interferente)	600	45°
Sinal 4 (interferente)	800	0°

Na a Figura 37, percebe-se que dois sinais ficaram presentes na saída do GSC, sendo estes, o sinal de interesse (ângulo de chegada =  $90^\circ$  e frequência = 400 Hz) e um segundo sinal cuja frequência é 200 Hz. Pela Tabela 1, infere-se que este último sinal chega de um ângulo de  $90^\circ$ , validando a eficiência do GSC para este caso, uma vez que apenas sinais cujo ângulo de chegada é  $90^\circ$  estarão presentes na saída enquanto que os lóbulos secundários estarão atenuados.

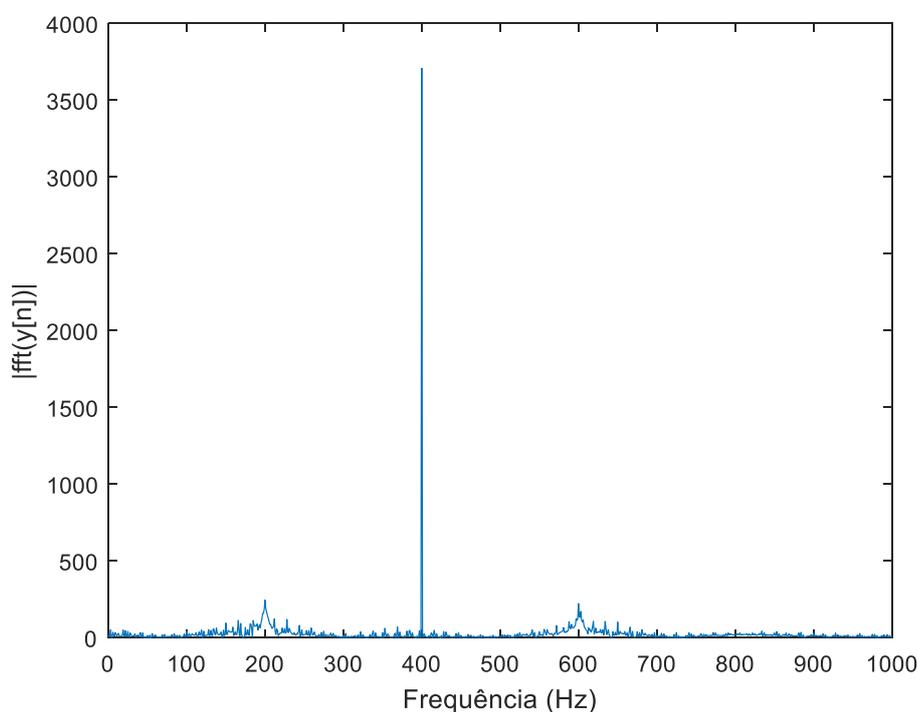
Figura 37: Módulo da transformada de Fourier da saída  
(ângulo de chegada=  $90^\circ$ )



Fonte: Script MATLAB do autor

Em um segundo cenário retratado pela Figura 38, é possível averiguar a atenuação dos sinais nas frequências de 200 Hz e 600 Hz. Para este caso, foi obtida uma atenuação de 48.84 dB em potência, representando uma atenuação significativa, quando comparada a atenuação nos lóbulos secundários do *Delay-Sum Beamformer* (Figura 8).

Figura 38: Módulo da transformada de Fourier da saída  
(ângulo de chegada= 30°)

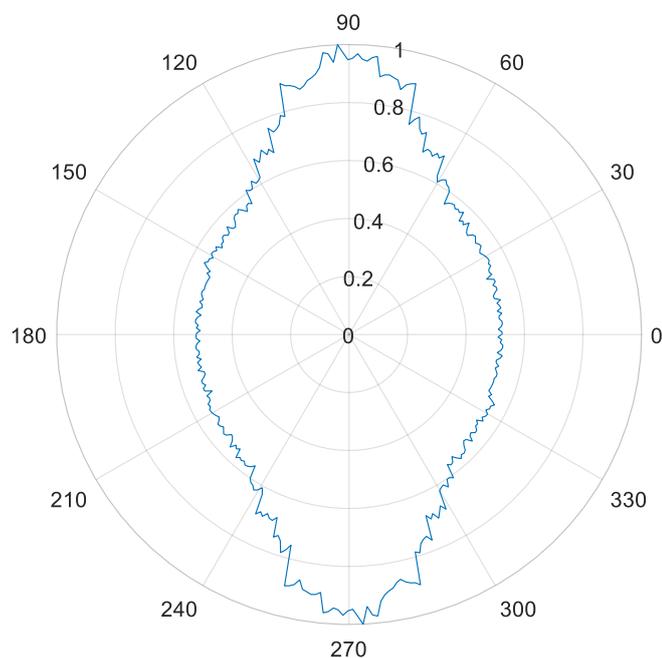


Fonte: Script MATLAB do autor

A resposta do arranjo de sensores quando dois locutores estão falando ao mesmo tempo pode ser simulada mantendo-se um dos locutores em um ângulo constante (no caso da Figura 39, a voz masculina foi mantida a 90°) e em seguida, variado-se a posição do segundo locutor (a voz feminina teve seu ângulo de chegada variado de 0° até 360°). Para simplificar, os dois sinais tiveram suas potências normalizadas e igualadas para que no resultado final, fique claro que quando metade da potência é encontrada na saída, o *Beamformer* GSC foi capaz de atenuar uma das duas fontes.

Desta forma, é possível observar na Figura 39 que de fato a potência na saída do arranjo de sensores é minimizada para aproximadamente metade da potência (apenas uma das duas fontes).

Figura 39: Potência normalizada na saída do GSC mantendo uma das fontes em 90° (voz masculina) e a segunda fonte variando de 0° até 180° (voz feminina).



Fonte: Script MATLAB do autor

Quando a direção de chegada da fonte de voz feminina se encontra no intervalo  $[60^\circ, 300^\circ] \cup [120^\circ, 240^\circ]$  é possível observar a partir da Figura 39 a ação do GSC. O sinal interferente é atenuado regularmente no intervalo, enquanto que se o sinal interferente se aproxima de  $90^\circ$  (assim como em  $270^\circ$ ) sua potência contribui para a saída, pois o GSC adaptou-se para sinais cujo ângulo de chegada é de  $90^\circ$ .

## 6 SISTEMA EMBARCADO

Para o desenvolvimento dos algoritmos simulados nas seções anteriores em um sistema embarcado, é preciso se preocupar com alguns aspectos inerentes à natureza do sinal.

Começando pela amostragem, o sistema deverá armazenar as amostras em um vetor, cujo tamanho deve ser prioritariamente uma potência de dois, podendo assim usufruir de um *Buffer* circular de memória.

O *Buffer* circular de memória é uma região na memória de um dispositivo cujo acesso é feito com dois apontadores de endereço (ponteiros na linguagem de programação C) e uma variável indicando o tamanho da região de memória. Um apontador tem o papel de identificar aonde a região do *Buffer* começa, e um segundo ponteiro indica a posição do último elemento inserido. Para um melhor desempenho, é exigido um tamanho igual a alguma potência de dois pois quando o *Buffer* precisa verificar o próximo endereço para inserção de elementos, não é necessário um condicional (expressão *if* na linguagem de programação C), mas uma operação de 'E' lógico é necessária.

A amostragem segue uma taxa constante de 8000 amostras/segundo, seguindo o padrão PCM (*Pulse Code Modulation*) para sinais de voz, no qual, todos seus formatos estarão contidos em uma banda de frequências limitada em 3400 Hz. Sendo assim, uma solução que é capaz de amostrar de várias fontes simultaneamente nesta taxa é requerida, assim como uma capacidade de processamento suficiente para que o atraso acrescido pelas filtragens não sejam inconvenientes no processo.

Seguindo os objetivos deste trabalho, foi escolhido um dos algoritmos simulados e implementaremos em uma arquitetura de *software* embarcado. A escolha do algoritmo foi a partir de critérios de custo computacional para os cálculos em cada quadro de amostras e na quantidade de memória utilizada.

Com os resultados das seções precedentes, verifica-se que para usar um *Beamformer* é necessário o conhecimento do ângulo de chegada da fonte de interesse. Desta forma, para a implementação no sistema embarcado, foi usado o *Beamformer Delay-Sum*, pois este, não recorre a nenhum processamento em alta resolução e pode ser efetuado no domínio do tempo.

Quanto ao algoritmo de detecção da direção de chegada é montada a Tabela 2 com as três opções.

Tabela 2: Complexidade dos algoritmos para encontrar a direção de chegada

Algoritmo	Complexidade
MUSIC	$O((M^2(D + N)) N_{bandas})$
Correlação	$O(MN^2)$
LMS	$O(MN)$

Fonte: Dhiman J (2013). e STOCKLE C. (2015).

Optou-se por usar a solução com o algoritmo adaptativo LMS, pois além de prover a solução de menor complexidade computacional, sua saída pode ser usada como um *Beamformer* primitivo, pois esta representa o esforço de uma das entradas de sinal tentando se aproximar da(s) outra(s), permitindo que componentes de frequências referentes às fontes indesejadas em uma entrada desapareçam.

Para a aquisição do sinal, foi utilizado um microfone eletreto cujo sinal foi amplificado pelo chip MAX9812 ilustrado na Figura 40.

Figura 40: Microfone eletreto pré-amplificador



Fonte: Artekit Italy (Acesso em 2017).

O MAX9812 oferece um ganho constante de 20dB, sendo uma solução apropriada pois foi necessário dois amplificadores (para dois conversores A/D) de mesmo ganho.

### 6.1 Software embarcado

Foi utilizado o microcontrolador TM4C123GH6PM da Texas Instruments, cujas especificações pertinentes para este projeto estão na Tabela 3.

A presença de dois conversores analógico/digital foi útil para o uso de dois microfones como entrada de sinal. Observa-se também que a taxa de amostragem destes ADC é 1M amostras/segundo, sendo possível escolher submúltiplos deste valor (500, 250 e 125 K amostras/segundo). O disparo do conversor pode ser realizado com um *TIMER* (temporizador), fornecendo a taxa de amostragem desejada. O periférico de comunicação SPI - *Serial Peripheral Interface* - é usado para comunicar com um *display* LCD Nokia 5110 e o periférico  $\mu$ DMA - *Micro Direct Memory Access* - é um subsistema no microcontrolador cuja função é a transferência de memória de forma independente, isentando o microcontrolador de operações como cópia de um bloco de memória para outra posição. O  $\mu$ DMA nos servirá para transferir as amostras do *buffer* circular preenchido pelo conversor analógico/digital para uma região de memória com a qual será feito o processamento do sinal.

Tabela 3: Especificações do microcontrolador.

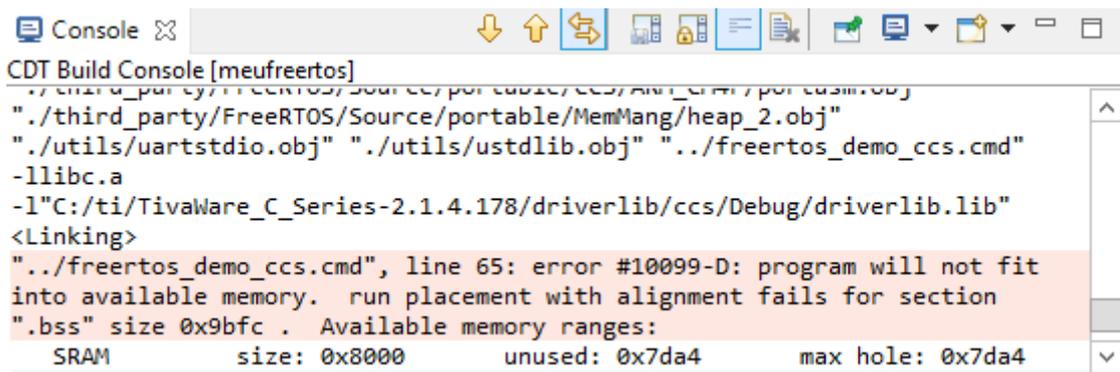
Frequência de operação (máxima)	80 MHz
Memória	256Kb Flash, 32Kb SRAM
Conversor A/D	2 ADC (1M amostras/segundo)
Periféricos	4 SPI e 1 $\mu$ DMA, 12 <i>TIMER</i> (16/32 bits)

Fonte: Texas Instruments (2007).

A arquitetura do *software* deverá permitir a execução de múltiplas tarefas de forma que o processamento do sinal não interfira na amostragem do sinal ou na exibição de informações no LCD. Neste trabalho foram testadas duas arquiteturas de *software* embarcado. Um sistema operacional de tempo real (*FreeRTOS*) e *Protothreads*.

Ao portar o sistema operacional de tempo real *FreeRTOS* no microcontrolador, foram encontrados problemas com a seção de memória *.bss* que contém variáveis alocadas estaticamente (variáveis cujo tempo de vida é o programa todo e é alocada durante a compilação). A Figura 41 mostra o erro gerado ao tentar compilar o programa. A quantidade de memória mapeada para a seção *.bss* é 0x8000 (hexadecimal) e o espaço necessário foi 0x9BFC.

Figura 41: Erro na compilação do software com o compilador da Texas Instruments



```

CDT Build Console [meufreertos]
./third_party/FreRTOS/Source/portable/CCS/ARM_Cortex_M/port_cmsm.obj
"/third_party/FreRTOS/Source/portable/MemMang/heap_2.obj"
"/utils/uartstdio.obj" "/utils/ustdlib.obj" "../freertos_demo_ccs.cmd"
-llibc.a
-l"C:/ti/TivaWare_C_Series-2.1.4.178/driverlib/ccs/Debug/driverlib.lib"
<Linking>
"./freertos_demo_ccs.cmd", line 65: error #10099-D: program will not fit
into available memory. run placement with alignment fails for section
".bss" size 0x9bfc . Available memory ranges:
  SRAM      size: 0x8000      unused: 0x7da4      max hole: 0x7da4

```

Fonte: Próprio Autor.

Uma solução para problema seria reduzir a precisão das variáveis dos *buffers* que armazenam os sinais e as variáveis dos pesos para os filtros (reduzir a precisão de 32 bits para 8 bits), porém, não foi adotada esta solução para que a precisão dos sinais não ficasse comprometida. Utilizou-se então as *Protothreads*, que fornecem uma solução menos exigente em memória.

## 6.2 Protothreads

As *Protothreads* são Macros (substituição de trechos de código no pré-processamento antes da compilação) escritos na linguagem de programação C que fornecem execução linear de código sem necessidade de interrupções ou pilhas. São extremamente leves e projetadas para sistemas com alta restrição de memória, como nos nós de redes sem fio. Para o chaveamento entre as tarefas, as *Protothreads* utilizam dois bytes de memória por tarefa, viabilizando a sua implementação no nosso sistema embarcado.

Explicações detalhadas para as *Protothreads* são fornecidos através de exemplos encontrados a partir do endereço eletrônico de seu criador, Dunkels A., que as publicou sob uma licença aberta.

Verifica-se na Listagem 1 a estrutura padrão de uma tarefa com um bloco condicional. Esta tarefa permanece bloqueada até que o contador '*counter*' seja igual a 1000, para em seguida exibir '*printf("Threshold reached\n");*'. A definição das principais macros é fornecida na Listagem 2.

Listagem 1 Exemplo de uma tarefa (escrito em C)

```
static PT_THREAD(example(struct pt *pt)){
    PT_BEGIN(pt);
    while(1) {
        PT_WAIT_UNTIL(pt, counter == 1000);
        printf("Threshold reached\n");
        counter = 0;
    }
    PT_END(pt);
}
```

Fonte: Dunkels A. (2010)

Listagem 2: Definição das principais macros utilizadas.

```
struct pt { unsigned short lc; };
#define PT_THREAD(name_args) char name_args
#define PT_BEGIN(pt) switch(pt->lc) { case 0:
#define PT_WAIT_UNTIL(pt, c) pt->lc = __LINE__; case __LINE__: \
    if(!(c)) return 0
#define PT_END(pt) } pt->lc = 0; return 2
#define PT_INIT(pt) pt->lc = 0
```

Fonte: Dunkels A. (2010)

Infere-se com a Listagem 2 que uma tarefa precisa sempre começar com ‘*PT\_BEGIN*’ e terminar com e ‘*PT\_END*’, pois estes abrem e fecham, respectivamente, um ‘*switch*’ na linguagem C. Na estrutura ‘*pt*’ encontra-se a variável ‘*short lc*’ (de onde vem os dois bytes por tarefa de cabeçalho). A Macro mais densa é a ‘*PT\_WAIT\_UNTIL(pt, c)*’ que pode ser compreendida ao observar que o fluxo da execução daquela tarefa será interrompido pelo ‘*return*’ quando a condição não é verdadeira. A macro ‘*\_\_LINE\_\_*’ é padrão da linguagem C e é substituída pela número da linha aonde é encontrada. Ao substituir as definições das macros na Listagem 1 para observar o código pré-processado obteremos a Listagem 3. O código da função principal (‘*main()*’) é encontrado na Listagem 4.

Listagem 3: Código original e versão estendida (pré-processada)

<pre> Static PT_THREAD(example(struct pt *pt)) {     PT_BEGIN(pt);     while(1) {         PT_WAIT_UNTIL(pt,             counter == 1000);         printf("Threshold reached\n");         counter = 0;     }     PT_END(pt); } </pre>	<pre> static char example(struct pt *pt) {     switch(pt-&gt;lc) { case 0:         while(1) {             pt-&gt;lc = 12; case 12:                 if(!(counter == 1000)) return 0;                 printf("Threshold reached\n");                 counter = 0;         }     } pt-&gt;lc = 0; return 2; } </pre>
--	---

Fonte: Dunkels A. (2010)

Na primeira linha do código, a macro `'PT_THREAD'` expandiu, se tornando uma declaração de função. O valor do retorno da função é usado para determinar se a tarefa está bloqueada (esperando uma condição) ou se a tarefa já executou até completar. As macros `'PT_BEGIN'` e `'PT_END'` expandiram-se formando um `'switch'`. `PT_INIT` configura `'pt->lc'` para zero e deve ser executado na função `'main()'` (Listagem 4).

Ao observar o `'while(1)'` nota-se a atribuição `'pt->lc=12'` pois 12 é o número da linha em que o bloco `'PT_WAIT_UNTIL'` é chamado. E dentro deste mesmo bloco observa-se a condição `'if(!(counter == 1000))'` que modificará o fluxo de execução do programa, saindo desta função e retornando zero (indicando que a função está aguardando a condição).

O fluxo de execução do código quando a tarefa retorna à função principal é entendido ao observar a função `'main()'` na Listagem 4.

Listagem 4: Função main() com a inicialização e execução da tarefa

```
int main(void){
    counter = 0;
    PT_INIT(&example_pt);
    while(1) {
        example(&example_pt);
        counter++;
    }
    return 0;
}
```

Fonte: Dunkels A. (2010)

A próxima vez que a função ‘*example()*’ for chamada a partir da função ‘*main()*’, a variável ‘*pt->lc*’ não será zero, mas sim, 12, como foi definida na expansão da macro ‘*PT\_WAIT\_UNTIL*’ na Listagem 3. Isso fará com que o ‘*switch(pt-> lc)*’ salte diretamente para onde a condição ‘*if(!(counter == 1000))*’ será verificada.

A instrução *printf* só será executada quando a condição for falsa e o ‘*return*’ não for executado.

Neste exemplo, é compreendido como as *Protothreads* podem ser usadas como uma alternativa de baixo consumo de memória quando é necessário um sistema com várias tarefas e um chaveamento condicional entre elas.

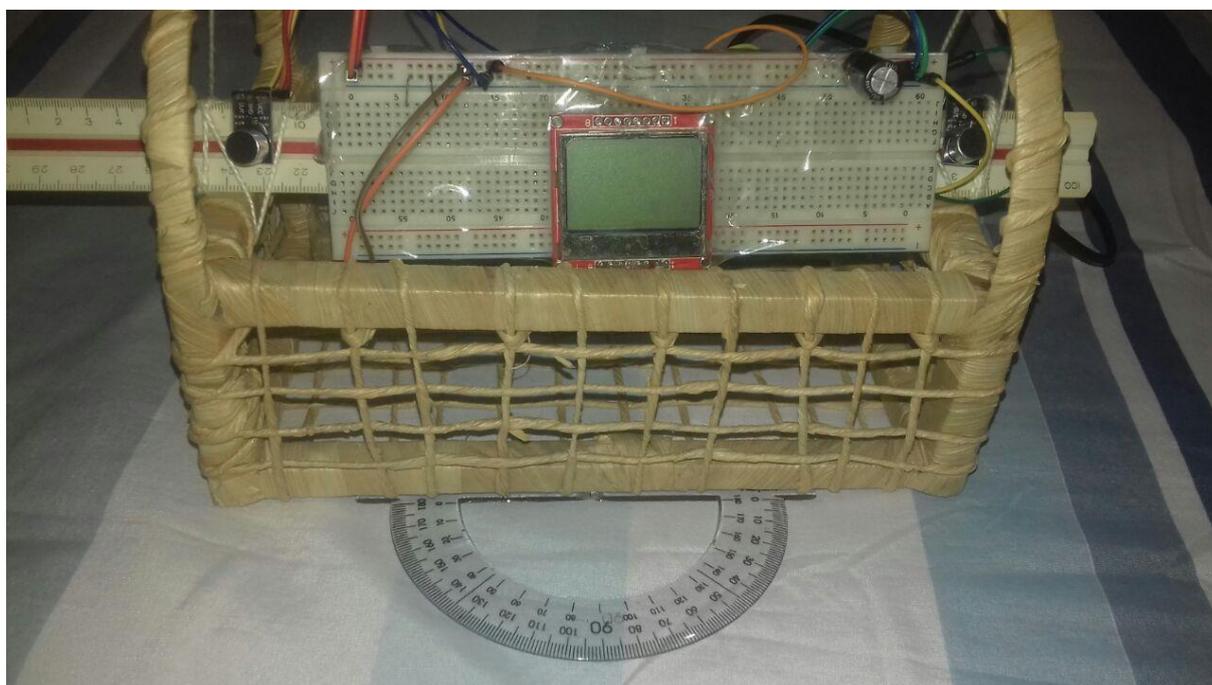
### 6.3 Implementação

O sistema foi montado em uma *Protoboard* sobre uma régua para que os microfones permanecessem em posições definidas. Os sensores foram mantidos a uma distância de 0,2 metros, como nas simulações. O *aliasing* espacial será evitado ao utilizar uma velocidade do som igual a 340,29 m/s e assumindo um sinal cuja frequência fundamental é encontrada no intervalo de 80 Hz até 500 Hz. A estimação da frequência fundamental do sinal foi identificada pelo aplicativo *PitchLab Pro* para a plataforma *Android*, no qual, a frequência fundamental é obtida ao analisar os picos da estimação do espectrograma do sinal advindo do microfone do celular.

A relação  $L \leq \frac{\lambda}{2}$  nos fornece um intervalo de frequências de [0 Hz, 850 Hz], no qual não ocorrerá o *aliasing* espacial e o sinal trabalhado estará contido neste intervalo.

Foi escolhida como fonte de tensão uma bateria de 3.7 V (o regulador de tensão na placa do microcontrolador se encarrega de entregar uma tensão igual a 3.3 V) para evitar problemas de oscilação e ruído na fonte (advindos da rede elétrica). Foi mantida a ordem do filtro adaptativo igual a 16, pois a variação máxima de amostras entre um microfone e outro é no máximo 11 (seção 4.2). Na Figura 42 é possível observar o sistema montado com um transferidor para medir os ângulos e uma régua fixando a posição dos dois microfones.

Figura 42: Arranjo de microfones com dois microfones e LCD

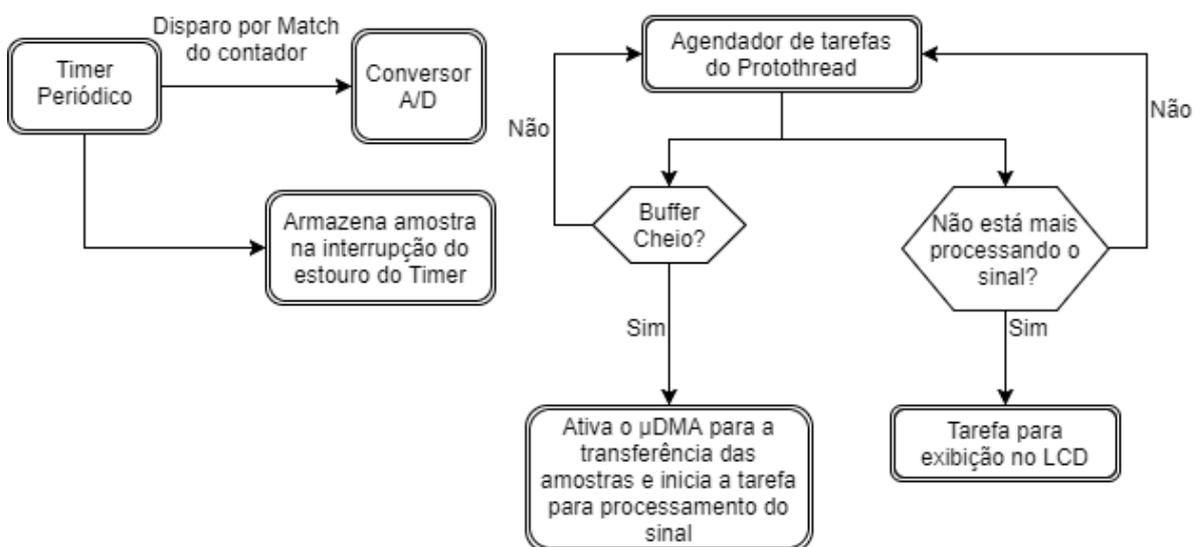


Fonte: Próprio Autor

O *software* embarcado foi feito com *Protothreads* e segue o fluxograma na Figura 43. A conversão A/D é disparada por um evento de comparação do *TIMER* com um valor (*Match*). Este valor é configurado para que o conversor A/D tenha tempo suficiente para entregar o valor convertido a um *buffer* circular de memória. No instante em que o *TIMER* alcançar o máximo de sua contagem, este ativará uma interrupção que preencherá o buffer circular de memória. Desta forma, a taxa de amostragem de 8000 amostras/segundo é garantida através do *TIMER*.

No programa principal, a tarefa na qual são processados os dados ficará bloqueada até que o *buffer* esteja completo. Com o *buffer* completo é ativado o *hardware* do  $\mu$ DMA para realizar a transferência do *buffer* com as amostras para outro espaço na memória que será usado para o processamento, evitando que novas amostras sejam inseridas enquanto o processamento é realizado. Terminado o processamento, a tarefa que exibe informações no LCD é desbloqueada, permitindo que a interface SPI do microcontrolador envie informações ao LCD.

Figura 43: Fluxograma do *Software* embarcado

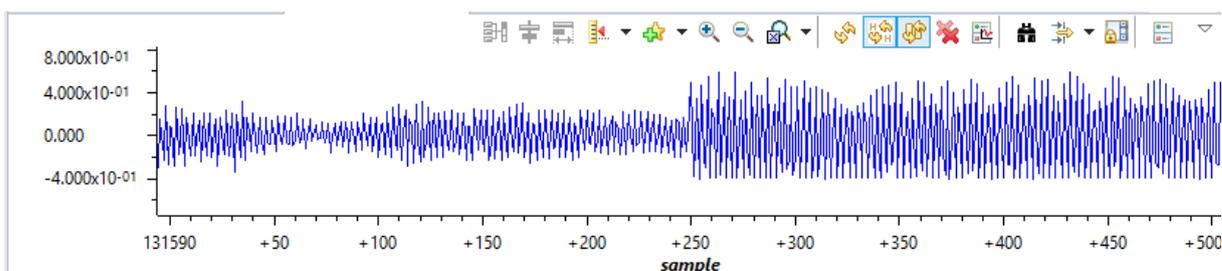


Fonte: Próprio autor

## 6.4 Resultados

É possível observar o sinal amostrado pelo microcontrolador na Figura 44. Nesta figura, nota-se a variação da amplitude do sinal recebido quando o locutor está mais perto do microfone da direita.

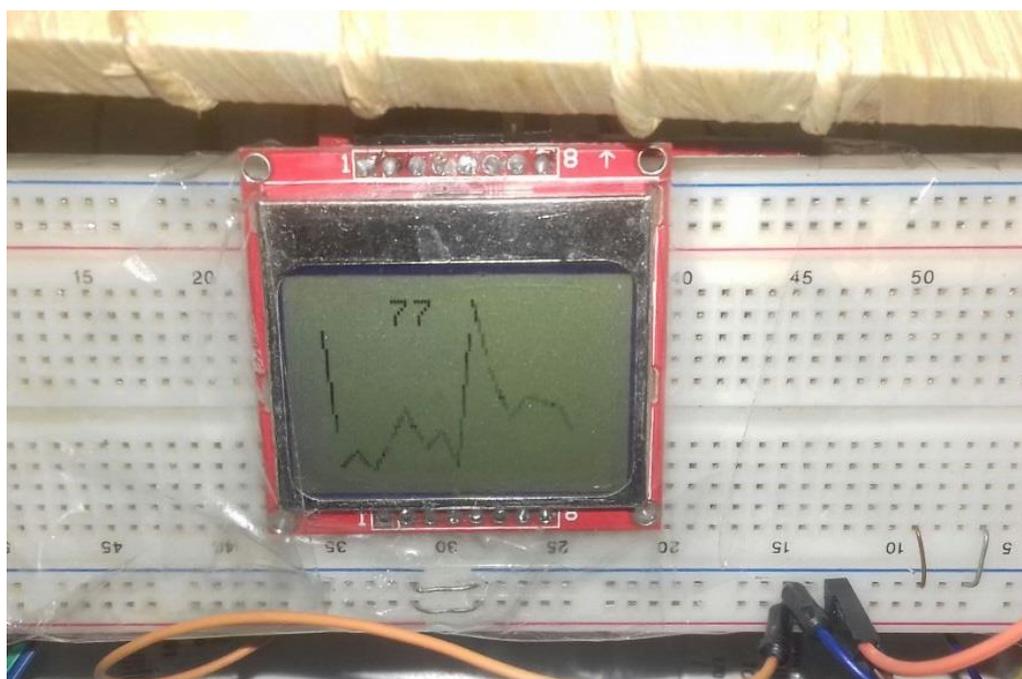
Figura 44: Locutor masculino falando mais perto do microfone da direita (a metade da direita do sinal corresponde ao microfone da direita e respectivamente, a metade da esquerda, ao microfone da esquerda)



Fonte: Próprio autor, obtido com o *debugger* da Texas Instruments

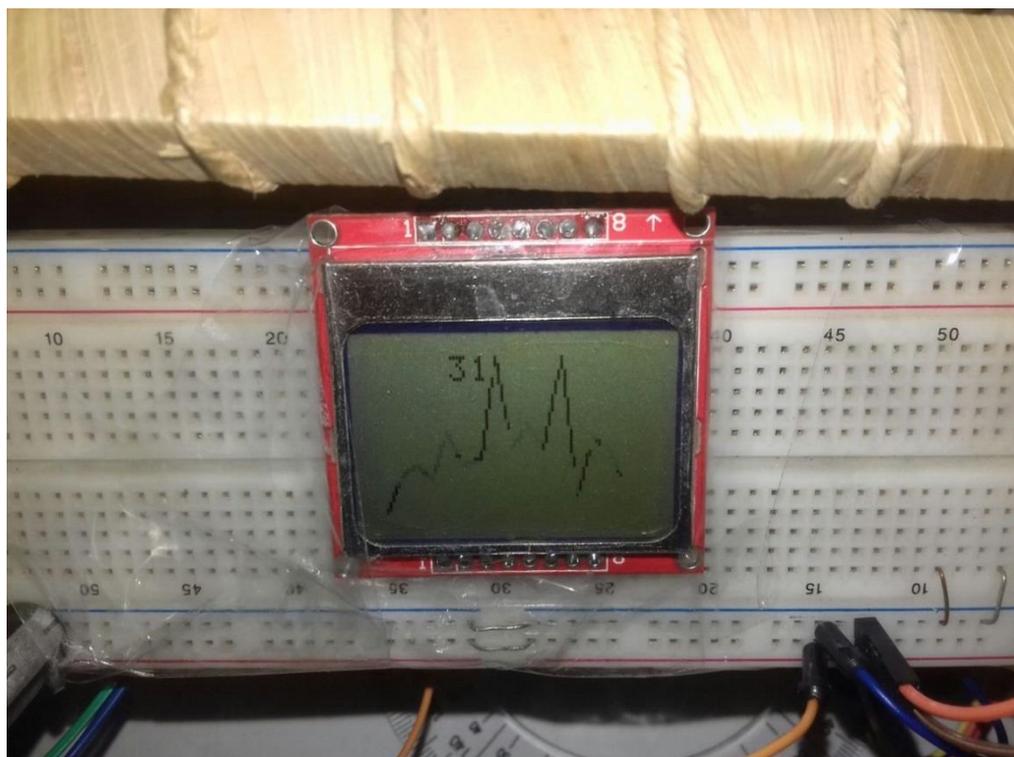
No LCD é exibido o módulo dos coeficientes do filtro adaptativo na forma de um gráfico assim como o ângulo de chegada estimado. São obtidos resultados para ângulos de chegada com um locutor masculino e um feminino encontrando o índice aonde o módulo dos coeficientes do filtro é máximo. O sistema funcionando para cada caso é mostrado nas Figuras 45, 46, 47 e 48.

Figura 45: Locutor masculino em 60° (identificou uma fonte em 77°).



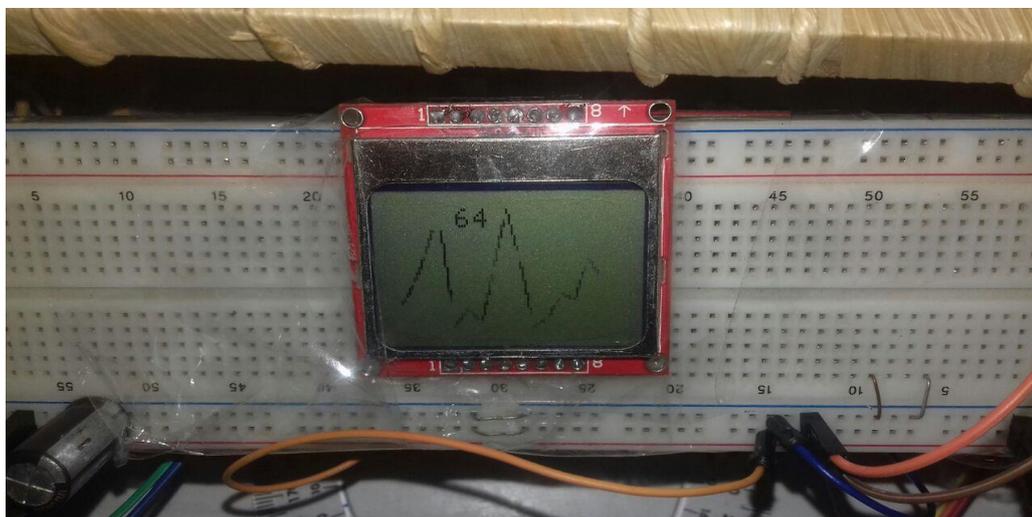
Fonte: Próprio autor

Figura 46: Locutor masculino em 30° (identificou uma fonte em 31°).



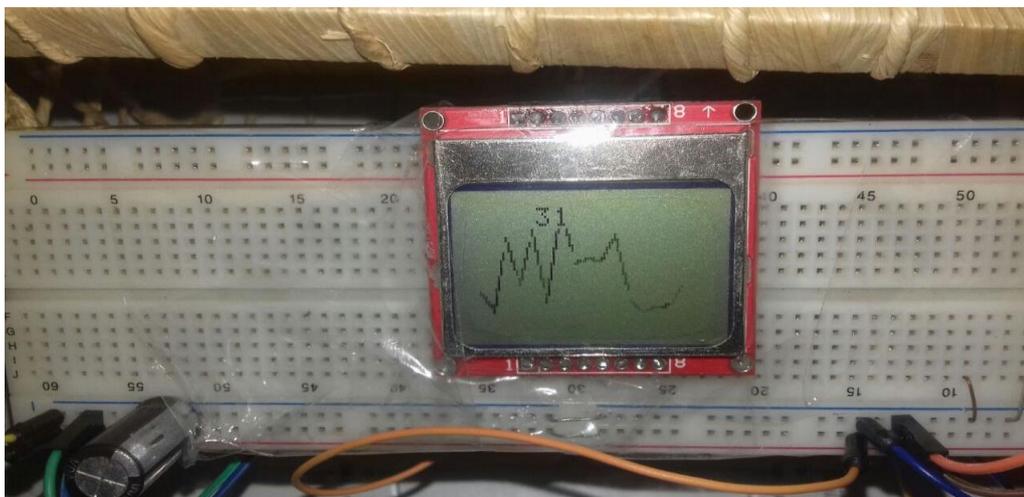
Fonte: Próprio autor

Figura 47: Locutor feminino em 60° (identificou uma fonte em 64°).



Fonte: Próprio autor

Figura 48: Locutor feminino em 30° (identificou uma fonte em 31°).

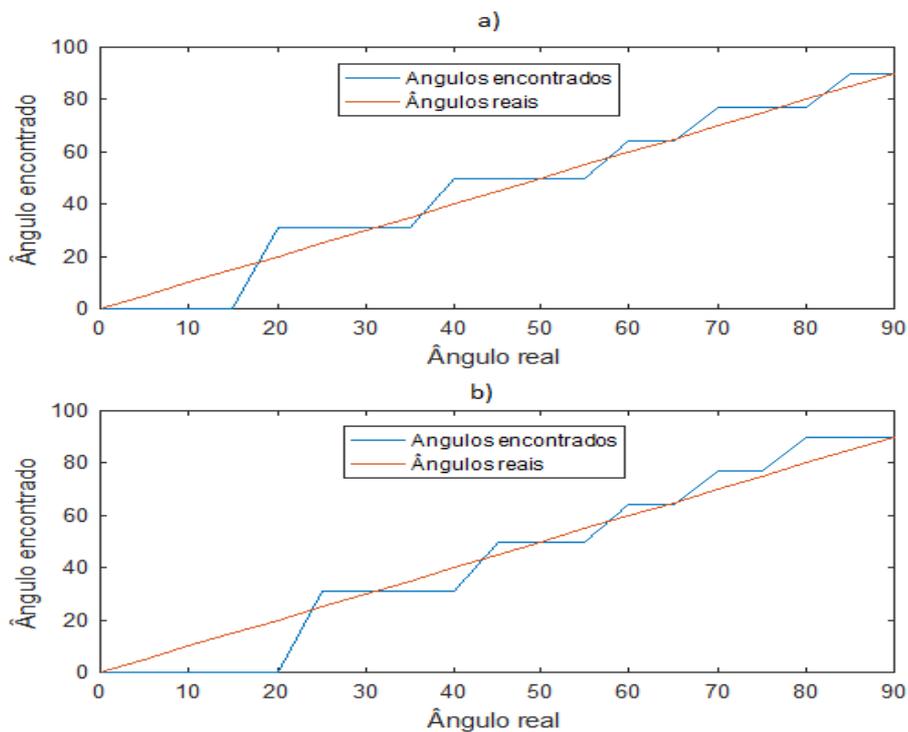


Fonte: Próprio autor

É possível notar que o problema relacionado com a precisão ao tentar encontrar o ângulo de chegada (seção 4.1.2) está presente nesta implementação com dois microfones, uma vez que estará disponível uma variação de 11 amostras para mapear os ângulos de 0° até 180°. O gráfico da Figura 49 mostra as tentativas de encontrar o ângulo real de chegada realizadas pelo arranjo de microfones. Confirma-se pela Figura 49 que a resolução do gráfico do ângulo de chegada estimado em função do real quando dois microfones estão presentes é afetada quando comparada com um arranjo de três microfones (Figura 30). Nota-se na Figura 50, o histograma do erro para os ângulos encontrados. É possível notar que os erros têm uma maior amplitude e seu histograma, uma maior variância, quando comparados com o sistema com três microfones (Figuras 24, 30 e 31).

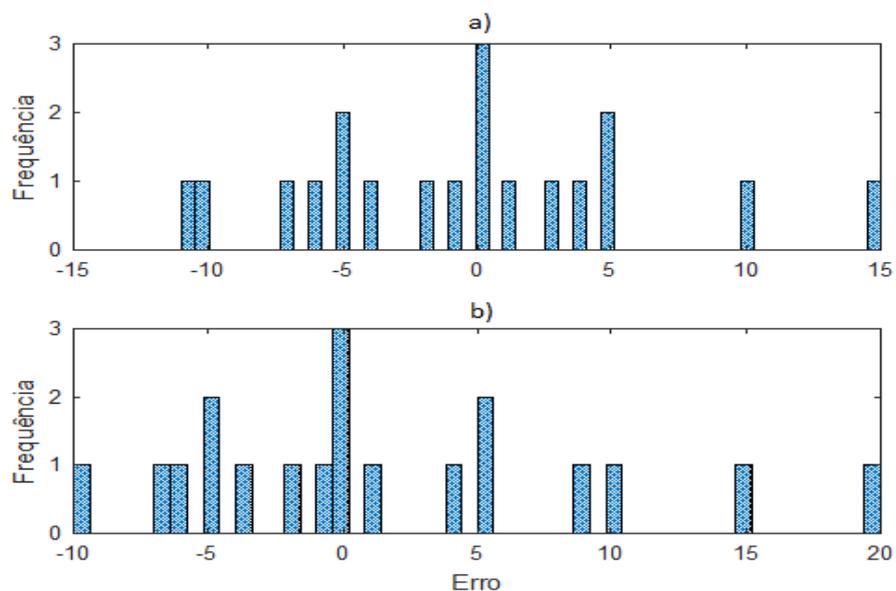
Figura 49: Ângulo de chegada encontrado em função do ângulo de chegada real.

a) Voz masculina. b) Voz feminina



Fonte: Próprio autor

Figura 50: Histograma do erro para os ângulos encontrados. a) voz masculina. b) voz feminina



Fonte: Próprio autor

## 7 CONCLUSÕES

O capítulo 4 mostrou resultados satisfatórios nas simulações ao observar que a utilização dos três microfones com o cálculo da média ponderada nos fornece um método eficaz para suavizar as curvas mostradas nas Figuras 22, 23, 30 e 31. Foi observado também que além de mais microfones, uma taxa de amostragem maior poderia mitigar o efeito da baixa resolução, pois mais amostras deslocadas estariam presentes para mapear os ângulos de 0° até 180°.

No capítulo 5 foi possível verificar a eficácia do *Beamformer* GSC ao avaliar as Figura 37 e Figura 38, indicando que sinais interferentes presentes em lóbulos secundários são atenuados. O resultado que mais se destaca é mostrado na Figura 39 quando é mostrada a potência normalizada da saída em função do ângulo de chegada da fonte interferente e percebe-se que metade (um dos dois locutores) da potência é recebida quando o locutor interferente está em qualquer posição diferente de onde o locutor principal está.

Os testes com o arranjo de sensores feito com o microcontrolador se mostraram satisfatórios dentro das suas limitações de precisão, pois dois microfones foram utilizados. Problemas com limitações com a memória *SRAM* do microcontrolador ocorreram durante a fase de desenvolvimento com o *FreeRTOS*, o que levou a optar pelas *Protothreads*. Devido à pouca quantidade de conversores A/D (dois conversores) do TM4C123, foi pesquisada uma alternativa que tivesse mais conversores A/D e suporte às instruções de processamento digital de sinais (DSP). Foi encontrado o microcontrolador STM32L476 que atendia aos critérios pesquisados e este foi comprado (Figura 51). Este microcontrolador oferece três conversores A/D e 128 *KBytes* de memória *SRAM* (em contraste com os 32Kb *SRAM* do TM4C123).

Figura 51: Placa com o microcontrolador STM32L476



Fonte: Próprio autor

O microcontrolador STM32L476 não chegou em tempo hábil para refazer toda a implementação nele, por isto, todo o projeto permaneceu no TM4C123.

Avistando trabalhos futuros, é possível pesquisar e trazer melhorias no *software* e no *hardware*.

No que concerne ao *hardware*, é planejado utilizar dois microcontroladores STM32L476, com os quais, seriam feitos o GSC *Beamforming* em um deles e a amostragem e a descoberta do ângulo de chegada com três microfones no segundo microcontrolador. Em *software*, é possível utilizar técnicas de análise de componentes independentes. Uma versão mais complexa do algoritmo ICA (conhecida como *convolutive-ICA*) pode ser usada para misturas convolutivas de sons. Contudo, seria preciso um processador digital de sinais (DSP) dedicado

ou um computador, dado que este algoritmo é custoso computacionalmente e exige soluções de problemas de ambiguidades na reconstrução dos sinais (DA COSTA J. P. *et al*, 2013).

## 8 REFERÊNCIAS BIBLIOGRÁFICAS

FURST M, BRESLOFF I, LEVINE RA, MERLOB PL, ATTIAS JJ. *Interaural time coincidence detectors are present at birth: evidence from binaural interaction*. *Hearing Research*. Vol. 187, Edição 2, Páginas 63–72, Janeiro 2004.

DE OLIVEIRA, A. C.; ROCHA, L. P.; LIMA, V. P.; NETTO, L. M.; MENEZES, P. M.; COLAFÊMINA, J. F. **Localização de fontes sonoras: a importância das diferenças dos limiares auditivos interaurais**. *Revista da Sociedade Brasileira de Fonoaudiologia*. Vol.13 No.1 São Paulo Jan./Mar. 2008.

BLAUERT J. *Spatial hearing : the psychophysics of human sound localization*. MIT Press; 1999. Páginas 30-245.

BRONSTEIN, M. *Blind source separation and deconvolution*. Disponível em <[http://www.cs.technion.ac.il/~mbron/images/research\\_bss.jpg](http://www.cs.technion.ac.il/~mbron/images/research_bss.jpg)>, Acesso em: 15 jun. 2017.

CARDOSO, F. *Blind Signal Separation: Statistical Principles*. *Proceedings of the IEEE*. Volume 86. N. 10. Outubro, 1998.

ELECTRONIC FRONTIER FOUNDATION. *Speech Compression*. Disponível em: <<http://www.data-compression.com/mmodel.gif>> Acesso em: 20 Out. 2016.

KRIM H. VIBERG M. *Two decades of array signal processing research*. IEEE *Transactions on Signal Processing Magazine*. 1996.

SILVEIRA, M. A.; SCHROEDER, C. P.; DA COSTA, J. P. *Convolutive ica-based forensic speaker identification using mel frequency cepstral coefficients and gaussian mixture models*. *The International Journal of Forensic Computer Science*. Volume. 4, Páginas 27–34. 2013.

J. CAPON, *High-Resolution Frequency-Wavenumber Spectrum Analysis*. IEEE, Volume 57. Páginas 1408-1418. 1969

LOPES, L. C. *Blind source separation applied to the analysis of rat somatosensory evoked response*. Departamento de Engenharia Elétrica, UnB, Jul 2016

BERTRAND A E BERNARDI G. *Digital Audio Signal Processing*. *Audio processing: lab sessions Session 3: DOA-informed Beamforming*. Disponível em: <[http://homes.esat.kuleuven.be/~dspuser/dasp/material/lab\\_sessions/session3\\_2014.pdf](http://homes.esat.kuleuven.be/~dspuser/dasp/material/lab_sessions/session3_2014.pdf)>. Acesso em: 5 Maio 2017.

FALLON, M. F. *Acoustic Source Localisation and Tracking using Track Before Detect*. IEEE Transactions on audio, speech, and language processing. Volume 18, No. 6, Agosto 2010.

FANT, G. *Acoustic Theory of Speech Production: With Calculations based on X-Ray Studies of Russian Articulations*. Vol 2. The Netherlands: Mouton. p. 15–45. Alemanha 1971

SCOLA, C. F. ORTEGA, M. *Direction of arrival estimation – A two microphones approach*. Blekinge Institute of Technology. Department of Signal Processing. 2010

YULE, G. U. *On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers*. Philosophical Transactions of the Royal Society. Páginas 267–298. 1927

FROST, O. L. *An Algorithm For Linearly Constrained Adaptive Array Processing*. PROCEEDINGS OF THE IEEE, Volume 60, No.8, Agosto 1972

TOWNSEND, P. *Enhancements to the Generalized Sidelobe Canceller for Audio Beamforming in an Immersive Environment*. 2009. Tese. University of Kentucky. Lexington

AARTS, M.; PRIES, H.; DOFF, *A Two Sensor Array Beamforming Algorithm for Android Smartphones*. Dissertação (Engenharia Elétrica). Delft University of Technology. Delft, Holanda. 2012.

IAIN. MCCOWAN. *Robust Speech Recognition using Microphone Arrays*, Tese (PhD em engenharia elétrica), Queensland University of Technology, Australia, 2001.

WIDROW B. M. HOFF. *Adaptive switching circuits*. IRE WASCON. Volume 4. P. 96-104.1960.

PATEL. D. M. PATEL A. M. *Simulation of MUSIC and RLS Algorithm for Adaptive Antenna Array*. International Journal Of Advanced Research In Computer Engineering & Technology (Ijarcet). Volume 4, Fevereiro, 2015

KWOK ,N. M.; BUCHHOLZ, J. *Sound Source Localization: Microphone Array Design and Evolutionary Estimation*. IEEE International Conference. p.14 – 17. 2005.

STOCKLE, C. *DoA Estimation Performance and Computational Complexity of Subspace- and Compressed Sensing-based Methods*. Workshop on Smart Antennas. 2015.

DHIMAN, J. *Comparison between Adaptive filter Algorithms(LMS, NLMS and RLS)*. International Journal of Science, Engineering and Technology Research (IJSETR). Volume 2. Maio 2013.

ARTEKIT ITALY. *AK-MAX9812 Amplified Microphone Breakout*. Disponível em: <<https://www.artekit.eu/products/breakout-boards/ak-max9812-amplified-microphone-breakout/>>. Acesso em: 15 Jun. 2017.

TEXAS INSTRUMENTS. **Tiva™ TM4C123GH6PM Microcontroller DATA SHEET**. Disponível em: < <http://www.ti.com/lit/ds/symlink/tm4c123gh6pm.pdf> >. Acesso em: 1 Jun. 2017.

DUNKELS A. **Protothreads**. Disponível em <<http://dunkels.com/adam/pt/expansion.html>>. Acesso em 10 Junho, 2017.

PITCHLAB SMARTPHONE APPS. **PitchLab Pro**. Disponível em < <https://play.google.com/store/apps/details?id=com.musician.tuner> >. Acesso em: 10 Maio 2017.

J. P. C. L. DA COSTA, C. G. DE OLIVEIRA, J. A. JUNIOR, A. M. RUBIO SERRANO. **Convolutive ICA-Based Forensic Speaker Identification Using Mel Frequency Cepstral Coefficients and Gaussian Mixture Models**. *International Journal of Forensic Computer Science (IJofCS)*, 2013.